

Combinatorische Algoritmen (a 149)

A.J. van Zanten.

mei 1988

INHOUDSOPGAVE

Hoofdstuk 1

- 1. Permutaties en combinaties
 - 1.1. Som- en produktregel
 - 1.2. Permutaties
 - 1.2.1. Definities
 - 1.2.2. Aantal permutaties
 - 1.2.3. Permutaties van niet-verschillende objecten
 - 1.3. Combinaties
 - 1.3.1. Definitie
 - 1.3.2. Aantal r -combinaties uit n
 - 1.3.3. Combinaties met herhaling
 - 1.3.4. Combinaties uit n met beperkt aantal herhalingen
 - 1.4. Verdelingen
 - 1.4.1. Definitie
 - 1.4.2. Verdelingen van verschillende objecten
 - 1.4.3. Verdelingen van niet-verschillende objecten

Hoofdstuk 2

- 2. Recurrente betrekkingen en genererende functies
 - 2.1. Inleiding
 - 2.2. Lineaire recurrente betrekkingen met konstante coëfficiënten
 - 2.2.1. Algemene oplossing homogene recurrente betrekking
 - 2.2.2. Algemene oplossing inhomogene recurrente betrekking
 - 2.2.3. Het probleem van de torens van Hanoi
 - 2.3. Genererende functies
 - 2.3.1. Definitie gewone genererende functie
 - 2.3.2. Oplossen van recurrente betrekkingen met genererende functies
 - 2.3.3. Aantal binaire bomen met n knopen
 - 2.3.4. Genererende functies voor combinaties
 - 2.3.5. Genererende functies voor permutaties
 - 2.3.6. Andere toepassingen van genererende functies

Hoofdstuk 3

- 3. Genereren van permutaties en combinaties
 - 3.1. Inleiding
 - 3.2. Genereren van permutaties
 - 3.2.1. Representatie van getallen door fakulteiten
 - 3.2.2. Lexicografische ordening
 - 3.2.3. Systematisch genereren van permutaties
 - 3.2.4. Kompleksiteit van algoritme 3.2
 - 3.2.5. Genereren van permutaties door verwisseling nabuurelementen
 - 3.3. Genereren van combinaties
 - 3.3.1. Genereren van alle combinaties uit n objecten
 - 3.3.2. Gray codes
 - 3.3.3. Genereren van de binair-gespiegelde Gray code
 - 3.3.4. Genereren van k-combinaties in lexicografische volgorde
 - 3.3.5. Genereren van k-combinaties door middel van minimale veranderingen

Hoofdstuk 4

- 4. Grafen en algoritmen
 - 4.1. Inleiding
 - 4.1.1. Definities
 - 4.1.2. Paden, afstanden en componenten
 - 4.1.3. Circuits in enkelvoudige ongerichte grafen
 - 4.1.4. Bomen
 - 4.1.5. Representatie van grafen in algoritmen
 - 4.2. Opspannende bomen
 - 4.2.1. Definities en eigenschappen
 - 4.2.2. Totale aantal opspannende bomen
 - 4.2.3. Algoritmen voor minimaalskelet
 - 4.2.4. Algoritme voor alle opspannende bomen
 - 4.2.5. Depth-first algoritme voor een opspannende boom
 - 4.3. Transitieve afsluiting van een graaf
 - 4.4. Afstanden in grafen
 - 4.4.1. Kortste pad tussen twee knopen
 - 4.4.2. Kortste paden tussen alle paren knopen
 - 4.5. Breadth-first techniek

4.6.	Circuits in een graaf
4.6.1.	Circuitruimte
4.6.2.	Snedenruimte
4.6.3.	Genereren van fundamentealcircuits
4.6.4.	Genereren van alle circuits
5.	Appendix
5.1.	Asymptotiek
5.2.	Formule van Stirling
	Lijst van referenties
	Vraagstukken
	Antwoorden

Dit diktaat is een inleiding in de diskrete wiskunde en in de theorie van kombinatorische algoritmen en is bestemd voor tweedejaars-studenten in de informatica. In het eerste gedeelte (hoofdstuk 1 en 2) worden enige eenvoudige hulpmiddelen uit de kombinatoriek behandeld. In het resterende gedeelte van het diktaat valt de nadruk meer op algoritmen, die zekere kombinatorische structuren genereren. In hoofdstuk 4 worden tevens enige elementaire eigenschappen uit de grafentheorie behandeld.

Het diktaat volgt -met name wat de algoritmen in de hoofdstukken en en 4 betreft- de stof van [1]. Voor gebruik naast dit diktaat en voor verdere studie wordt dit boek ten zeerste aanbevolen.

Andere boeken die een goede inleiding bieden wat betreft de diskrete wiskunde zijn [5] en [11] en wat betreft de theorie der kombinatorische algoritmen [7] en [9]. Degene die speciaal geïnteresseerd is in andere aspecten van de diskrete wiskunde wordt naar [2] verwezen. In [3] tenslotte vindt men een goede inleiding in de (niet-algoritmische) grafentheorie.

Het tentamen bestaat voornamelijk uit vraagstukken, waarmee de vaardigheid wordt getoetst in het oplossen van kombinatorisch getinte problemen. Echter wordt er met nadruk op gewezen dat tentamenvragen ook de in het diktaat behandelde theorie kunnen betreffen.

Als oefenstof is achter in dit diktaat een aantal naar hoofdstuk gerangschikte vraagstukken opgenomen. Tevens is er een bundel oude tentamenvraagstukken (waarvan sommige met uitwerking) verkrijgbaar getiteld Vraagstukken Combinatorische Algoritmen.

De van een † voorziene gedeelten in dit diktaat behoren niet tot de tentamenstof. De van een * voorziene gedeelten dienen slechts als nadere toelichting op de stof. Hierover zullen géén tentamenvragen worden gesteld.

1. PERMUTATIES EN KOMBINATIES

In de kombinatoriek houdt men zich bezig met het tellen van hoe vaak één of meerdere gebeurtenissen kunnen optreden. In de eerste hoofdstukken van dit diktaat zullen een aantal elementaire hulpmiddelen uit de kombinatoriek besproken worden.

1.1. Som - en produktregel

Bij het oplossen van een groot aantal eenvoudige kombinatorische problemen kan men gebruik maken van de volgende regels, die we hier zonder bewijs meedelen.

a. Somregel

Als de gebeurtenissen A en B niet tegelijkertijd plaats kunnen vinden en als A op m manieren gerealiseerd kan worden en B op n manieren dan zijn er $m + n$ manieren om precies één van beide gebeurtenissen te realiseren ($A \oplus B$).

b. Produktregel

Als A een gebeurtenis is, die op m manieren gerealiseerd kan worden en B op n manieren, dan zijn er mn manieren om beide gebeurtenissen te realiseren ($A \wedge B$).

In bovenstaande regels is aangenomen dat A en B onafhankelijk van elkaar zijn. Is dit niet het geval dan wordt het tellen van het aantal mogelijkheden i.h.a. ingewikkelder.

Beide regels worden meestal toegepast zonder dat dit expliciet vermeld wordt en vaak gebeurt dit zelfs geheel intuïtief.

Voorbeeld 1

Op het menu in een restaurant staan 3 verschillende soepen, 4 vleesgerechten en 2 nagerechten. Hoeveel verschillende maaltijden zijn er mogelijk van 3 gangen?

Oplossing

De volgorde ligt - althans in onze kultuur - vast: soep, vlees, toetje. Deze "gebeurtenissen" kunnen op resp. 3, 4 en 2 manieren gerealiseerd worden. Door de produktregel twee keer toe te passen volgt dat er $3 \times 4 \times 2 = 24$ verschillende maaltijden mogelijk zijn. □

Voorbeeld 2

Als in voorbeeld 1, maar nu laten we ook toe dat een maaltijd bestaat uit 2 gangen nl. soep, vlees of vlees, nagerecht. (Hier wordt het exclusieve of bedoeld!)

Oplossing

Een maaltijd bestaat nu uit 2 of uit 3 gangen; we mogen dus de somregel toepassen.

De maaltijden van 2 gangen kunnen we weer splitsen: ze bestaan uit soep + vlees of uit vlees + toetje. Het is niet mogelijk dat één bepaalde maaltijd zowel in de ene als in de andere categorie valt; dus mag de somregel worden toegepast. Zij het totaal aantal maaltijden N_T en gebruiken we N_{SVN} , N_{SV} , N_{VN} resp. voor de aantallen maaltijden bestaande uit de in de index aangeduide zaken (S = soep, V = vlees, N = nagerecht) dan volgt $N_T = N_{SVN} + N_{SV} + N_{VN}$. Met behulp van de produktregel volgt $N_{SV} = 3 \times 4 = 12$; $N_{VN} = 4 \times 2 = 8$, zodat $N_T = 24 + 12 + 8 = 44$. $\square$

1.2. Permutaties

Een veel voorkomend begrip in de kombinatoriek is het begrip permutatie, waarvan we de volgende definitie geven.

1.2.1. Definitie

Een r-permutatie uit n is een geordende selectie van r verschillende objecten gekozen uit een verzameling R van n verschillende objecten.

Voor $r < n$ noemt men een r-permutatie ook wel een variatie; voor $r = n$ spreekt men eenvoudig van een permutatie.

Voor $R := \{a, b, c\}$ zijn alle 2-permutaties uit de 3 objecten van R : ab , ba , ac , ca , bc en cb .

Uit de definitie en dit voorbeeld volgt dat een r-permutatie

$w := a_{i_1} a_{i_2} \dots a_{i_r}$ geïnterpreteerd kan worden als een strikt woord van de lengte r over het alfabet $R := \{a_1, a_2, \dots, a_n\}$.

Onder een strikt woord over een alfabet R verstaan we een woord, waarin een letter uit R hoogstens één keer kan voorkomen.

Men kan een r-permutatie ook opvatten als een injectie $f: N_r \rightarrow R$,

$f(k) = a_k$. Hierin is N_r de verzameling van de eerste r natuurlijke getallen.

Is $R := \{a, b, c, d\}$ dan is de 3-permutatie abc een strikt woord ter lengte 3 over R ofwel het is de injectie $f : N_3 \rightarrow R$ met $f(1) = a$, $f(2) = b$ en $f(3) = c$.

1.2.2. Aantal permutaties

Zij $P(n, r)$ het totale aantal r -permutaties uit n . Interpreteer een r -permutatie als een strikt woord ter lengte r . De eerste letter kan dan op n manieren gekozen worden, de tweede op $(n-1)$ manieren etc. Door toepassing van de produktregel vinden we

$$P(n, r) = n(n-1) \dots (n-r+1) = \frac{n!}{(n-r)!} \quad (1.1)$$

De functie $n!$ wordt op de gebruikelijke wijze gedefinieerd volgens

$$n! := \begin{cases} (n-1)!n, & n = 1, 2, \dots \\ 1, & n = 0. \end{cases}$$

Tevens vermelden we hier de verwante functies

$$[n]_r := \begin{cases} n(n-1)(n-2)\dots(n-r+1), & r = 1, 2, \dots, n \\ 1, & r = 0 \end{cases} \quad (1.2)$$

en

$$[n]^r := \begin{cases} n(n+1)(n+2)\dots(n+r-1), & r = 1, 2, \dots \\ 1, & r = 0 \end{cases} \quad (1.3)$$

die beide voor $n \in \mathbb{N}$ gedefinieerd zijn. Ze worden de dalende resp. stijgende fakulteit genoemd. Er geldt dus $P(n, r) = [n]_r$.

Voorbeeld 3

Op een bijeenkomst moeten 5 sprekers (A, B, C, D en E) het woord voeren; op hoeveel manieren kan dit?

Oplossing

Dit is een probleem waarbij alleen de volgorde van de 5 voorwerpen (sprekers) van belang is. De overige mogelijkheden, zoals b.v. alle 5 's ochtends of des middags, met of zonder microfoon kunnen buiten beschouwing blijven omdat hierover in de opgave niets gezegd of gesuggered wordt. Het antwoord is dus $P(5, 5) = 5! = 120$. $\square$

Voorbeeld 4

Als in voorbeeld 3 maar nu met de eis dat spreker A vóór spreker B aan de beurt moet komen.

Oplossing

Het gaat alleen om het aantal volgorden, maar nu is er geen formule beschikbaar die direkt het juiste antwoord oplevert. Op grond van de symmetrie volgt dat het antwoord luidt $\frac{1}{2} 5!$, want bij elke volgorde met A vòòr B behoort één volgorde met B vòòr A. We kunnen ook het aantal volgorden splitsen in 5 groepen al naar gelang de plaats die A inneemt, het aantal mogelijkheden per groep bepalen en dan die 5 aantallen bij elkaar optellen. (de som-regel mag hier worden toegepast). $\square$

Voorbeeld 5

Als in voorbeeld 4 maar nu moet B direkt na A spreken.

Oplossing

We kunnen nu AB samen als 1 voorwerp beschouwen en de zo gevormde 4 voorwerpen AB,C,D en E over 4 plaatsen verdelen. De 5e plaats wordt dan na de vaststelling der volgorde toegekend aan het voorwerp AB dat op die manier aan de 2 benodigde plaatsen komt. Er zijn dus $P(4,4) = 4! = 24$ volgorden mogelijk. $\square$

Voorbeeld 6

Op hoeveel manieren kunnen de 5 sprekers na afloop van de bijeenkomst aan een ronde tafel gaan zitten?

Oplossing

Ook hier gaat het alleen om de volgorde. Als de stoelen rond de tafel genummerd waren dan zouden er $5!$ mogelijkheden zijn om plaatsen aan te wijzen. Nu dat niet zo is maakt het niets uit waar de eerste van de 5 gaat zitten. De onderlinge volgorde wordt bepaald door de manier waarop de overige 4 hun plaats innemen; dus er zijn $4! = 24$ manieren. $\square$

Voorbeeld 7

Op hoeveel manieren kunnen 5 mannen en 5 vrouwen aan een ronde tafel gaan zitten zodat elke man tussen 2 vrouwen zit?

Oplossing

Plaats eerst de 5 mannen op 5 stoelen zó dat om en om een stoel open blijft; dat kan op $4!$ manieren. Plaats vervolgens de 5 vrouwen; dat kan nu op $5!$ manieren (de plaatsing van de eerste vrouw is op 5 verschillende manieren mogelijk). Totaal zijn er dus $4! \times 5!$ mogelijkheden. $\square$

Voorbeeld 8

Als D een verzameling is met r elementen en R is een verzameling met n elementen, bewijs dan dat het totaal aantal afbeeldingen $f : D \rightarrow R$ gelijk is aan n^r . Toon ook aan dat het totale aantal injectieve afbeeldingen gelijk is aan $[n]_r$.

Oplossing

Het eerste antwoord volgt onmiddellijk na toepassing van de produktregel en het tweede volgt direkt uit de interpretatie van een r -permutatie uit n als een injectie. $\square$

Voorbeeld 9

Een collegezaal die 201 zitplaatsen heeft, wordt gebruikt om een college voor 200 studenten te geven. Op hoeveel manieren kan de eerste rij die 20 plaatsen heeft bezet worden?

Oplossing

Op de eerste rij kunnen 20 of 19 studenten zitten; minder dan 19 is niet mogelijk.

Definieer nu N_T := totale aantal plaatsingen van studenten op de eerste rij, N_{20} := aantal plaatsingen van 20 studenten op de eerste rij en N_{19} analoog. Nu geldt: $N_T = N_{19} + N_{20}$ en $N_{20} = P(200, 20) = \frac{200!}{180!}$. Het aantal N_{19} kan op verschillende manieren bepaald worden.

a) Met de produktregel. 2 handelingen: I wijs een lege plaats aan; II vul de overige 19 plaatsen. I kan op 20 manieren en II kan op $P(200, 19) = \frac{200!}{181!}$ manieren; dus $N_{19} = 20 \times \frac{200!}{181!}$.

b) Met de somregel. De lege plaats is of de 1^e of de 2^e of ... of de 20^e plaats van de eerste rij. Definieer N_{19}^i := aantal manieren om de 1^e rij te vullen waarbij plaats i leeg blijft ($i = 1, 2, \dots, 20$); hieruit volgt $N_{19} = \sum_{i=1}^{20} N_{19}^i$. Voor $i = 1, 2, \dots, 20$ geldt

$$N_{19}^i = P(200, 19) \text{ zodat } N_{19} = 20 \times \frac{200!}{181!} \quad (\text{als bij a}).$$

$$\text{Uit } N_T = N_{20} + N_{19} \text{ volgt dat } N_T = \frac{200!}{180!} + 20 \times \frac{200!}{181!}. \quad \square$$

Opmerking Men kan ook als 201^e voorwerp aan de 200 studenten een lege plaats toevoegen en dan $N_T = P(201, 20)$ berekenen. Hierdoor heeft men in één stap het totaal aantal mogelijkheden zowel met 19 als met 20 studenten op de 1e rij.

Probeer zelf een kombinatorische afleiding te vinden van de betrekkingen

$$P(n, r) = (n - r + 1) P(n, r - 1), \quad (1.4)$$

$$P(n, r) = P(n - 1, r) + rP(n - 1, r - 1). \quad (1.5)$$

1.2.3. Permutaties met niet-verschillende objecten

In de vorige paragrafen ging het om permutaties van objecten die alle verschillend waren. In deze paragraaf onderstellen we, dat we n objecten hebben, waarvan er m_1 zijn van de 1^e soort, m_2 van de 2^{de} soort, ..., m_k van de k^{de} soort. Verder nemen we aan dat objecten van de i^{de} soort ($i = 1, 2, \dots, k$) onderling niet van elkaar te onderscheiden zijn. In een n -permutatie van de n objecten zal een verwisseling van twee objecten van dezelfde soort dan ook géén aanleiding geven tot een andere permutatie.

Het aantal verschillende (n -)permutaties van de n objecten wordt nu als volgt berekend.

Breng op elk van de n objecten een eigen kenmerk aan; er zijn dan $n!$ permutaties. Verwijder deze kenmerken van de m_1 objecten van de 1ste soort; er zijn nu nog $n!/m_1!$ verschillende permutaties mogelijk. Doe nu hetzelfde met de m_2 objecten van de tweede soort; dan zijn er nog $n!/(m_1! \cdot m_2!)$ verschillende permutaties mogelijk.

Zo voortgaande vinden we dat het aantal permutaties van n objecten waarvan er m_i tot de i -de soort behoren en $m_1 + m_2 + \dots + m_k = n$ gelijk is aan

$$\frac{n!}{m_1! \cdot m_2! \cdot \dots \cdot m_k!} \quad (1.6)$$

Voorbeeld 10

Op hoeveel manieren kunnen 3 rode, 2 witte en 5 blauwe ballen op een rij gelegd worden?

Oplossing

Dit kan op $\frac{10!}{3! \cdot 2! \cdot 5!} = 2520$ manieren. $\square$

Voorbeeld 11

In een vaas zitten 4 blauwe, 2 rode en 2 witte ballen. Men haalt er 6 ballen uit en legt die op een rij. Hoeveel verschillende rijen zijn er mogelijk?

Oplossing

We onderscheiden de volgende mogelijkheden met de daarbij behorende aantallen, waarop de ballen gerangschikt kunnen worden:

4b, 2r	$\frac{6!}{4!2!0!}$
4b, 2w	$\frac{6!}{4!2!0!}$
4b, 1r, 1w	$\frac{6!}{4!1!1!}$
3b, 2r, 1w	$\frac{6!}{3!2!1!}$
3b, 1r, 2w	$\frac{6!}{3!2!1!}$
2b, 2r, 2w	$\frac{6!}{2!2!2!}$

Met de somregel volgt dan dat we deze aantallen bij elkaar op moeten tellen om het totaal aantal mogelijkheden te krijgen; dit is dus gelijk aan

$$2 \cdot \frac{6!}{4!2!} + \frac{6!}{4!} + 2 \cdot \frac{6!}{3!2!} + \frac{6!}{2!2!2!} = 5 \cdot 6 + 5 \cdot 6 + 4 \cdot 5 \cdot 6 + 3 \cdot 5 \cdot 6 = 9 \cdot 30 = 270$$

□

Voorbeeld 12

Twee schakers A en B konstateren nadat ze 24 partijen tegen elkaar geschaakt hebben dat A 8 partijen gewonnen heeft, B 7 partijen en dat de overige 9 partijen zijn geëindigd in remise.

Op hoeveel manieren kan hun tweekamp verlopen zijn?

Oplossing

Op $\frac{24!}{8!7!9!}$ manieren. □

Voorbeeld 13

Hoe groot is het aantal binaire getallen ter lengte n , dat een even aantal nullen bevat?

Oplossing

Rechtstreeks toepassen van (1.6) levert dat het gevraagde aantal gelijk is aan

$$N = \sum_{k \text{ even}} \frac{n!}{k!(n-k)!} = \sum_{k \text{ even}} \binom{n}{k},$$

waarbij $\binom{n}{k}$ ("n over k") een binomiaalkoefficient is. Met behulp van de bekende uitdrukkingen

$$\sum_{k=0}^n \binom{n}{k} = 2^n \tag{1.7}$$

en

$$\sum_{k=0}^n (-1)^k \binom{n}{k} = 0 \tag{1.8}$$

kan men nu afleiden dat $N = 2^{n-1}$.

Dit antwoord laat zich veel sneller verkrijgen op de volgende manier.

Door het n^{de} bit te veranderen kunnen we inzien dat bij elk binair getal met een even aantal nullen een getal hoort met een oneven aantal nullen en omgekeerd.

Het aantal getallen met een even aantal nullen moet dus gelijk zijn aan het aantal getallen met een oneven aantal nullen en dus is dit aantal $\frac{1}{2} 2^n = 2^{n-1}$. $\square$

Uit bovenstaand voorbeeld blijkt dat symmetriebeschouwingen soms bijzonder nuttig kunnen zijn. Een andere illustratie daarvan volgt hieronder.

Voorbeeld 14

Hoe groot is het aantal quaternaire getallen (getallen bestaande uit de cijfers 0,1,2 en 3) ter lengte n met een even aantal nullen?

Oplossing

Verdeel nu alle 4^n quaternaire getallen in twee groepen: 2^n getallen die slechts tweeën en drieën bevatten en $4^n - 2^n$ getallen die 1 of meer nullen of enen bevatten. De getallen van de eerste groep bevatten uiteraard een even aantal nullen. De getallen van de 2de groep kunnen worden onderverdeeld in klassen die worden gekenmerkt door een bepaald patroon van tweeën en drieën, bijv. $13xx3x2x$, waarin de symbolen x nullen en enen voorstellen. Wegens symmetrieredenen bevat de helft van het totale aantal getallen in elke klasse een even aantal nullen. Daarom zijn er $2^n + \frac{1}{2}(4^n - 2^n) = \frac{2^n + 4^n}{2}$ getallen met een even aantal nullen. $\square$

Toon nu zelf aan, dat het aantal quaternaire getallen ter lengte n met een even aantal nullen en een even aantal enen gelijk is aan

$$\frac{4^n}{4} + \frac{2^n}{2}.$$

Opmerking

In de zogenaamde Polya-theorie wordt op elegante wijze verband gelegd tussen symmetrie en kombinatoriek. De geïnteresseerde lezer zij hiervoor verwezen naar hoofdstuk 5 van het boek van Liu [2].

1.3. Combinaties

Naast de in 1.2. besproken permutaties speelt vooral het begrip combinatie een belangrijke rol in de kombinatoriek.

1.3.1. Definitie

Een r -combinatie uit n is een ongeordende selectie van r verschillende objecten uit een verzameling R van n verschillende objecten.

Voor $R := \{a,b,c\}$ luiden alle 2-combinaties uit 3 : ab , ac en bc .

Een r -kombinatie $w := a_{i_1} a_{i_2} \dots a_{i_r}$ uit $R := \{a_1, a_2, \dots, a_n\}$ kan weer op verschillende manieren geïnterpreteerd worden, analoog als bij r -permutaties.

Omdat de volgorde der elementen van R in w er niet toe doet, kunnen we er altijd voor zorgen dat $i_1 < i_2 < \dots < i_r$. Daarom kan w geïnterpreteerd worden als een strikt monotoon woord ter lengte r over het alfabet R . (N.B. aap is geen strikt monotoon woord, alp is dat wel).

Vat men w op als een strikt monotoon woord dan is hiermee tevens te definiëren een monotone injectie $f : N_r \rightarrow R$, $f(k) = a_{i_k}$ d.w.z. $f(k)$ is gelijk aan de k^{de} letter van het woord w .

Schrijven we tenslotte w als $\{a_{i_1}, a_{i_2}, \dots, a_{i_r}\}$ dan beschouwen we w als deelverzameling van R met r elementen.

1.3.2. Aantal r -kombinaties uit n

Zij $C(n, r)$ het totale aantal r -kombinaties uit n . Elke r -kombinatie kan op $r!$ manieren geordend worden en geeft aldus aanleiding tot $r!$ permutaties. Hieruit volgt

$$C(n, r) = \frac{P(n, r)}{r!} = \frac{n!}{r!(n-r)!} = \binom{n}{r}. \quad (1.9)$$

Opmerkingen

1. Voor $0 \leq r \leq n$ hebben we nu afgeleid dat $C(n, r) = \binom{n}{r}$. Voor $r < 0$ of $r > n$ heeft $C(n, r)$ geen fysische betekenis.
2. Voor $n < r$ definieren we $C(n, r) := 0$.
3. Voor $n < 0$ definieren we $n! := (-1)^n (-n)!$

Hiermee consistent is de afspraak

$$\binom{-n}{r} = (-1)^r \binom{n+r-1}{r}, \quad r > 0. \quad (1.10)$$

(Zie ook [4, (4.73)])

Daarentegen definieert men meestal

$$\binom{m}{-r} := 0, \quad \forall m \in \mathbb{Z}. \quad (1.11)$$

Voorbeeld 15

Hoeveel diagonalen zijn er in een konvexe 10-hoek?

Oplossing

Bekijk 1 hoekpunt. Vanuit dat punt kunnen we $10 - 3 = 7$ diagonalen trekken. Dat geldt voor alle 10 hoekpunten. Zo vinden we 10×7 diagonalen. Elke diagonaal wordt echter 2 keer geteld (vanaf zijn "beginpunt" en vanaf zijn "eindpunt"). Er zijn dus maar $\frac{1}{2} \cdot 70 = 35$ diagonalen.

Dit is ook te vinden m.b.v. combinaties. Elke combinatie van 2 uit de 10 hoekpunten levert namelijk een zijde of een diagonaal op door de 2 gekozen hoekpunten met elkaar d.m.v. een rechte lijn te verbinden. Daar er bij combinaties niet op de volgorde gelet wordt komt de diagonaal van hoekpunt A naar hoekpunt C alleen tevoorschijn bij de combinatie (A,C) en niet nogmaals bij (C,A) want dat is dezelfde combinatie van A en C. Er volgt dus dat het aantal diagonalen + het aantal zijden gelijk is aan $C(10,2)$.

Bij een regelmatige 10-hoek zullen er 10 zijden zijn zodat het aantal diagonalen gelijk is aan

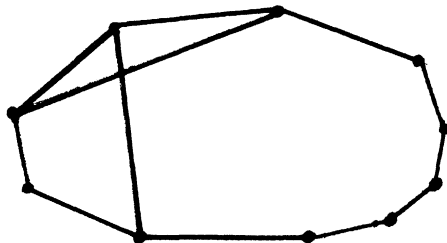
$$C(10,2) - 10 = \frac{10!}{2!(10-2)} - 10 = \frac{10 \cdot 9}{2} - 10 = 35. \quad \square$$

Voorbeeld 16

Wanneer in een konvexe 10-hoek géén drietal diagonalen door één punt gaan binnen de 10-hoek, in hoeveel lijnstukken worden de diagonalen dan verdeeld door hun onderlinge snijpunten?

Oplossing

In het vorige voorbeeld zagen we dat er 35 diagonalen zijn. Uit het konvexe karakter van de 10-hoek volgt, dat er bij elk viertal hoekpunten precies één snijpunt van de diagonalen hoort, dat binnen de 10-hoek is gelegen.



Er zijn dus $C(10,4) = 210$ snijpunten tussen de diagonalen, die binnen de 10-hoek liggen. Als er k van zulke snijpunten op een diagonaal liggen, wordt deze diagonaal in $k + 1$ stukken verdeeld. Daar elk snijpunt op 2 diagonalen is gelegen is het totale aantal segmenten, waarin de diagonalen worden verdeeld gelijk aan

$$35 + 2 \times 210 = 455. \quad \square$$

Voorbeeld 17

Iemand heeft 6 vrienden en wil er op 20 verschillende dagen telkens 3 uitnodigen. Hij wil echter nooit een bepaald gezelschap voor een 2e keer bij zich hebben. Op hoeveel manieren kan hij dat doen?

Oplossing

De volgorde op een bepaalde dag doet niet ter zake. Het gaat dus om combinaties van 3 uit 6. Hiervan is het totale aantal gelijk aan

$C(6,3) = \frac{6!}{3!(6-3)!} = \frac{6!}{3!3!} = \frac{4 \cdot 5 \cdot 6}{1 \cdot 2 \cdot 3} = 20$. Het is dus juist mogelijk om het plan uit te voeren en dat kan op $P(20,20) = 20!$ manieren, omdat op elke dag 1 van de 20 combinaties gevraagd moet worden en de volgorde hierbij wel van belang is. $\square$

Voorbeeld 18

Op hoeveel manieren kunnen we een binair getal vormen dat bestaat uit 5 enen en 3 nullen?

Oplossing

Op $C(8,3) = \frac{8!}{5!3!} = \frac{6 \cdot 7 \cdot 8}{1 \cdot 2 \cdot 3} = 56$ manieren. $\square$

Voorbeeld 19

Bewijs dat $\sum_{i=0}^n \binom{n}{i} = 2^n$ langs kombinatorische weg.

Oplossing

Het linkerlid van de betrekking stelt het totale aantal combinaties voor, die we uit n objecten kunnen selecteren, ongeacht de grootte.

(N.B. Er is $\binom{n}{0} = 1$ manier om 0 objecten te kiezen, nl. door alle n objecten te laten liggen!)

We kunnen dit aantal ook krijgen door de rij voorwerpen langs te gaan en bij elk voorwerp opnieuw te kiezen of we het wel of niet in de selectie zullen opnemen. De produktregel levert dan, dat er 2^n verschillende selecties mogelijk zijn. $\square$

Bewijs nu zelf langs kombinatorische weg de betrekkingen ($0 \leq r \leq n$)

$$C(n,r) = C(n-1,r-1) + C(n-1,r), \quad (1.12)$$

$$C(n,r) = C(n,n-r), \quad (1.13)$$

Opmerkingen

1. De betrekkingen (1.12) en (1.13) geven twee bekende eigenschappen weer van de driehoek van Pascal (zie [4, (1.56)]).
2. Uit de betekenis van $C(n,k)$ volgt direkt, dat men het binomium van Newton

$$(a+b)^n = \sum_{k=0}^n C(n,k) a^k b^{n-k}$$

ook kombinatorisch kan bewijzen.

1.3.3. Kombinaties met herhaling

In 1.3.1 en 1.3.2 beschouwden we combinaties uit een verzameling van verschillende objecten. In deze paragraaf laten we toe dat er ook niet van elkaar te onderscheiden objecten zijn (vgl. 1.2.3).

Stel we hebben n soorten verschillende objecten. Hieruit kiezen we r objecten, zodanig dat van één soort meer dan één object mag voorkomen in de selectie. Een dergelijke selectie heet r -kombinatie uit n met herhaling, omdat we dit ook kunnen opvatten als een selectie uit n verschillende objecten, zodanig dat een object vaker dan één keer gekozen mag worden.

Een voorbeeld is het gooien van 10 dobbelstenen tegelijkertijd, hetgeen een 10-kombinatie uit 6 oplevert.

We kunnen op eenvoudige wijze een uitdrukking afleiden voor het aantal combinaties van dit type. We nummeren de n (soorten) objecten met $1, 2, \dots, n$. Een bepaalde r -selectie geven we aan met een rij van de corresponderende getallen $(i, j, k, \dots, m)$, gerangschikt naar toenemende grootte. Zo stelt bijvoorbeeld de rij $(1, 1, 1, 3, 4, 5, 5, \dots)$ een selectie voor, waarbij het eerste object drie keer wordt gekozen, het tweede in het geheel niet, het derde één keer etc. We tellen nu bij het eerste getal in de rij 0 op, bij het tweede getal tellen we 1 op, bij het derde getal 2, ... en tenslotte bij het r^{de} getal $r - 1$. De rij $(i, j, k, \dots, m)$ gaat dus over in $(i, j+1, k+2, \dots, m+r-1)$. Elke r -selectie wordt zo op één-éénduidige wijze gekarakteriseerd door een r -kombinatie van r verschillende getallen uit de $n + r - 1$ getallen $1, 2, \dots, n + r - 1$. Dus is het totale aantal r -kombinaties uit n met herhaling gelijk aan

$$C(n + r - 1, r) = \binom{n + r - 1}{r}. \quad (1.14)$$

Voorbeeld 20

Schrijf alle 3-kombinaties uit 4 met herhaling op, waarbij $R := \{a, b, c, d\}$.

Oplossing

aaa aab aac aad abb abc abd acc acd add
bbb bbc bbd bcc bcd bdd ccc ccd cdd ddd

□

Uit de definitie en uit dit voorbeeld blijkt, dat een r -kombinatie uit n met $R := \{a_1, a_2, \dots, a_n\}$ ook beschouwd kan worden als een monotoon woord ter lengte r over het alfabet R .

Voorbeeld 21

Hoeveel r -kombinaties uit n met herhaling zijn er als elk soort object minstens één keer gebruikt moet worden?

Oplossing

Als $r < n$ dan is het antwoord 0. Voor $r \geq n$ nemen we eerst van elk soort één exemplaar, zodat aan de eis voldaan is. De resterende $r - n$ plaatsen in de combinatie mogen we naar vrije keuze opvullen, d.w.z. we moeten dan een $(r - n)$ -combinatie uit n met herhaling nemen. Het aantal mogelijkheden daartoe is gelijk aan

$$C(n + r - n - 1, r - n) = C(r - 1, r - n) = \binom{r - 1}{n - 1}. \quad \square$$

Voorbeeld 22

Er wordt een worp gedaan met 3 dobbelstenen. Hoeveel verschillende uitkomsten zijn er als de dobbelstenen

- van elkaar te onderscheiden zijn;
- niet van elkaar te onderscheiden zijn?

Oplossing

- Het totale aantal mogelijkheden is $6^3 = 216$;
- In dit geval moeten we een combinatie van 3 uit 6 met herhaling kiezen.

Dit kan op

$$\binom{6 + 3 - 1}{3} = 56 \text{ manieren.} \quad \square$$

Voorbeeld 23

Bewijs kombinatorisch dat

$$\binom{n + r - 1}{r} = \sum_{i=1}^r \binom{r - 1}{i - 1} \binom{n}{i}, \quad r \leq n, \quad (1.15)$$

en

$$\binom{n + r - 1}{r} = \sum_{i=1}^n \binom{r - 1}{i - 1} \binom{n}{i}, \quad r > n. \quad (1.16)$$

Oplossing

Het totale aantal r -kombinaties uit n met herhaling kan gesplitst worden in groepen, naar gelang het aantal verschillende voorwerpen dat in de combinatie voorkomt. Dit aantal kan zijn $1, 2, \dots$. Hoeveel zitten er in elke groep? Bekijk de groep die bestaat uit de combinaties waarin i verschillende soorten voorwerpen voorkomen. Op $\binom{n}{i}$ manieren kunnen die uit de n soorten gekozen worden. Ze moeten alle i minstens 1x gebruikt worden; dit kan op $\binom{r - 1}{i - 1}$ manieren zodat er $\binom{r - 1}{i - 1} \binom{n}{i}$ combinaties in de i^{de} groep zijn. M.b.v. de somregel vinden we zo betrekking (1.15).

De afleiding van (1.16) gaat analoog. $\square$

1.3.4. Kombinaties uit n met beperkt aantal herhalingen

Een veel voorkomend probleem is om het aantal r -kombinaties uit n met herhaling te tellen, met de extra voorwaarden dat er van elke soort slechts een beperkt aantal objecten gekozen mag worden. Een algemene uitdrukking voor dit aantal geven we nu niet. We zullen ons alleen bezig houden met de vraag naar het totale aantal combinaties in dit geval, zonder daarbij op de grootte van r te letten. In hoofdstuk 2 zullen we het algemene probleem oplossen.

Stel er zijn q_1 objecten van de eerste soort, q_2 objecten van de tweede soort ... en q_n objecten van de n^{de} soort. Het aantal mogelijkheden om 0 of meer objecten te kiezen, zonder daarbij op de volgorde te letten, is volgens de produktregel gelijk aan

$$(q_1 + 1)(q_2 + 1) \dots (q_n + 1), \quad (1.17)$$

Een bijzonder geval hiervan is wanneer geldt $q_1 = q_2 = \dots = q_n = 1$. Het aantal mogelijkheden wordt dan 2^n . Anderzijds is het aantal combinaties in dit geval volgens 1.3.2 gelijk aan $\sum_{r=0}^n \binom{n}{r}$, zodat opnieuw de gelijkheid uit voorbeeld 19 is aangetoond.

Voorbeeld 24

Hoeveel delers heeft 1400 ?

Oplossing

We ontbinden 1400 in priemfactoren:

$$1400 = 2^3 \times 5^2 \times 7.$$

Een deler van 1400 bevat hoogstens 3 factoren 2, hoogstens 2 factoren 5 en hoogstens 1 faktor 7, terwijl hij géén andere priemfactoren bevat. Omdat de volgorde der priemfactoren niet relevant is, is het gevraagde aantal gelijk aan $(3+1)(2+1)(1+1) = 24$. $\square$

1.4. Verdelingen

Een nauw met permutaties en combinaties verwante klasse van problemen is die van verdelingen.

1.4.1. Definitie

Een verdeling is een manier om een aantal verschillende of niet-verschillende objecten te verdelen over een aantal verschillende of niet te onderscheiden posities.

In plaats van posities spreekt men meestal over cellen.

GevalLEN waarin de cellen niet onderscheiden kunnen worden zullen in deze paragraaf niet aan de orde komen. Zijn de cellen wel van elkaar te onderscheiden dan kan men ze een nummer geven en spreekt men over genummerde cellen (soms verkeert men in de omgekeerde situatie en kan men de cellen juist daarom van elkaar onderscheiden, omdat ze een nummer hebben).

Voorts kan men nog met allerlei extra voorwaarden te maken hebben, zoals het aantal toegelaten objekten per cel, de volgorde binnen één cel, etc.

In de volgende twee paragrafen berekenen we het aantal verdelingen van r objecten over n verschillende cellen. We zullen de resultaten meedelen in de vorm van een aantal stellingen.

1.4.2. Verdelingen van verschillende objekten

Stelling (1.1)

Het aantal verdelingen van r verschillende objekten over n ($\geq r$) genummerde cellen, met hoogstens één object per cel is gelijk aan

$$P(n, r) = [n]_r.$$

Bewijs

Elke r -permutatie uit n correspondeert één-éénduidig met een verdeling: het i^{de} object ligt in de cel met nummer i . Hieruit volgt het gestelde onmiddellijk. $\square$

Laat men meer dan één object per cel toe, dan moet men weten of binnen één cel de volgorde waarin de objekten geplaatst worden van belang is of niet. In het eerste geval spreekt men van een geordende cel en in het tweede geval van een ongeordende cel.

Stelling (1.2)

Het aantal verdelingen van r verschillende objekten over n genummerde, ongeordende cellen is gelijk aan n^r .

Bewijs

In dit geval komt het erop neer dat men aan elk van de r objekten een cel toewijst. Per object heeft men de keuze uit n mogelijkheden. $\square$

Stelling (1.3)

Het aantal verdelingen van r verschillende objekten over n genummerde, geordende cellen is gelijk aan $[n]^r$.

Bewijs

Aan de r verschillende objecten voegt men nog toe $n-1$ objecten van dezelfde soort, nl. de "tussenwanden" van de cellen.

Elke permutatie van deze $r+n-1$ objecten correspondeert één-éénduidig met een verdeling van de gevraagde soort. Het aantal permutaties van $r+n-1$ objecten, waaronder $n-1$ van dezelfde soort, is gelijk aan $(r+n-1)!/(n-1)! = [n]^r$. □

Voorbeeld 25

Op hoeveel manieren kan men 7 **verschillende** vlaggen aan 5 masten hangen, als niet alle masten gebruikt hoeven te worden.

Oplossing

Wanneer er meer dan één vlag aan een mast moet worden bevestigd is de volgorde - van boven naar beneden - van belang. We hebben dus te maken met verdelingen van 7 verschillende objecten over 5 genummerde en geordende cellen. Het antwoord is

$$[5]^7 = 5 \times 6 \times 7 \times 8 \times 9 \times 10 \times 11.$$
 □

1.4.3. Verdelingen van niet-verschillende objecten

Stelling (1.4)

Het aantal verdelingen van r niet-verschillende objecten over n ($\geq r$) genummerde cellen, met hoogstens één object per cel is gelijk aan $C(n, r) = \binom{n}{r}$.

Bewijs

Het resultaat is onmiddellijk duidelijk als men zich realiseert dat men een r -kombinatie uit n cellen moet kiezen, om er de **niet** te onderscheiden objecten in op te bergen. □

Stelling (1.5)

Het aantal verdelingen van r niet-verschillende objecten over n genummerde cellen is gelijk aan $\binom{n+r-1}{r}$.

Bewijs

Ook dit resultaat is onmiddellijk duidelijk als men beseft dat een dergelijke verdeling equivalent is met een r -kombinatie uit n cellen met herhaling. □

Opmerking

Het heeft geen zin om in stelling (1.5) onderscheid te maken tussen geordende en ongeordende cellen, omdat de objecten binnen één cel toch niet van elkaar te onderscheiden zijn.

Voorbeeld 26

Op hoeveel manieren kan men p enen en q nullen achter elkaar zetten, zonder dat de rij twee direkt op elkaar volgende nullen bevat?

Oplossing

Plaats de p enen in een rij. Er zijn dan $p+1$ plaatsen (cellen) die elk hoogstens één nul mogen bevatten. Het antwoord luidt dus volgens stelling (1.4) $\binom{p+1}{q}$. □

Ga nu zelf na dat het aantal verdelingen van r niet-verschillende objecten over n genummerde cellen, met minstens één object per cel gegeven wordt door $C(r-1, n-1)$.

Voorbeeld 27

Men zendt 5 verschillende letters door een communicatiekanaal. In totaal moeten 15 spaties tussen de letters meegestuurd worden, met minstens 3 spaties tussen 2 letters. Op hoeveel manieren kan men de letters en spaties verzenden?

Oplossing

Men kan de letters op 5! manieren rangschikken. Bij elke rangschikking moet men vervolgens nog 15 (identieke) spaties verdelen over 4 verschillende cellen met minstens 3 spaties per cel. Men doet dit door eerst in elke cel 3 spaties te stoppen en daarna de 3 overgebleven spaties te verdelen over de 4 cellen. Dit kan op $C(4+3-1, 3)$ manieren. Het antwoord wordt dus $5! \cdot C(6,3) = 2400$. □

Voorbeeld 28

Een kongres heeft $2n+1$ zetels. Er zijn 3 politieke partijen, die meedoen aan de verkiezingen van kongresleden. Hoeveel zetelverdelingen zijn mogelijk, opdat een coalitie van elk tweetal partijen een meerderheid heeft?

Oplossing

We hebben hier te maken met 3 verschillende soorten objecten: leden van de 3 politieke partijen. Een zetelverdeling is een $(2n+1)$ -kombinatie hiervan, met herhaling. Of ook: een verdeling van $2n+1$ identieke objecten over 3 verschillende cellen.

Het totale aantal zetelverdelingen is dus $C(3 + 2n + 1 - 1, 2n + 1) = C(2n + 3, 2)$.

Sommige zetelverdelingen leiden niet tot meerderheidskoalities voor elk tweetal partijen, nl. die waarbij één partij minstens $n + 1$ zetels heeft. Het totale aantal verdelingen, waarbij dit het geval is (voor één specifieke partij) verkrijgt men door de resterende n zetels willekeurig te verdelen over de 3 partijen. Dit kan op $C(3 + n - 1, n) = C(n + 2, 2)$ manieren. Het gevraagde antwoord luidt

$$C(2n + 3, 2) - 3C(n + 2, 2) = \frac{1}{2}n(n + 1). \quad \square$$

Opmerking

Een andere (equivalente) interpretatie van een zetelverdeling is: een verdeling van $2n + 1$ niet-verschillende objecten over 3 genummerde cellen.

Ga zelf na, dat voor een kongres met $2n$ zetels het antwoord op dezelfde vraag zou luiden:

$$C(2n + 2, 2) - 3C(n + 2, 2) + 3 = \frac{1}{2}(n - 1)(n - 2).$$

Hoe worden de antwoorden als men in plaats van de gestelde voorwaarde zou eisen, dat géén der partijen de absolute meerderheid mag hebben?

2. RECURRENTE BETREKKINGEN EN GENERERENDE FUNKTIES

2.1. Inleiding

In dit hoofdstuk beschouwen we functies $f : \mathbb{N} \rightarrow \mathbb{R}$. De funktiewaarden $f(n)$, $n \in \mathbb{N}$, zullen we meestal noteren als a_n . Een recurrente betrekking of differentievergelijking is één of andere relatie tussen $a_n, a_{n-1}, \dots, a_0$, die geldig is voor alle $n \geq n_0$ ($n_0 \in \mathbb{N}$). Voorbeelden hiervan zijn:

$$a_n = n a_{n-1}, \quad n \geq 1;$$

$$3a_n + 5a_{n-1} + 6a_{n-2} = n^2 + 1, \quad n \geq 7;$$

$$a_n = a_{n-1} a_1 + a_{n-2} a_2 + \dots + a_1 a_{n-1}, \quad n \geq 1.$$

Om de berekening van de rij (a_n) uit een gegeven recurrente betrekking mogelijk te maken hebben we in het algemeen één of meer (begin)termen nodig, de zogenaamde randvoorwaarden.

In het voorbeeld $a_n = n a_{n-1}$ is kennis van de grootte van a_0 voldoende om alle andere a_n te kunnen bepalen. In het tweede voorbeeld zijn twee randvoorwaarden nodig, bijvoorbeeld a_5 en a_6 , om met de recurrente betrekking andere a_n te kunnen berekenen.

Let er bij dit voorbeeld nog op dat de betrekking geen enkele uitspraak doet over a_0 t/m a_4 .

Voorts valt er een zekere overeenkomst op met differentiaalvergelijkingen. In plaats van functies van een continue variabele x heeft men hier functies van een diskrete variabele n en in plaats van afgeleiden - die te maken hebben met verschillen van funktiewaarden in punten, die op infinitesimale afstand van elkaar liggen - vergelijkt men nu funktiewaarden in punten op eindige afstand van elkaar.

De terminologie is daarom ook vaak analoog. Zo is de betrekking in het eerste voorbeeld een differentievergelijking van de 1e orde en die in het tweede voorbeeld één van de 2e orde. Verder is zowel de eerste als de tweede betrekking lineair, omdat er geen produkten van funktiewaarden in voorkomen. De derde betrekking is een voorbeeld van een niet-lineaire differentievergelijking.

Veel problemen uit de kombinatoriek of meer algemeen uit de diskrete wiskunde geven aanleiding tot recurrente betrekkingen. Een standaardvoorbeeld is het zogenaamde **Fibonacci** probleem, dat we vanwege het grote belang van de resulterende vergelijking bij de studie van algoritmen, hier even in het kort bespreken.

We beginnen met een paar (bedoeld wordt een echt_paar) konijnen. Deze worden na één maand vruchtbaar, waarna ze na verloop van elke volgende maand precies twee jongen krijgen van verschillend geslacht, die vervolgens ook deel gaan nemen aan het vermenigvuldigingsproces. Als deze konijnen nooit dood gaan hoe groot is dan het aantal paren konijnen na n maanden?

Het gevraagde aantal F_n wordt opgebouwd door een verdubbeling van de bevolking die er 2 maanden daarvoor was ($2 F_{n-2}$) plus de nieuwe generatie die 1 maand geleden op de wereld gekomen is ($F_{n-1} - F_{n-2}$). De recurrente betrekking wordt dus

$$F_n = 2F_{n-2} + (F_{n-1} - F_{n-2}) = F_{n-1} + F_{n-2}. \quad (2.1)$$

De randvoorwaarden die men hier oplegt zijn betrekkelijk irrelevant. In bovenstaand probleem moet men nemen $F_0 = 1, F_1 = 1$. Een veel gebruikte konventie is ook $F_0 = 0, F_1 = 1$, hetgeen tot dezelfde rij getallen aanleiding geeft, afgezien van de beginterm. De getallen van de rij (F_n) heten Fibonacci getallen.

Voor het berekenen van getallen a_n , die gedefinieerd worden door een recurrente betrekking, zou men natuurlijk een programma kunnen schrijven. Vaak loont het echter de moeite om eerst oplossingen te zoeken van de recurrente betrekking, d.w.z. functies $f(n)$ die voor alle n aan de betrekking voldoen. In de volgende paragraaf zullen we een oplossingsmethode bespreken voor een veel voorkomend type vergelijking.

2.2. Lineaire recurrente betrekkingen met konstante coëfficiënten

De relatie (2.1) is een bijzonder geval van een recurrente betrekking van de gedaante

$$c_0 a_n + c_1 a_{n-1} + \dots + c_r a_{n-r} = h(n) \quad , \quad (2.2)$$

die geldig is voor alle $n \geq n_0$ met $n_0 \geq r \geq 0$.

De coëfficiënten $c_0, c_1, \dots, c_r$ zijn alle reële getallen, terwijl $h(n)$ een bekende functie van n is. We onderstellen nog dat c_0 en c_r beide ongelijk aan 0 zijn.

Een dergelijke relatie heet een lineaire recurrente betrekking met konstante coëfficiënten.

Als er r opeenvolgende termen, zeg $a_k, a_{k+1}, \dots, a_{k+r-1}$, bekend zijn, dan kan men met behulp van (2.2) recursief alle volgende a_n berekenen ($c_0 \neq 0$!) en ook alle er aan voorafgaande termen met $n \geq n_0$ ($c_r \neq 0$). Een oplossing van (2.2) is dus éénduidig bepaald door de waarden van r opeenvolgende termen: de randvoorwaarden. Vaak zijn de eerste r termen $a_{n_0-r}, \dots, a_{n_0-1}$, gegeven.

De recurrente betrekking is homogeen als $h(n) = 0$, voor alle $n \geq n_0$. Is dit niet het geval, dan heet de betrekking inhomogeen.

2.2.1. Algemene oplossing homogene recurrente betrekking

Beschouw

$$c_0 a_n + c_1 a_{n-1} + \dots + c_r a_{n-r} = 0, \quad n \geq n_0, \quad a_i \in \mathbb{R}. \quad (2.3)$$

Men kan eenvoudig verifiëren dat de volgende eigenschappen gelden (vergelijk weer de theorie van de lineaire differentiaalvergelijkingen met konstante coëfficiënten):

- a. Als de rij (a_n) - ofwel de functie f met $f(n) = a_n$ - een oplossing is van (2.3) dan is ook de rij (λa_n) met $\lambda \in \mathbb{R}$ een oplossing.
- b. Als de rijen (a_n) en (b_n) oplossingen zijn van (2.3) dan is ook de rij $(a_n + b_n)$ een oplossing.

Uit deze eigenschappen volgt, dat de oplossingen van (2.3) een vektorruimte vormen. Men kan bewijzen, dat de dimensie van deze ruimte gelijk is aan r .

Opmerking

Het is gemakkelijk in te zien dat de eigenschappen a en b ook geldig zijn voor willekeurige, homogene, lineaire, recurrente betrekkingen, d.w.z. als $c_0, \dots, c_r$ functies van n zijn. Ook dan vormen de oplossingen een vektorruimte van dimensie r .

Definitie

De functies $f_1, \dots, f_r$ heten lineair afhankelijk, wanneer er konstanten $\lambda_1, \dots, \lambda_r$ bestaan die niet alle 0 zijn, zodanig dat

$$\lambda_1 f_1(n) + \dots + \lambda_r f_r(n) = 0,$$

voor alle $n \in \mathbb{N}$. Kunnen zulke konstanten niet gevonden worden, dan zijn $f_1, \dots, f_r$ lineair onafhankelijk.

We zullen nu proberen r onafhankelijke oplossingen van (2.3) te vinden. Daartoe substitueren we $a_n = \alpha^n$, hetgeen oplevert

$$c_0 \alpha^r + c_1 \alpha^{r-1} + \dots + c_r = 0. \quad (2.4)$$

Deze vergelijking heet de karacteristieke vergelijking van (2.3) en de oplossingen $\alpha_1, \dots, \alpha_r$ heten de karacteristieke wortels.

Als $\alpha_1, \dots, \alpha_r$ alle verschillend zijn, dan kan men bewijzen dat de zogenaamde basisoplossingen $\alpha_1^n, \dots, \alpha_r^n$ lineair onafhankelijk zijn. De algemene oplossing van (2.3) is dan te schrijven als

$$a_n = \lambda_1 \alpha_1^n + \lambda_2 \alpha_2^n + \dots + \lambda_r \alpha_r^n, \lambda_i \in \mathbb{R}. \quad (2.5)$$

In het geval, dat de karakteristieke vergelijking meervoudige wortels heeft is er de volgende aanvulling op (2.5) te geven. Stel α_p is een k -voudige wortel van (2.4). De functies $\alpha_p^n, n\alpha_p^n, n^2\alpha_p^n, \dots, n^{k-1}\alpha_p^n$ vormen dan een onafhankelijk stelsel oplossingen, die bovendien een onafhankelijk stelsel vormen met de overige basisoplossingen. We geven van deze bewering geen bewijs.

In alle gevallen krijgen we dus een algemene oplossing van (2.3) die nog r willekeurige konstanten bevat, die eventueel nader bepaald kunnen worden met de randvoorwaarden. Het kan zijn, dat niet alle wortels van de karakteristieke vergelijking reëel zijn. Is men slechts geïnteresseerd in reële oplossingen van (2.3) dan moet men de reële en imaginaire gedeelten nemen van de complexe oplossingen.

Dat deze operatie weer tot oplossingen leidt van (2.3) vindt zijn oorzaak in het feit dat complexe wortels altijd in paren optreden, die elkaars toegevoegd complexe zijn (de koëfficiënten $c_0, \dots, c_r$ zijn reëel) en dat de eigenschappen a en b gelden.

We zullen de hierboven besproken methode illustreren met behulp van een aantal voorbeelden.

Voorbeeld 1

Bepaal het n^{de} Fibonacci getal en ga na hoe zich dit gedraagt voor grote n .
Neem $F_0 = 0$ en $F_1 = 1$.

Oplossing

De karakteristieke vergelijking van (2.1) luidt

$$\alpha^2 - \alpha - 1 = 0,$$

met als wortels $\alpha_1 = \frac{1}{2}(1 + \sqrt{5})$, $\alpha_2 = \frac{1}{2}(1 - \sqrt{5})$. Dus is de algemene oplossing van (2.1)

$$F_n = \lambda_1 \left(\frac{1 + \sqrt{5}}{2}\right)^n + \lambda_2 \left(\frac{1 - \sqrt{5}}{2}\right)^n.$$

De randvoorwaarden $F_0 = 0$, $F_1 = 1$ bepalen de konstanten λ_1 en λ_2 : $\lambda_1 = -\lambda_2 = \frac{1}{\sqrt{5}}$.

Voor grote waarden van n draagt de tweede term in de uitdrukking voor F_n weinig bij, omdat $|\frac{1}{2}(1 - \sqrt{5})| < 1 < \frac{1}{2}(1 + \sqrt{5})$. Dus geldt asymptotisch $F_n \sim \frac{1}{\sqrt{5}} (\frac{1 + \sqrt{5}}{2})^n$.

(Voor de preciese betekenis van het symbool $\sim$ wordt naar de appendix verwezen). □

Voorbeeld 2

Bereken de $n \times n$ - determinant

$$d_n = \begin{vmatrix} 1 & 1 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & . & . & . & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & . & . & . & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & . & . & . & 0 & 0 & 0 & 0 & 0 \\ . & . & . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & 0 & . & . & . & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 & 1 & 1 \end{vmatrix}$$

Oplossing

Door ontwikkeling naar de eerste rij van deze determinant vinden we de recurrente betrekking

$$d_n = d_{n-1} - d_{n-2}, \quad n \geq 3.$$

De karakteristieke vergelijking $\alpha^2 - \alpha + 1 = 0$ heeft als wortels

$$\alpha_1 = \frac{1}{2} + \frac{1}{2} i \sqrt{3} = e^{i\frac{\pi}{3}} \quad \text{en} \quad \alpha_2 = \frac{1}{2} - \frac{1}{2} i \sqrt{3} = e^{-i\frac{\pi}{3}}.$$

De algemene oplossing is

$$d_n = \lambda_1 e^{i\frac{n\pi}{3}} + \lambda_2 e^{-i\frac{n\pi}{3}} = (\lambda_1 + \lambda_2) \cos n\frac{\pi}{3} + i(\lambda_1 - \lambda_2) \sin n\frac{\pi}{3},$$

waarin λ_1 en λ_2 nog willekeurige komplekse getallen zijn. Daar uit de aard van het probleem volgt dat d_n reëel is, moeten we λ_1 en λ_2 zo kiezen dat $\mu_1 := \lambda_1 + \lambda_2$ en $\mu_2 := i(\lambda_1 - \lambda_2)$ beide reëel zijn. Ga na dat door geschikte keuze van λ_1 en λ_2 elk reëel paar getallen μ_1 en μ_2 te "realiseren" is.

De algemene reële oplossing luidt dus

$$d_n = \mu_1 \cos n\frac{\pi}{3} + \mu_2 \sin n\frac{\pi}{3}.$$

Uit de randvoorwaarden $d_1 = 1$ en $d_2 = 0$ vinden we tenslotte $\mu_1 = 1$, $\mu_2 = \frac{1}{\sqrt{3}}$, zodat

$$d_n = \cos n\frac{\pi}{3} + \frac{1}{\sqrt{3}} \sin n\frac{\pi}{3}.$$

Voorbeeld 3

Bereken de $n \times n$ - determinant

$$d_n = \begin{vmatrix} 2 & 1 & 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 & 0 & . & . & . & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 & . & . & . & 0 & 0 & 0 & 0 & 0 \\ . & . & . & . & . & . & . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & 0 & 0 & . & . & . & 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 & 1 & 2 \end{vmatrix} .$$

Oplossing

Uit de ontwikkeling van deze determinant volgt

$$d_n = 2d_{n-1} - d_{n-2} \quad (n \geq 3),$$

met als bijbehorende karakteristieke vergelijking $\alpha^2 - 2\alpha + 1 = 0$.

Deze vergelijking heeft een 2-voudige wortel 1. Dus geldt

$$d_n = (\lambda_1 n + \lambda_2) 1^n = \lambda_1 n + \lambda_2 .$$

Uit $d_1 = 2$, $d_2 = 3$ vinden we dan

$$d_n = n+1.$$

□

2.2.2. Algemene oplossing inhomogene recurrente betrekking

Analoog aan de situatie bij lineaire differentiaalvergelijkingen geldt hier de volgende eigenschap.

De algemene oplossing van een lineaire, recurrente betrekking is te schrijven als de som van een partikuliere oplossing van die betrekking en de algemene oplossing van de bijbehorende homogene recurrente betrekking.

Voor het vinden van een partikuliere oplossing bestaat geen algemene regel. Alleen in het geval dat $h(n)$ betrekkelijk eenvoudig is kunnen we op heuristische wijze een oplossing bepalen. Meer in het bijzonder kunnen we van de volgende regel gebruik maken.

Stel $h(n)$ in het rechter lid van (2.2) is van de gedaante $p(n)\alpha^n$, met $p(n)$ een polynoom in n van graad 1. Er bestaat dan een oplossing van (2.2) van de vorm $q(n)n^k \alpha^n$, waarbij $q(n)$ eveneens een polynoom is van graad 1, terwijl k gelijk is aan de multipliciteit van α als wortel van de karakteristieke

vergelijking van (2.2).

Voor het bepalen van de expliciete gedaante van de oplossing is het nog slechts nodig om de 1 + 1 onbekende koëfficiënten in $q(n)$ uit te rekenen. Dat kan men doen door te substitueren $a_n = q(n)n^k \alpha^n$ in (2.2) en door vervolgens de koëfficiënten van de verschillende machten van n links en rechts met elkaar te vergelijken.

Opmerking

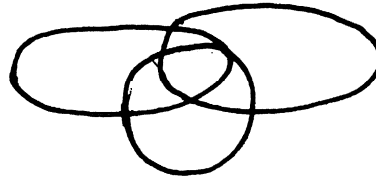
Vaak is α géén karakteristieke wortel. Dan is er een oplossing van (2.2) van de gedaante $q(n)\alpha^n$ ($k=0$), waarin $q(n)$ een polynoom is van dezelfde graad als $p(n)$. Ook komt het veel voor dat $h(n) = p(n) = p(n) \cdot 1^n$. In dat geval moet men erop verdacht zijn dat 1 een karakteristieke wortel zou kunnen zijn.

Voorbeeld 4

In het platte vlak zijn n ovalen getekend. (Een ovaal is een gesloten, convexe kromme, die zichzelf niet snijdt). Elk van deze ovalen snijdt elke andere ovaal in precies 2 punten.

Bovendien gaat geen enkel drietal ovalen door 1 punt.

In hoeveel gebieden wordt
het platte vlak door
deze n ovalen verdeeld?



Oplossing

Noem het gevraagde aantal a_n , $n \geq 1$.

De n^{de} ovaal snijdt elk van de voorafgaande $n-1$ ovalen in precies 2 punten en wordt dus door die snijpunten in $2(n-1)$ lijnstukken verdeeld. Elk van deze lijnstukken verdeelt één van de a_{n-1} gebieden, waarin het platte vlak door de voorafgaande ovalen is verdeeld, in 2 nieuwe gebieden. Hieruit volgt

$$a_n = a_{n-1} + 2(n-1), \quad n \geq 2, \quad (2.6)$$

met randvoorwaarde $a_1 = 2$.

De karakteristieke vergelijking is $\alpha - 1 = 0$ en de algemene oplossing van de homogene vergelijking luidt dus $a_n = A \cdot 1^n = A$.

Als partikuliere oplossing proberen we

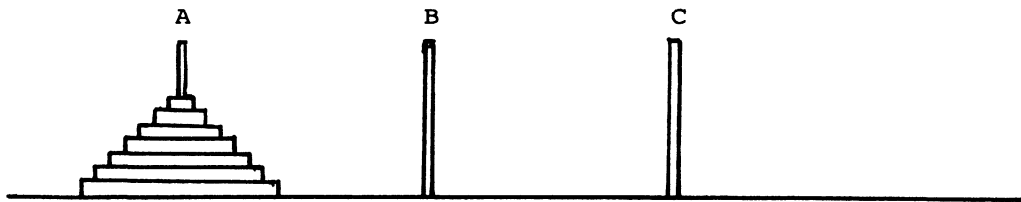
$$a_n = (pn + q)n^1 = pn^2 + qn.$$

Substitutie levert $p = 1$ en $q = -1$. De algemene oplossing is dus

$$a_n = n^2 - n + A, \text{ waaruit met de randvoorwaarde volgt } a_n = n^2 - n + 2. \quad \square$$

2.2.3. Het probleem van de torens van Hanoi

Gegeven zijn de drie palen A, B en C. Om paal A zitten n schijven met stralen $1, 2, \dots, n$ (in het midden van elke schijf zit een gat). De schijven zijn zo gestapeld, dat de i^{de} schijf vanaf de top straal i heeft.



Het probleem is nu, om de schijven naar paal C over te brengen, zodanig dat ze daar op dezelfde wijze gestapeld zijn als nu om A. Men mag slechts één schijf tegelijk verplaatsen van een paal naar een andere paal en het mag nooit gebeuren dat een schijf op een andere schijf ligt met kleinere straal. Paal B mag als hulppaal gebruikt worden.

Dit probleem is in [6] behandeld als een probleem, dat bij uitstek geschikt is, om rekursief te worden opgelost. Wij zullen ons hier bezig houden met de vraag hoeveel verplaatsingen er gedaan worden in een rekursieve algoritme. Onder een verplaatsing verstaan we het overbrengen van één ring van de ene naar een andere paal. Meer in het bijzonder zijn we erin geïnteresseerd hoe dit aantal zich gedraagt als functie van n , of anders gezegd, we onderzoeken de komplexiteit van een dergelijke algoritme.

Noem het gevraagde aantal a_n . We brengen eerst de bovenste $n-1$ ringen over naar paal B, door rekursie toe te passen van n naar $n-1$, waarbij C als hulppaal dient. Dit houdt a_{n-1} verplaatsingen in. Daarna brengen we de n^{de} ring naar paal C (één verplaatsing) en brengen tenslotte de $n-1$ ringen van B naar C, met A als hulppaal. Het probleem is dus teruggebracht tot een soortgelijk probleem, maar nu voor $n-1$. Wordt deze rekursie voortgezet, dan komt men uiteindelijk terecht bij $n = 1$. Hiervoor is het probleem op triviale wijze op te lossen en dus hebben we voor willekeurige n een rekursieve algoritme. Uit bovenstaande redenering volgt tevens

$$a_n = 2a_{n-1} + 1. \quad (2.7)$$

De oplossing van de homogene vergelijking is $A \cdot 2^n$ en een partikuliere oplossing is $a_n = -1$.

Met de randvoorwaarde $a_1 = 1$ krijgen we dan $a_n = 2^n - 1$.

Opmerking

Bovenstaand resultaat toont aan, dat het aantal verplaatsingen zeer snel toeneemt als n groter wordt (exponentiële groei). Om enig inzicht in die snelheid te krijgen ga men na, dat een gestaag doorwerkend persoon, die per seconde één ring kan verplaatsen en dat vanaf zijn geboorte dag en nacht volhoudt, toch niet meer dan 30 ringen in zijn leven kan verwerken. Een computer die per seconde 10^6 verplaatsingen kan simuleren zou voor 80 ringen ongeveer $3 \cdot 10^{10}$ jaar nodig hebben, een tijd in grootteorde gelijk aan de geschatte leeftijd van ons heelal!

2.3. Genererende funkties

2.3.1. Definitie gewone genererende funktie

In 2.2. en ook in een aantal paragrafen van hoofdstuk 1 hebben we ons bezig gehouden met telproblemen, waarbij nog een parameter - meestal n geheten - een rol speelde. De antwoorden a_n , $n \geq n_0$, werden opgevat als de elementen van een rij (a_n) . In deze paragraaf voeren we een nieuw hulpmiddel in om met deze getalrijen om te gaan. Voor het gemak nemen we $n_0 = 0$. Bij de rij $(a_0, a_1, a_2, \dots, a_n, \dots)$ definiëren we de funktie

$$GF(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n + \dots, \quad (2.8)$$

waarin x een reële variabele is.

We noemen (2.8) de gewone, genererende funktie of voortbrenger van de rij (a_n) . Het zal duidelijk zijn, dat verschillende rijen aanleiding geven tot verschillende genererende funkties. Men kan daarom zeggen dat $GF(x)$ de rij (a_n) representeert. Als de reeks $\sum_i a_i x^i$ sommeerbaar is stellen we $GF(x)$ gelijk aan deze som. In plaats van de algemene gedaante van de getallen a_n hoeven we dan "slechts" te onthouden de uitdrukking voor $GF(x)$. Willen we a_n weten voor een zekere waarde van n dan moeten we $GF(x)$ ontwikkelen naar machten van x , waarvoor meestal nog verschillende methoden beschikbaar zijn.

Het belangrijkste aspekt van genererende funkties is echter, dat allerlei operaties op rijen korresponderen met algebraïsche of analytische bewerkingen op de bijbehorende genererende funkties.

Als de rij (a_n) oneindig veel termen bevat, dan staat in het rechter lid van (2.8) een oneindige machtreeks en zouden er konvergentieproblemen kunnen ontstaan. Wij zullen ons daarover echter niet te zeer bekommeren. Dat wil zeggen, dat we aannemen dat er rondom 0 altijd wel een interval is aan te wijzen, waarop alle in aanmerking komende reeksen konvergeren. Een ander standpunt is, dat we hier met formele machtreeksen te maken hebben, waarin de variabele x slechts een formele betekenis heeft. Optellen en vermenigvuldigen van twee formele machtreeksen definiëren we dan op de "gebruikelijke" wijze resp. als

$$\begin{aligned} (a_0 + a_1x + a_2x^2 + \dots) + (b_0 + b_1x + b_2x^2 + \dots) = \\ (a_0 + b_0) + (a_1 + b_1)x + (a_2 + b_2)x^2 + \dots \end{aligned} \quad (2.9)$$

en

$$(a_0 + a_1x + a_2x^2 + \dots)(b_0 + b_1x + b_2x^2 + \dots) = a_0b_0 + (a_0b_1 + a_1b_0)x + (a_0b_2 + a_1b_1 + a_2b_0)x^2 + \dots \quad (2.10)$$

Deze laatste formule staat bekend als konvolutieprdukt of kortweg als konvolutie.

Het rechter lid van (2.10) is de gewone genererende funktie van de getallen $c_n = \sum_{k=0}^n a_{n-k} b_k$.

Ook andere manipulaties als termsgewijs differentiëren en integreren worden op de gewone manier uitgevoerd.

Opmerking

In plaats van $\sum_i a_i x^i$ beschouwt men algemener wel genererende funkties $\sum_i a_i \mu_i(x)$. De funkties μ_i moeten weer zodanig gekozen worden dat verschillende rijen niet dezelfde genererende funktie voortbrengen. Een voorbeeld is $\mu_i(x) = \cos(ix)$.

Als voorbeeld van een gewone, genererende funktie nemen we

$$GF(x) = (1+x)^n = \sum_{r=0}^n \binom{n}{r} x^r \quad (2.11)$$

waarvan de r^{de} coëfficiënt in de machtreeksontwikkeling het aantal r -kombinaties uit n voorstelt. (Let op de gewijzigde rol van n t.o.v. de vorige blz.!)

Verder is de genererende funktie voor het aantal r -kombinaties uit n met herhaling gelijk aan

$$GF(x) = \frac{1}{(1-x)^n} = \sum_{r=0}^{\infty} \binom{n+r-1}{r} x^r = \sum_{r=0}^{\infty} \binom{n-1+r}{n-1} x^r. \quad (2.12)$$

Toon zelf aan, dat de genererende funktie voor het aantal r -kombinaties uit n met herhaling (n is vast en bovendien $n \leq r$), zodanig dat elk van de n objecten minstens één keer vertegenwoordigd is, gelijk is aan

$$\frac{x^n}{(1-x)^n}.$$

2.3.2. Oplossen van recurrente betrekkingen met genererende funkties

Genererende funkties kan men gebruiken bij het oplossen van, niet noodzakelijk lineaire, recurrente betrekkingen. De methode komt erop neer, dat men de informatie, verschaft door de recurrente betrekking en de gegeven randvoorwaarden, omzet in informatie betreffende de genererende

funktie die de gezochte termen a_n , $n \geq n_0$ voortbrengt. Dit resulteert dan in een vergelijking (dit kan een algebraïsche vergelijking zijn, maar ook een differentiaal- of integraalvergelijking) die men mogelijk op kan lossen, waarna men door ontwikkeling naar machten van x ten slotte de gezochte term a_n vindt. Zonder op de algemeenheden van de methode verder in te gaan lichten we een en ander toe aan de hand van enkele voorbeelden.

Voorbeeld 5

Los de recurrente betrekking (2.6) op met behulp van genererende functies.

Oplossing

De betrekking $a_n = a_{n-1} + 2(n-1)$, $a_1 = 2$ was geldig voor $n \geq 2$. Wanneer we de a_n identificeren met de coëfficiënten van de funktie

$f(x) = \sum_{n=0}^{\infty} a_n x^n$ dan ligt de coëfficiënt a_0 hierin nog niet vast. We zijn vrij om a_0 elke waarde te geven die we maar willen. We kiezen $a_0 = 2$, waarmee bereikt wordt dat de recurrente betrekking ook voor $n = 1$ geldt.

Er volgt nu uit sommeren over n dat

$$\sum_{n=1}^{\infty} a_n x^n = \sum_{n=1}^{\infty} a_{n-1} x^n + 2 \sum_{n=1}^{\infty} (n-1) x^n.$$

Nu is

$$\sum_{n=1}^{\infty} (n-1) x^n = x^2 \sum_{n=1}^{\infty} (n-1) x^{n-2} =$$

$$x^2 \sum_{n=0}^{\infty} n x^{n-1} = x^2 \frac{d}{dx} \sum_{n=0}^{\infty} x^n =$$

$$x^2 \frac{d}{dx} \frac{1}{1-x} = \frac{x^2}{(1-x)^2}.$$

We vinden dus de vergelijking

$$f(x) - a_0 = xf(x) + \frac{2x^2}{(1-x)^2}$$

ofwel

$$(1-x)f(x) = a_0 + \frac{2x^2}{(1-x)^2} = 2 + \frac{2x^2}{(1-x)^2},$$

zodat

$$f(x) = \frac{2}{1-x} + \frac{2x^2}{(1-x)^3} = 2 \sum_{n=0}^{\infty} x^n + 2x^2 \sum_{n=0}^{\infty} \binom{n+2}{2} x^n =$$

$$2 \sum_{n=0}^{\infty} x^n + 2 \sum_{n=2}^{\infty} \binom{n}{2} x^n = 2 \sum_{n=0}^{\infty} x^n + 2 \sum_{n=0}^{\infty} \binom{n}{2} x^n = \sum_{n=0}^{\infty} (n^2 - n + 2) x^n.$$

Dus is het resultaat $a_n = n^2 - n + 2$. $\square$

Los zelf de recurrente betrekking voor Fibonacci getallen uit voorbeeld 1 op, met behulp van genererende funkties.

Voorbeeld 6

Los op de recurrente betrekking

$$n a_n + (n - 1)a_{n-1} = 0, \quad n \geq 2,$$

$$a_1 = 1.$$

Oplossing

De betrekking is lineair, maar de koëfficiënten zijn niet konstant, zodat we de methode van 2.2.1 niet kunnen toepassen.

We voeren in de funktie $f(x) = \sum_{n=0}^{\infty} a_n x^n$.

Allereerst merken we op, dat a_0 niet zo gekozen kan worden dat er aan de gegeven relatie wordt voldaan voor $n = 1$. Gemakshalve kiezen we $a_0 = 0$.

Er volgt nu via sommatie over n , termsgewijze differentiatie en delen door x :

$$\sum_{n=2}^{\infty} n a_n x^n + \sum_{n=2}^{\infty} (n - 1)a_{n-1} x^n = 0,$$

$$\sum_{n=2}^{\infty} n a_n x^n + x \sum_{n=1}^{\infty} n a_n x^{n-1} = 0,$$

$$(1 + x) x \sum_{n=1}^{\infty} n a_n x^{n-1} - x = 0,$$

$$(1 + x) \frac{d}{dx} \sum_{n=1}^{\infty} a_n x^n - 1 = 0.$$

De funktie f moet dus aan de volgende differentiaalvergelijking voldoen

$$(1 + x)f'(x) = 1,$$

waaruit we met $f(0) = 0$ vinden $f(x) = \ln(1 + x)$. Dit levert met behulp van de bekende reeksontwikkeling voor $\ln(1 + x)$ in het punt 0 het uiteindelijke resultaat $a_n = \frac{(-1)^{n+1}}{n}$. $\square$

Opmerkingen

1. In het kader van genererende funkties verdient het aanbeveling om ook de andere "bekende" Taylorreeksontwikkelingen in gedachten te houden. Zie hiervoor [4, 4.8.8].
2. De betrekking uit voorbeeld 6 is ook eenvoudig op te lossen via de substitutie $b_n = n a_n$.

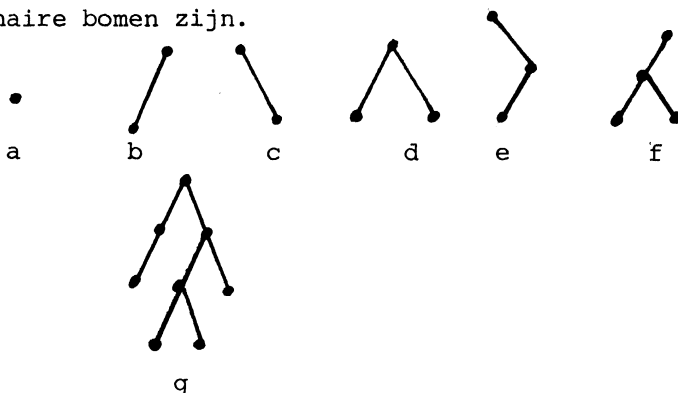
2.3.3. Aantal binaire bomen met n knopen

Met genererende funkties slaagt men er soms ook in, om niet-lineaire recurrente betrekkingen op te lossen. Hiervan geven we een tweetal voorbeelden. Ter voorbereiding daarvan voeren we eerst het begrip binaire boom in. We doen dat door middel van een rekursieve definitie.

Definitie

Een binaire boom T is of leeg of het is een knoop, waaraan twee binaire (sub) bomen T_l en T_r hangen, die respectievelijk linker en rechter subboom genoemd worden.

Met deze definitie vindt men gemakkelijk, dat de volgende structuren binaire bomen zijn.



In bovenstaande figuren stellen de stippen de knopen voor. De verbindingslijnen tussen de knopen heten takken; hiermee zijn de subbomen aan de knopen bevestigd.

Merk op, dat **b** en **c** twee verschillende binaire bomen voorstellen. In **b** is de rechter subboom leeg en in **c** de linker subboom. De boom in **a** is een boom met één knoop en niet de lege boom !

We vragen ons nu af hoe groot het totale aantal binaire bomen is met n knopen. Noem dit aantal a_n . Uit de definitie volgt onmiddellijk $a_0 = 1$, terwijl verder door telling blijkt $a_1 = 1$, $a_2 = 2$, $a_3 = 5$.

Gaat men uit van de wortel van de binaire boom (d.i. de beginknoop waaruit men de hele boom kan opbouwen door steeds nieuwe subbomen aan te hangen), dan kan men hieraan een linker subboom met l knopen bevestigen en een rechter subboom met $n - l - 1$ knopen ($0 \leq l \leq n - 1$). De recurrente betrekking is daarom

$$a_n = \sum_{l=0}^{n-1} a_l a_{n-l-1}, \quad n \geq 1. \quad (2.13)$$

Deze betrekking is niet-lineair en bovendien is het aantal termen dat erin een rol speelt afhankelijk van n ! Toch is hier een eenvoudige oplossing.

Het rechterlid van (2.13) is namelijk de $(n-1)$ -ste coëfficiënt van $\{f(x)\}^2$ als $f(x) = \sum_{k=0}^{\infty} a_k x^k$ en is een bijzonder geval van konvolutie (zie (2.10)).

Vermenigvuldigen we (2.13) met x^n en nemen we vervolgens de som voor $n = 1, 2, \dots$ dan krijgen we links $f(x)-1$ en rechts $x\{f(x)\}^2$ en dus voldoet $f(x)$ aan $x\{f(x)\}^2 - f(x) + 1 = 0$. Hiervan is de oplossing

$$f(x) = \frac{1}{2x} (1 \pm \sqrt{1 - 4x}). \quad (2.14)$$

De binomiaalontwikkeling van $\sqrt{1 - 4x}$ geeft

$$\begin{aligned} (1 - 4x)^{\frac{1}{2}} &= \sum_{k=0}^{\infty} \binom{\frac{1}{2}}{k} (-4x)^k \\ &= 1 - \frac{1}{2}(4x) + \frac{\frac{1}{2}(-\frac{1}{2})}{2!} (4x)^2 - \frac{\frac{1}{2}(-\frac{1}{2})(-\frac{3}{2})}{3!} (4x)^3 + \dots \\ &= 1 - \sum_{k=1}^{\infty} \frac{2}{k} \binom{2k-2}{k-1} x^k. \end{aligned}$$

Omdat de getallen a_n niet negatief mogen zijn vervalt de oplossing met het + teken in (2.14) en we vinden dus

$$\begin{aligned} f(x) &= \frac{1}{2x} (1 - 1 + \sum_{k=1}^{\infty} \frac{2}{k} \binom{2k-2}{k-1} x^k) = \sum_{n=0}^{\infty} \frac{1}{n+1} \binom{2n}{n} x^n, \\ a_n &= \frac{1}{n+1} \binom{2n}{n}, \quad n \geq 0. \end{aligned} \quad (2.15)$$

Deze getallen heten wel de getallen van Catalan. Door de formule van Stirling (zie appendix) toe te passen vinden we

$$a_n \sim \frac{4^n}{n \sqrt{\pi n}}, \quad (2.16)$$

hetgeen enig inzicht verschaft in het gedrag van a_n voor grote n .

Een probleem dat aanleiding geeft tot een verwante recurrente betrekking is het volgende.

Voorbeeld 7

Bepaal het aantal manieren, waarop men haakjes kan aanbrengen in de uitdrukking

$$w_1 + w_2 + \dots + w_{n-1} + w_n,$$

zodanig dat elke keer slechts twee termen opgeteld moeten worden.

Zo kan $w_1 + w_2 + w_3 + w_4$ op de volgende manieren van haakjes worden voorzien:

$$((w_1 + w_2) + (w_3 + w_4)), (((w_1 + w_2) + w_3) + w_4), ((w_1 + (w_2 + w_3)) + w_4), \\ (w_1 + ((w_2 + w_3) + w_4)), (w_1 + (w_2 + (w_3 + w_4))).$$

Oplossing

Zij a_n het gevraagde aantal. Splits $w_1 + w_2 + \dots + w_n$ in $w_1 + w_2 + \dots + w_l$ en $w_{l+1} + w_{l+2} + \dots + w_n$ ($1 \leq l \leq n-1$). Er zijn a_l manieren om de eerste som van haakjes te voorzien en a_{n-l} manieren voor de tweede. Dit geeft aanleiding tot de betrekking

$$a_n = \sum_{l=1}^{n-1} a_l a_{n-l}, \quad n \geq 2. \quad (2.17)$$

Hierin is per definitie $a_1 = 1$. (Vergelijk (2.17) met (2.13)).

Definieren we $a_0 = 0$ dan kunnen we in plaats van (2.17) schrijven

$$a_n = \sum_{l=0}^n a_l a_{n-l}, \quad n \geq 2$$

hetgeen met $f(x) = \sum_{n=0}^{\infty} a_n x^n$ om te zetten is in

$$\{f(x)\}^2 - f(x) + x = 0.$$

De uiteindelijke oplossing hiervan luidt

$$f(x) = \frac{1}{2} - \frac{1}{2} \sqrt{1 - 4x} = \sum_{n=1}^{\infty} \frac{1}{n} \binom{2n-2}{n-1} x^n, \\ a_n = \frac{1}{n} \binom{2n-2}{n-1}, \quad n \geq 1. \quad (2.18)$$

Probeer zelf (2.18) af te leiden door het aanbrengen van haakjes in $w_1 + w_2 + \dots + w_n$ te interpreteren als de konstruktie van een binaire boom.

2.3.4. Genererende funkties voor kombinaties

In 1.3.3 voerden we in het begrip herhalingskombinatie van r uit n ofwel een r -kombinatie uit n met herhalingen. Daarbij werd ervan uitgegaan dat elk van de n objekten ongelimiteerd vaak gekozen mocht worden.

In deze paragraaf zullen we een algemener probleem beschouwen, waarbij het aantal objekten van de i^{de} soort ($1 \leq i \leq n$) gekozen moet worden uit een verzameling voorgeschreven waarden.

Een zekere r -kombinatie duiden we aan met α . We definiëren de frequentie van α als de vektor

$$f := (f_1, f_2, \dots, f_n),$$

waarin f_i het aantal objecten van de i^{de} soort aangeeft, dat in α voorkomt.

Uit de definitie volgt dus

$$f_1 + f_2 + \dots + f_n = r.$$

De voorgeschreven of toegelaten waarden van f worden met behulp van gegeven verzamelingen $F_i \subset \mathbb{N}$, $i \in \{1, 2, \dots, n\}$, vastgelegd door de eis

$$f \in F := F_1 \times F_2 \times \dots \times F_n.$$

De produktverzameling F noemt men een frequentieverdeling.

We lichten bovenstaande begrippen toe met het volgende voorbeeld.

Voorbeeld 8

Schrijf uit voor $r = 2$ en voor $r = 3$ alle r -kombinaties uit de 3 objecten van de verzameling $R := \{a, b, c\}$. De toegelaten frequenties worden bepaald door $F_1 = \{0, 1\}$, $F_2 = \{0, 1, 2\}$ en $F_3 = \{0, 1\}$.

Oplossing

Voor $r = 2$ hebben we

$$ab, ac, b^2, bc,$$

en voor $r = 3$

$$ab^2, abc, b^2c.$$

□

Definitie

Het aantal herhalingskombinaties van r uit n met $f \in F$, $f_1 + f_2 + \dots + f_n = r$ noemen we $C(n, r; F)$.

Opmerkingen

1. Als $F_i = \{f_i\}$, $1 \leq i \leq n$, dan noteren we

$$C(n, r; F) \text{ als } C(n, r; f) \text{ met } f = (f_1, f_2, \dots, f_n).$$

Hieruit volgt direkt

$$C(n, r; f) = 1, \text{ als } f_1 + f_2 + \dots + f_n = r,$$

$$C(n, r; F) = \sum_{f \in F} C(n, r; f).$$

Waarbij we definiëren $C(n, r; f) = 0$ als $f_1 + f_2 + \dots + f_n \neq r$.

2. Voor $F_i = \{0, 1\}$, $1 \leq i \leq n$, geldt

$$C(n, r; F) = C(n, r) = \binom{n}{r}.$$

Hieruit blijkt dat de getallen $C(n, r; F)$ opgevat kunnen worden als een generalisatie van gewone binomiaalkoefficiënten.

3. Een herhalingskombinatie van r uit de n objecten van $R := \{a_1, a_2, \dots, a_n\}$ met $f \in F$ kan men één-éénduidig interpreteren als een monotoon woord

$$\overset{f_1}{a_1} \overset{f_2}{a_2} \dots \overset{f_n}{a_n}$$

van de lengte r ($f_1 + f_2 + \dots + f_n = r$) over het alfabet R . Het getal f_i is de frequentie van de letter a_i in het betreffende woord. (Vergelijk voorbeeld 8) Het getal $C(n, r; F)$ geeft dus het totale aantal van zulke woorden.

4. Een andere interpretatie van $C(n, r; F)$ is dat dit getal het totale aantal oplossingen $f = (f_1, f_2, \dots, f_n) \in F$ geeft van de vergelijking

$$f_1 + f_2 + \dots + f_n = r. \quad (2.19)$$

Een krachtig hulpmiddel bij het bepalen van de getallen $C(n, r; F)$ voor een gegeven frequentieverdeling wordt verschaft door de volgende stelling.

Stelling (2.1)

Als F een gegeven frequentieverdeling is, dan is de gewone genererende funktie voor de getallen $C(n, r; F)$ gelijk aan

$$GF(x) = \prod_{i=1}^n \left(\sum_{f_i \in F_i} x^{f_i} \right). \quad (2.20)$$

Bewijs

Bij uitwerking van het rechterlid van (2.20) blijkt, dat de koefficiënt van x^r gelijk is aan het aantal oplossingen van

$$f_1 + f_2 + \dots + f_n = r,$$

met $f = (f_1, f_2, \dots, f_n) \in F$.

Op grond van opmerking 4 hierboven volgt dat dit het getal $C(n, r; F)$ is. $\square$

We passen deze stelling nu toe op een aantal voorbeelden.

Voorbeeld 9

Iemand heeft 3 postzegels van 5 cent, 1 postzegel van 10 cent en 2 postzegels van 20 cent. Hoeveel verschillende frankeringen van 40 cent kan hij hiermee samenstellen, als de volgorde waarin de zegels op de brief worden geplakt er niet toe doet?

Oplossing

Definieer a_r als het aantal verschillende frankeringen van r cent. We hebben drie verschillende objecten, namelijk centen vertegenwoordigd door 5-cents postzegels, door 10-cents postzegels en door 20-cents postzegels. Als we een frankering van r cent willen hebben betekent dit dat we een herhalingskombinatie van r uit 3 objecten willen hebben, waarbij de toegelaten frequenties worden gegeven door $F_1 = \{0, 5, 10, 15\}$, $F_2 = \{0, 10\}$, $F_3 = \{0, 20, 40\}$. Volgens stelling (2.1) is de genererende functie

$$GF(x) = (1 + x^5 + x^{10} + x^{15})(1 + x^{10})(1 + x^{20} + x^{40}).$$

Bij uitwerking van het rechterlid blijkt $a_{40} = 2$. $\square$

Voorbeeld 10

Een gewichtendoos bevat gewichten van 1, 1, 2, 5, 10, 10, 20 en 50 gram. Men beschikt verder over een balans. Als men op slechts één schaal gewichten mag zetten, op hoeveel manieren kan men dan een gewicht van r gram samenstellen?

Men kan hierbij nog uitgaan van twee standpunten, namelijk dat wegingen met verschillende gewichten die even zwaar zijn als

- verschillend worden beschouwd,
- als niet-verschillend worden beschouwd.

Oplossing

Gevraagd wordt naar het aantal r -kombinaties uit 8 respektievelijk 6 met herhaling, met frequenties

- $F_1 = F_2 = \{0, 1\}$, $F_3 = \{0, 2\}$, $F_4 = \{0, 5\}$, $F_5 = F_6 = \{0, 10\}$,
 $F_7 = \{0, 20\}$, $F_8 = \{0, 50\}$
- $F_1 = \{0, 1, 2\}$, $F_2 = \{0, 2\}$, $F_3 = \{0, 5\}$, $F_4 = \{0, 10, 20\}$,
 $F_5 = \{0, 20\}$, $F_6 = \{0, 50\}$.

De genererende functies worden

$$GF(x) = (1 + x)^2(1 + x^2) \dots (1 + x^{50})$$

respektievelijk

$$GF(x) = (1 + x + x^2)(1 + x^2) \dots (1 + x^{50}). \quad \square$$

Opmerking

De situatie, waarin men twee schalen mag gebruiken is lastiger.

Onder voorwaarde a kan men nemen

$$F_1 = F_2 = \{-1, 0, 1\} \dots, F_8 = \{-50, 0, 50\}$$

Ga zelf na dat onder voorwaarde b de aanname

$$F_1 = \{-2, -1, 0, 1, 2\}, \dots, F_6 = \{-50, 0, 50\}$$

tot verkeerde resultaten leidt.

Voorbeeld 11

Bepaal het totale aantal r -kombinaties uit n , waarbij een ongelimiteerd aantal herhalingen is toegestaan.

Oplossing

De frequentieverdeling is in dit geval

$$F = F_1 \times F_2 \times \dots \times F_n,$$

met

$$F_i = \{0, 1, 2, \dots\}, \quad 1 \leq i \leq n.$$

Volgens stelling (2.1) is

$$\begin{aligned} GF(x) &= (1 + x + x^2 + \dots)^n = \frac{1}{(1 - x)^n} = \\ &= \sum_{k=0}^{\infty} \binom{n + k - 1}{k} x^k. \end{aligned}$$

$$\text{Dus vinden we weer } a_r = \binom{n + r - 1}{r}. \quad \square$$

Voorbeeld 12

Vijf personen gooien ieder éénmaal met een dobbelsteen. Hoe groot is het aantal manieren, waarop het totaal geworpen ogenaantal 18 kan bedragen?

Oplossing

Er zijn 5 verschillende soorten objecten, namelijk de "ogen" gegooid door de 5 verschillende personen. Laat a_r het aantal manieren voorstellen om het ogenaantal r te werpen. De genererende functie hiervoor luidt dan

$$\begin{aligned} GF(x) &= (x + x^2 + \dots + x^6)^5 = x^5 (1 - x^6)^5 (1 - x)^{-5} = \\ &= x^5 (1 - 5x^6 + 10x^{12} - \dots) \left(1 + \frac{5}{1!}x + \frac{5 \cdot 6}{2!}x^2 + \dots\right). \end{aligned}$$

De koëfficiënt van x^{18} is gelijk aan

$$\frac{(17!)}{4! 13!} - 5 \frac{(11)!}{4! 7!} + 10 \frac{5!}{4! 1!} = 780. \quad \square$$

2.3.5. Genererende funkties voor permutaties

We zullen nu permutaties behandelen op een soortgelijke wijze als in 2.3.4 kombinaties. Daartoe voeren we naast de in (2.8) gedefinieerde gewone genererende funktie in de funktie

$$EF(x) = a_0 + a_1 \frac{x}{1!} + a_2 \frac{x^2}{2!} + \dots + a_n \frac{x^n}{n!} + \dots, \quad (2.21)$$

die de exponentiële genererende funktie van de rij $(a_0, a_1, a_2, \dots, a_n, \dots)$ genoemd wordt.

Ten aanzien van het al of niet konvergeren van de reeks in het rechterlid van (2.21) gelden dezelfde opmerkingen als die welke in 2.3.1 gemaakt zijn voor de gewone genererende funktie.

In plaats van (2.9) en (2.10) heeft men nu

$$\begin{aligned} (a_0 + a_1 \frac{x}{1!} + a_2 \frac{x^2}{2!} + \dots) + (b_0 + b_1 \frac{x}{1!} + b_2 \frac{x^2}{2!} + \dots) = \\ (a_0 + b_0) + (a_1 + b_1) \frac{x}{1!} + (a_2 + b_2) \frac{x^2}{2!} + \dots \end{aligned} \quad (2.22)$$

en

$$\begin{aligned} (a_0 + a_1 \frac{x}{1!} + a_2 \frac{x^2}{2!} + \dots) (b_0 + b_1 \frac{x}{1!} + b_2 \frac{x^2}{2!} + \dots) = \\ a_0 b_0 + (a_1 b_0 + a_0 b_1) \frac{x}{1!} + (\frac{2!}{2! 0!} a_2 b_0 + \frac{2!}{1! 1!} a_1 b_1 + \frac{2!}{0! 2!} a_0 b_2) \frac{x^2}{2!} + \dots \end{aligned} \quad (2.23)$$

Opmerking

De funktie in het rechter lid van (2.23) is de exponentiële genererende funktie van de getallen

$$c_n := \sum_{k=0}^n \binom{n}{k} a_{n-k} b_k, \quad n \geq 0. \quad (2.24)$$

Exponentiële genererende funkties spelen een rol bij het bepalen van aantallen permutaties. In 1.2.3 is gesproken over permutaties van objecten die niet allemaal verschillend zijn.

Daarbij gingen we ervan uit dat er van elke soort een vast aantal objecten vertegenwoordigd was in de permutatie. Die eis laten we hier varen. Een herhalingspermutatie van r uit n ofwel een r -permutatie uit n met herhaling is een geordende selectie van r objecten uit een kollektie die n soorten bevat. We laten daarbij in het midden hoeveel exemplaren van de i^{de} soort ($1 \leq i \leq n$) voorkomen in de permutatie. Analooq als bij combinaties definiëren we bij een gegeven herhalingspermutatie α van r uit n de frequentie

$$f := (f_1, f_2, \dots, f_n).$$

Hierin is f_i weer het aantal objecten van de i^{de} soort dat in α voorkomt. Er geldt dus ook weer

$$f_1 + f_2 + \dots + f_n = r.$$

Bovendien is f weer een vektor uit de frequentieverdeling $F := \prod_{i=1}^n F_i$, $f_i \in F_i$, $F_i \subset \mathbb{N}$ voor $i \in \{1, 2, \dots, n\}$.

Definitie

Het aantal herhalingspermutaties van r uit n met $f \in F$, $f_1 + f_2 + \dots + f_n = r$ noemen we $P(n, r; F)$.

Voorbeeld 13

Schrijf uit voor $r = 2$ en voor $r = 3$ alle r -permutaties uit de 3 objecten van de verzameling $R := \{a, b, c\}$. De toegelaten frequenties worden bepaald door $F_1 = \{0, 1\}$, $F_2 = \{0, 1, 2\}$ en $F_3 = \{0, 1\}$.

Oplossing

Voor $r = 2$ hebben we

$ab, ba, ac, ca, b^2, bc, cb,$

en voor $r = 3$

$ab^2, b^2a, bab, abc, acb, bac, bca, cab, cba, b^2c, cb^2, bcb.$

□

Opmerkingen

1. Als $F_i = \{f_i\}$, $1 \leq i \leq n$, dan noteren we $P(n, r; F)$ als $P(n, r; f)$ met $f = (f_1, f_2, \dots, f_n)$. Hieruit en uit (1.6) volgt

$$P(n,r;f) = \frac{r!}{f_1! f_2! \dots f_n!}$$

als $f_1 + f_2 + \dots + f_n = r$

Bovendien definiëren we $P(n,r;f) = 0$ als $f_1 + f_2 + \dots + f_n \neq r$.

2. Een herhalingspermutatie van r uit de n objecten van

$R := \{a_1, a_2, \dots, a_n\}$ met $f \in F$ kan men één-éénduidig interpreteren als een woord van de lengte r , waarin de letter a_i met frequentie f_i optreedt ($1 \leq i \leq n$).

(Vergelijk voorbeeld 13). Het getal $P(n,r;F)$ geeft het totale aantal van zulke woorden.

3. Voor $F_i = \{0,1\}$, $1 \leq i \leq n$ geldt $P(n,r;F) = P(n,r) = [n]_r$.

De exponentiële genererende funktie van deze getallen is

$$\begin{aligned} EF(x) &= 1 + [n]_1 \frac{x}{1!} + [n]_2 \frac{x^2}{2!} + \dots \\ &= 1 + \frac{n!}{(n-1)!1!} x + \frac{n!}{(n-2)!2!} x^2 + \dots \\ &= (1+x)^n. \end{aligned}$$

(Let op het verschil met $GF(x) = (1+x)^n$).

4. Als de notatie $|f| = r$ betekent $f_1 + f_2 + \dots + f_n = r$, dan volgt uit opmerking 1, met de somregel, dat

$$\begin{aligned} P(n,r;F) &= \sum_{f \in F} P(n,r;f) \\ &= \sum_{f \in F \wedge |f| = r} \frac{r!}{f_1! f_2! \dots f_n!}. \end{aligned}$$

Om de getallen $P(n,r;F)$ te bepalen voor een gegeven frequentieverdeling kan men gebruik maken van de volgende stelling.

Stelling (2.2)

Als F een gegeven frequentieverdeling is, dan is de exponentiële genererende funktie voor de getallen $P(n,r;F)$ gelijk aan

$$EF(x) = \prod_{i=1}^n \left(\sum_{f_i \in F_i} \frac{x^{f_i}}{f_i!} \right).$$

Bewijs

Toepassen van opmerking 4 levert de volgende afleiding

$$\begin{aligned}
 EF(x) &= \sum_{r=0}^{\infty} P(n, r; F) \frac{x^r}{r!} \\
 &= \sum_{r=0}^{\infty} \sum_{f \in F} P(n, r; f) \frac{x^r}{r!} \\
 &= \sum_{r=0}^{\infty} \sum_{f \in F \wedge |f|=r} \frac{r!}{f_1! f_2! \dots f_n!} \frac{x^r}{r!} \\
 &= \sum_{f \in F} \frac{x^{f_1}}{f_1!} \frac{x^{f_2}}{f_2!} \dots \frac{x^{f_n}}{f_n!} \\
 &= \prod_{i=1}^n \left(\sum_{f_i \in F_i} \frac{x^{f_i}}{f_i!} \right).
 \end{aligned}$$

□

Voorbeeld 14

Bepaal het totale aantal woorden van lengte r over het alfabet

$$R := \{a_1, a_2, \dots, a_n\}.$$

Oplossing

We passen opmerking 2 toe met $F_i = \mathbb{N}$, voor $i \in \{1, 2, \dots, n\}$, alsmede stelling (2.2), hetgeen levert

$$EF(x) = \left(1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots\right)^n = e^{nx} = \sum_{k=0}^{\infty} \frac{n^k x^k}{k!}.$$

Het gevraagde aantal is gelijk aan

$$b_r = n^r,$$

zoals we op grond van veel eenvoudiger overwegingen ook wel hadden kunnen afleiden. □

Leid nu zelf de genererende funktie uit opmerking 3 af.

Voorbeeld 15

Bepaal het aantal woorden van lengte r over het alfabet

$R := \{a_1, a_2, \dots, a_n\}$, waarin elke letter minstens één keer voorkomt.

Oplossing

Stel b_r is het gevraagde aantal. De genererende funktie hiervoor is volgens stelling (2.2)

$$\begin{aligned} EF(x) &= \left(\frac{x}{1!} + \frac{x^2}{2!} + \dots \right)^n = (e^x - 1)^n \\ &= \sum_{i=0}^n \binom{n}{i} (-1)^{n-i} e^{ix} \\ &= \sum_{i=0}^n \binom{n}{i} (-1)^{n-i} \sum_{k=0}^{\infty} \frac{i^k x^k}{k!} \\ &= \sum_{k=0}^{\infty} \sum_{i=0}^n \binom{n}{i} (-1)^{n-i} i^k \frac{x^k}{k!} . \end{aligned}$$

$$\text{Dus } b_r = \sum_{i=0}^n (-1)^{n-i} \binom{n}{i} i^r .$$

□

Opmerking

Men kan bovenstaand aantal opvatten als het aantal verdelingen van r verschillende objecten over n genummerde, ongeordende cellen, zodanig dat geén cel leeg is (zie ook 1.4.2). De getallen

$$S(r, n) = \frac{1}{n!} \sum_{i=0}^n (-1)^{n-i} \binom{n}{i} i^r = \frac{1}{n!} \sum_{j=0}^n (-1)^j \binom{n}{j} (n-j)^r$$

heten wel de getallen van Stirling van de tweede soort.

Uit een en ander volgt nog, dat het aantal verdelingen van r verschillende objecten over n niet-genummerde (niet te onderscheiden) en niet-geordende cellen, met geén cel leeg, gelijk is aan $S(r, n)$.

Voorbeeld 16

Bepaal het aantal getallenrijen van de lengte r bestaande uit de cijfers 0, 1, 2 of 3, waarin het cijfer 0 een even aantal keren voorkomt.

Oplossing

Als a_r het gevraagde aantal aanduidt dan is de genererende funktie, die daarbij hoort gelijk aan

$$\begin{aligned} EF(x) &= \left(1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots\right) \left(1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots\right)^3 \\ &= \frac{e^x + e^{-x}}{2} e^{3x} = \frac{e^{4x} + e^{2x}}{2} = \\ &= \sum_{k=0}^{\infty} \frac{4^k + 2^k}{2} \frac{x^k}{k!}. \end{aligned}$$

Dus is $a_r = \frac{4^r + 2^r}{2}$. (Zie ook hoofdstuk 1, voorbeeld 14). $\square$

2.3.6. Andere toepassingen van genererende funkties

Door voor x speciale waarden te substitueren in genererende funkties is het mogelijk allerlei betrekkingen af te leiden tussen de coëfficiënten. Een andere mogelijkheid daartoe is het nemen van speciale konvolutieprodukten in (2.10) en in (2.23).

We geven hiervan een aantal voorbeelden.

Voorbeeld 17

Bewijs voor $n \in \mathbb{N}$, dat

$$\binom{n}{0}^2 + \binom{n}{1}^2 + \dots + \binom{n}{n}^2 = \binom{2n}{n}. \quad (2.25)$$

Oplossing

Neem in (2.10) $a_i = b_i = \binom{n}{i}$. We vinden dan dat $(1+x)^{2n} = (1+x)^n (1+x)^n$ de gewone genererende functie is van de rij

$$\binom{n}{0} \binom{n}{0}, \binom{n}{0} \binom{n}{1} + \binom{n}{1} \binom{n}{0}, \dots$$

Het k^{de} getal in deze rij is

$$\binom{n}{0} \binom{n}{k} + \binom{n}{1} \binom{n}{k-1} + \dots + \binom{n}{k} \binom{n}{0}.$$

Anderzijds is de coëfficiënt van x^k in $(1+x)^{2n}$ gelijk aan $\binom{2n}{k}$. Dus geldt

$$\sum_{i=0}^k \binom{n}{i} \binom{n}{k-i} = \binom{2n}{k}.$$

Hiervan is (2.25) een bijzonder geval. $\square$

Probeer (2.25) ook op kombinatorische wijze af te leiden!

Voorbeeld 18

Bewijs voor $n \in \mathbb{N}$, dat

$$\binom{n}{1} + 2\binom{n}{2} + \dots + n\binom{n}{n} = n2^{n-1}. \quad (2.26)$$

Oplossing

Differentieer de relatie

$$(1+x)^n = \sum_{k=0}^n \binom{n}{k} x^k$$

naar x en substitueer vervolgens $x = 1$. □

Een zeer bijzondere toepassing verkrijgt men door in de konvolutie (2.10) het produkt te nemen van de genererende functie $f(x)$ van de rij $(a_0, a_1, a_2, \dots)$ en de functie $\frac{1}{1-x}$, die de rij $(1, 1, 1, \dots)$ genereert.

De produktfunctie $\frac{f(x)}{1-x}$ genereert dus de rij

$$(a_0, a_0 + a_1, a_0 + a_1 + a_2, \dots),$$

d.w.z. de partiële sommen van de reeks $\sum_{k=0}^{\infty} a_k$. De functie $\frac{1}{1-x}$ doet dus hier dienst als een sommatie-operator.

Voorbeeld 19

Bepaal een uitdrukking voor

$$0^2 + 1^2 + 2^2 + \dots + n^2.$$

Oplossing

Allereerst trachten we de gewone genererende functie te vinden van de rij $(0^2, 1^2, 2^2, \dots)$.

Differentiëren van de reeks $\frac{1}{1-x} = \sum_{k=0}^{\infty} x^k$ levert

$$\frac{1}{(1-x)^2} = \sum_{k=0}^{\infty} kx^{k-1} \quad \text{ofwel}$$

$$\frac{x}{(1-x)^2} = \sum_{k=0}^{\infty} kx^k.$$

Nogmaals differentiëren geeft

$$\frac{1+x}{(1-x)^3} = \sum_{k=0}^{\infty} k^2 x^{k-1}.$$

Hieruit volgt onmiddellijk

$$\frac{x + x^2}{(1-x)^3} = \sum_{k=0}^{\infty} k^2 x^k.$$

Volgens het hierboven besprokene genereert dus $\frac{x + x^2}{(1-x)^4}$ de rij
 $(0^2, 0^2 + 1^2, 0^2 + 1^2 + 2^2, \dots)$.

Ontwikkeling van deze funktie geeft

$$\frac{x + x^2}{(1-x)^4} = (x + x^2) \sum_{r=0}^{\infty} \binom{3+r}{3} x^r,$$

en dus is de koëfficiënt van x^n gelijk aan (neem $r = n-1$ en $r = n-2$)

$$\binom{n+2}{3} + \binom{n+1}{3} = \frac{1}{6} n(n+1)(2n+1),$$

hetgeen gelijk moet zijn aan $0^2 + 1^2 + 2^2 + \dots + n^2$. $\square$

Voorbeeld 20

Bereken $\sum_{k=1}^n k^2 \binom{n}{k}$.

Oplossing

Uit de relatie $\sum_{k=0}^{\infty} \binom{n}{k} x^k = (1+x)^n$ leiden we achtereenvolgens de volgende

formules af:

$$\begin{aligned} \sum_{k=0}^{\infty} k \binom{n}{k} x^{k-1} &= n(1+x)^{n-1}, \\ \sum_{k=0}^{\infty} k \binom{n}{k} x^k &= nx(1+x)^{n-1}, \\ \sum_{k=0}^{\infty} k^2 \binom{n}{k} x^{k-1} &= n(1+nx)(1+x)^{n-2}, \\ \sum_{k=0}^{\infty} k^2 \binom{n}{k} x^k &= nx(1+nx)(1+x)^{n-2}. \end{aligned}$$

Volgens de konvolutiemethode geldt nu dat $\sum_{k=0}^n k^2 \binom{n}{k}$ gelijk is aan de koëfficiënt van x^n in $\frac{nx(1+nx)(1+x)^{n-2}}{1-x} = (nx+n^2x^2) \sum_{k=0}^{\infty} \binom{n-2}{k} x^k \sum_{l=0}^{\infty} x^l$.

Deze koëfficiënt is gelijk aan

$$\begin{aligned} n [\binom{n-2}{0} + \binom{n-2}{1} + \dots + \binom{n-2}{n-2} + 0] &+ n^2 [\binom{n-2}{0} + \binom{n-2}{1} + \dots + \binom{n-2}{n-2}] = \\ n2^{n-2} + n^2 2^{n-2} &= n(n+1)2^{n-2}. \end{aligned}$$

Er is hier echter een veel snellere weg. De funktie $nx(1+nx)(1+x)^{n-2}$ is een polynoom van graad n en dus kunnen we schrijven

$$nx(1+nx)(1+x)^{n-2} = \sum_{k=0}^n a_k x^k.$$

Door te substitueren $x = 1$ krijgen we rechtstreeks $\sum_{k=0}^n a_k = n(1+n)2^{n-2}$. $\square$

3. GENEREREN VAN PERMUTATIES EN KOMBINATIES

3.1. Inleiding

In algoritmen die betrekking hebben op problemen van kombinatorische aard (kombinatorische algoritmen) komt het vaak voor dat er elementen moeten worden gegenereerd van een bepaalde klasse van kombinatorische configuraties. Zo heeft men bijvoorbeeld voor het berekenen van een uitdrukking als

$$\sum_{x_1 \leq x_2 \leq \dots \leq x_k} f(x_1, x_2, \dots, x_k)$$

nodig alle geoorloofde configuraties van de x_i in een of andere systematische volgorde.

In dit hoofdstuk zullen we ons bezighouden met het geval dat die configuraties permutaties of combinaties zijn, hetgeen in de praktijk veel voorkomt. In beide gevallen kan men zich nog twee situaties voorstellen en wel dat men of alle configuraties systematisch wil laten voortbrengen of dat men een random gekozen configuratie nodig heeft. Een manier om zowel systematische als random generatie te verwezenlijken is, om de - zeg N - configuraties expliciet te nummeren met de gehele getallen $0, 1, \dots, N-1$. Systematische voortbrenging houdt dan in, dat men elk nummer konverteert naar de bijbehorende configuratie, terwijl random generatie neerkomt op het kiezen van een random getal tussen 0 en $N-1$ en de konversie daarvan.

Men moet er echter op verdacht zijn, dat het konversieproces bewerkelijk kan zijn, in welk geval men naar efficiëntere algoritmen moet zoeken. Als een algoritme alle elementen (hier configuraties) van een verzameling moet voortbrengen is een belangrijke eigenschap de totale tijd, waarin dit gebeurt. Als deze tijd (of het aantal bewerkingen nodig om de algoritme uit te voeren) een lineaire funktie is van het aantal elementen, spreekt men van een lineaire algoritme. Een dergelijk verband is in het algemeen het beste wat men kan verwachten.

Wat ook belangrijk kan zijn is de mate van verandering tussen twee opeenvolgende configuraties. Het kan soms wenselijk zijn deze verandering zo klein mogelijk te houden. Zo is er in het geval van permutaties een voortbrengende algoritme (zie 3.2.5), waarin twee opeenvolgende permutaties uit elkaar volgen door het verwisselen van slechts twee

(naburige) elementen. Een dergelijke algoritme duiden we kortweg aan met de term algoritme met minimale veranderingen (minimal change algorithm). Voor wat betreft de auteurs van de algoritmen uit hoofdstuk 3 en 4 en de plaatsen waar zij voor het eerst gepubliceerd werden verwijzen we naar het boek van Reingold et al [1] en naar de aldaar genoemde referenties. De meeste algoritmen zijn ook te vinden in Collected Algorithms from A C M.

3.2. Genereren van permutaties

3.2.1. Representatie van getallen door fakulteiten

Gegeven is een natuurlijk getal n . Beschouw de rij delingen

$$\begin{aligned} n &= 1q_0 + d_0, & d_0 &= 0, \\ q_0 &= 2q_1 + d_1, & 0 \leq d_1 &< 2, \\ q_1 &= 3q_2 + d_2, & 0 \leq d_2 &< 3, \\ q_2 &= 4q_3 + d_3, & 0 \leq d_3 &< 4, \\ &\vdots \\ &\vdots \\ &\vdots \end{aligned}$$

waarbij alle $q_i \geq 0$ zijn.

De rij $q_0, q_1, q_2, q_3, \dots$ is monotoon strikt afnemend en eindigt dus voor zekere $q_j = 0$. Door substitutie verkrijgt men de relatie

$$n = d_0 0! + d_1 1! + d_2 2! + d_3 3! + \dots + d_j j!, \quad (3.1)$$

$$0 \leq d_i < i + 1. \quad (3.2)$$

Let erop dat d_0 altijd gelijk is aan 0.

Stelling (3.1)

De koëfficiënten d_i uit (3.1) en (3.2) behoren op één-éénduidige wijze bij een natuurlijk getal n .

Bewijs

Evident is dat bij $d_0, d_1, \dots, d_j$ op éénduidige wijze een getal n hoort.

Stel het getal n heeft naast (3.1) nog de representatie

$$n = d_0' 0! + d_1' 1! + d_2' 2! + \dots.$$

Dan geldt

$$(d_1 - d_1') 1! + (d_2 - d_2') 2! + \dots = 0.$$

Stel m is de grootste index waarvoor geldt $d_m - d_m' \neq 0$. Hieruit volgt

$$m! = \frac{1}{d_m' - d_m} \{ (d_1 - d_1')1! + (d_2 - d_2')2! + \dots \\ + (d_{m-1} - d_{m-1}') (m-1)! \} ,$$

$$\text{en } \left| \frac{d_i - d_i'}{d_m - d_m'} \right| \leq |d_i - d_i'| \leq i.$$

Dus is

$$m! \leq 1.1! + 2.2! + \dots + (m-1).(m-1)! .$$

Dit is echter in tegenspraak met de algemeen geldende en met volledige inductie eenvoudig te bewijzen formule (zie vraagstuk 27d van hoofdstuk 1).

$$m! - 1 = 1.1! + 2.2! + \dots + (m-1).(m-1)! \quad \square$$

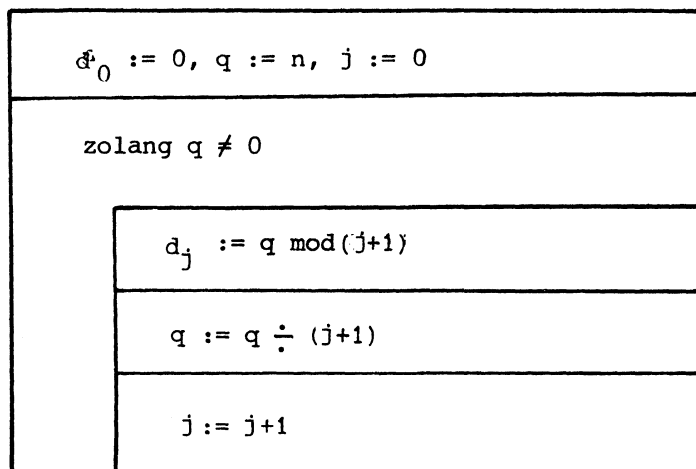
Voor een ander bewijs van stelling (3.1) wordt men verwezen naar vraagstuk 1 van hoofdstuk 3.

Uit de konstruktie die tot (3.1) leidde en uit bovenstaande stelling volgt nu dat we een natuurlijk getal n op één-éénduidige wijze kunnen representeren door de rij

$$(d_0, d_1, d_2, \dots, d_j). \quad (3.3)$$

Deze representatie met fakulteiten staat wel bekend als het factorial number system (fakulteitsstelsel).

Ga zelf na, dat de genoemde konstruktie door de volgende algoritme wordt weergegeven.



Algoritme 3.1

Opmerking

De in deze en in volgende algoritmen gebruikte konventies zijn vastgelegd in [10].

In de toekenningsopdracht $d_j := q \bmod (j+1)$ krijgt de variabele d_j de waarde van de rest van de deling $q \div (j+1)$.

3.2.2. Lexicografische ordening

Beschouw de $(n-)$ permutaties van de n verschillende elementen van de verzameling $\{1, 2, \dots, n\}$.

Definitie

De permutatie $\sigma := \sigma_1 \sigma_2 \dots \sigma_n$ gaat lexicografisch vooraf aan de permutatie $\tau := \tau_1 \tau_2 \dots \tau_n$ dan en slechts dan als voor zekere $k \geq 1$ geldt $\sigma_j = \tau_j$ als $j < k$ en $\sigma_k < \tau_k$.

Als voorbeeld nemen we de permutaties van drie elementen. In lexicografische volgorde zijn dat 123, 132, 213, 231, 312, 321.

Deze ordening is dus equivalent met de natuurlijke ordening die we kunnen aanbrengen als we de permutaties opvatten als getallen in het tientallig stelsel of algemener in het $(n+1)$ -tallig stelsel.

Wij stellen nu het rangnummer van een permutatie $\pi := \pi_1 \pi_2 \dots \pi_n$ voor door $\text{ind}_n(\pi)$ en spreken tevens af dat de permutatie $1\ 2 \dots n$ het rangnummer 0 krijgt. Er geldt dan de volgende stelling.

Stelling (3.2)

Het lexicografische rangnummer van de permutatie π wordt bepaald door de rekursieve betrekking

$$\text{ind}_n(\pi) = (\pi_1 - 1)(n-1)! + \text{ind}_{n-1}(\pi'), \quad (3.4)$$

$$\text{ind}_1(1) = 0, \quad (3.5)$$

waarin $\pi' := \pi'_1 \pi'_2 \dots \pi'_{n-1}$ de permutatie is van $1, 2, \dots, n-1$, die uit π ontstaat door π_1 weg te laten en alle elementen groter dan π_1 met 1 te verminderen.

Bewijs

We bewijzen (3.4) met volledige inductie. Neem een vaste $n \in \mathbb{N}$ en veronderstel, dat de formule juist is voor alle permutaties van de getallen $1, 2, \dots, n-1$. ($n \geq 1$)

Aan de beschouwde permutatie π gaan vooraf alle permutaties van de getallen $1, 2, \dots, n$, waarvan het eerste getal kleiner is dan π_1 . In totaal zijn dat $(\pi_1 - 1)(n-1)!$ permutaties.

Bovendien gaan aan π nog een aantal permutaties vooraf, waarvan het eerste getal gelijk is aan π_1 . Hun aantal kan men als volgt tellen. Men vat $\pi_2, \pi_3, \dots, \pi_n$ op als een permutatie van de getallen van $\{1, 2, \dots, n\} \setminus \{\pi_1\}$ en bepaalt hoeveel van dit soort permutaties in lexicografische zin voorafgaan aan $\pi_2 \pi_3 \dots \pi_n$. Omdat π_1 toch niet meedoet kan men evengoed beschouwen de permutatie, die men uit $\pi_2 \pi_3 \dots \pi_n$ verkrijgt door alle getallen groter dan π_1 met 1 te verminderen. Men krijgt dan de permutatie van $1, 2, \dots, n-1$, die we π' genoemd hebben.

Het gezochte aantal is gelijk aan het rangnummer van π' in lexicografische zin. Volgens de inductie-aanname is dit $\text{ind}_{n-1}(\pi')$.

Dus geldt nu

$$\text{ind}_n(\pi) = (\pi_1 - 1)(n-1)! + \text{ind}_{n-1}(\pi').$$

Voor $n = 1$ geldt per definitie $\text{ind}_1(1) = 0 = 0 \cdot 0! + 0$.

De formule (3.4) is dus juist voor alle $n \geq 1$. $\square$

Het resultaat van stelling (3.2) stelt ons in staat, om het rangnummer van een permutatie in de lexicografische ordening te berekenen.

Voorbeeld 1

Bepaal de index van de 9-permutatie

4 3 5 2 7 1 9 8 6

in de lexicografische ordening.

Oplossing

$$\text{ind}_9(4, 3, 5, 2, 7, 1, 9, 8, 6) =$$

$$3 \cdot 8! + \text{ind}_8(3, 4, 2, 6, 1, 8, 7, 5) =$$

$$3 \cdot 8! + 2 \cdot 7! + \text{ind}_7(3, 2, 5, 1, 7, 6, 4) =$$

.
.
.
.
.

$$3 \cdot 8! + 2 \cdot 7! + 2 \cdot 6! + 1 \cdot 5! + 2 \cdot 4! + 0 \cdot 3! + 2 \cdot 2! + \text{ind}_2(2, 1) =$$

$$\begin{aligned}
 & 3.8! + 2.7! + 2.6! + 1.5! + 2.4! + 0.3! + 2.2! + 1.1! + \text{ind}_1(1) = \\
 & 3.8! + 2.7! + 2.6! + 1.5! + 2.4! + 0.3! + 2.2! + 1.1! + 0.0! = \\
 & 132653. \qquad \qquad \qquad \square
 \end{aligned}$$

Past men (3.4) iteratief toe, dan krijgt men (zoals ook in voorbeeld 1 te zien is):

$$\text{ind}_n(\pi) = (\pi_1 - 1)(n-1)! + (\pi_1' - 1)(n-2)! + \dots,$$

met $\pi_1 - 1 < n$, $\pi_1' - 1 < n-1, \dots$. Met behulp van (3.1) en (3.2) zien we, dat de getallen $\pi_1 - 1, \pi_1' - 1, \dots$ (in omgekeerde volgorde) de representatie vormen van het getal $\text{ind}_n(\pi)$ in het fakulteitsstelsel. Deze opmerking maakt het mogelijk, om bij een gegeven rangnummer i de bijbehorende permutatie op betrekkelijk eenvoudige wijze te konstrueren. Ook hiervan geven we een voorbeeld.

Voorbeeld 2

Bepaal de miljoenste 12-permutatie in de lexicografische ordening of iets preciezer gezegd: bepaal π met $\text{ind}_{12}(\pi) = 10^6$.

Oplossing

Allereerst bepalen we de representatie van 10^6 in het fakulteitsstelsel:

$$\begin{aligned}
 10^6 &= 0.0! + 0.1! + 2.2! + 2.3! + 1.4! + 5.5! + 2.6! + 6.7! + 6.8! \\
 &\quad + 2.9! + 0.10! + 0.11!
 \end{aligned}$$

en we schrijven

$$(d_0, d_1, d_2, \dots, d_{11}) = (0, 0, 2, 2, 1, 5, 2, 6, 6, 2, 0, 0).$$

We beginnen met de 1-permutatie 1:

$$\text{ind}_1(1) = d_0 \cdot 0! = 0.0!$$

Aan deze permutatie voegen we aan de voorkant toe het getal $d_1 + 1 = 0 + 1 = 1$ en vermeerderen alle elementen $> d_1$ met 1:

$$\text{ind}_2(1, 2) = d_1 \cdot 1! + d_0 \cdot 0!$$

Vervolgens voegen we aan de voorkant toe het getal $d_2 + 1 = 2 + 1 = 3$ en vermeerderen alle elementen $> d_2$ met 1:

$$\text{ind}_3(3, 1, 2) = d_2 \cdot 2! + d_1 \cdot 1! + d_0 \cdot 0!$$

De volgende permutatie, die we zo krijgen is 3 4 1 2, waarvoor geldt

$$\text{ind}_4(3,4,1,2) = d_3 \cdot 3! + d_2 \cdot 2! + d_1 \cdot 1! + d_0 \cdot 0!$$

Zo voortgaande vinden we achtereenvolgens de permutaties

2 4 5 1 3
6 2 4 5 1 3
3 7 2 5 6 1 4
7 3 8 2 5 6 1 4
7 8 3 9 2 5 6 1 4
3 8 9 4 10 2 6 7 1 5
1 4 9 10 5 11 3 7 8 2 6
1 2 5 10 11 6 12 4 8 9 3 7 □

Op geheel analoge wijze kunnen k -permutaties uit n , met $k < n$, lexicografisch geordend worden. We geven hier geen algemene stelling omtrent de rangnummers van dergelijke permutaties, maar verwijzen naar de opmerking nã voorbeeld 10.

3.2.3. Systematisch genereren van permutaties

Zij gegeven een permutatie $\pi := \pi_1 \pi_2 \dots \pi_n$.

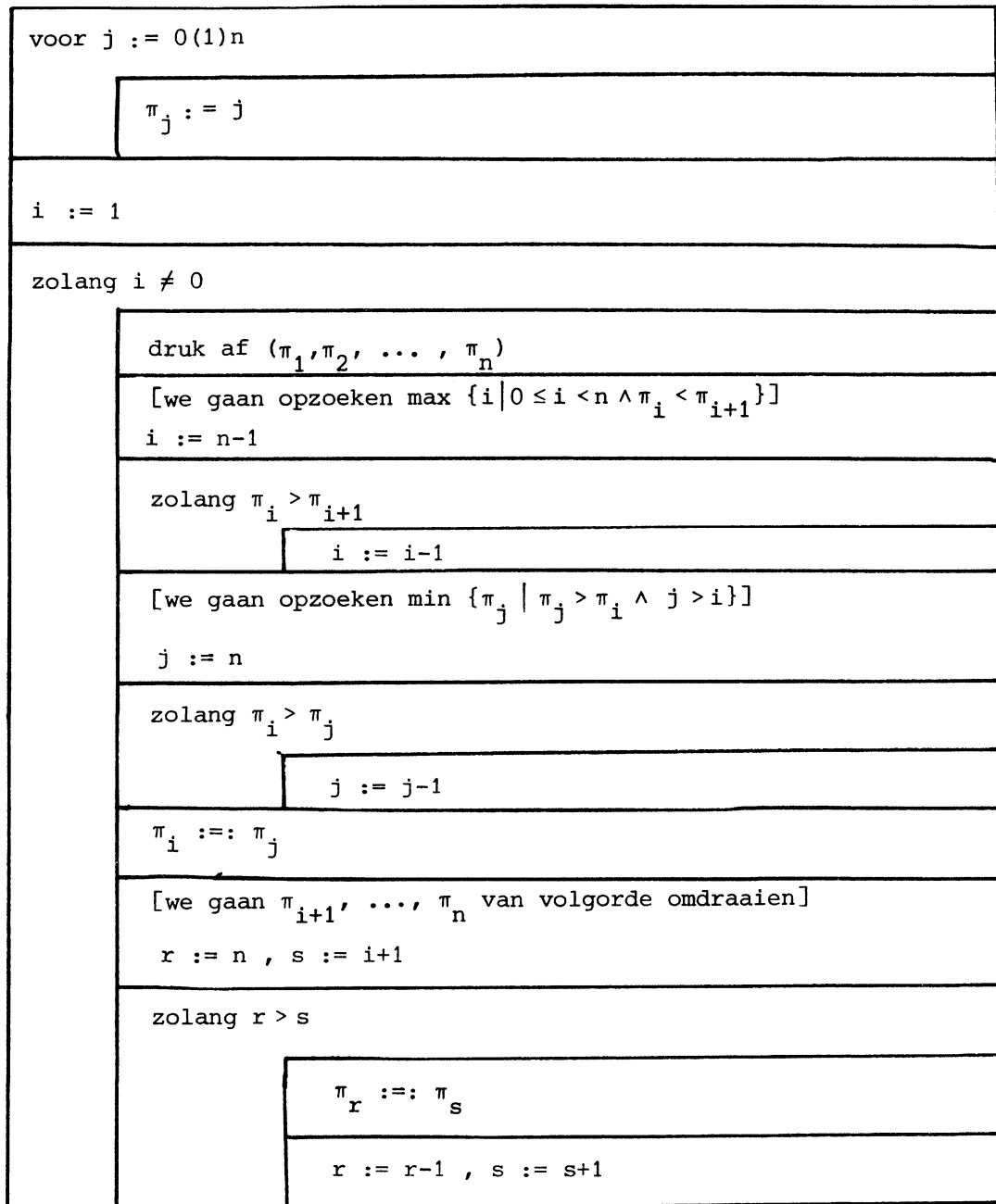
We zullen een manier aangeven, om de opvolger van π in de lexicografische ordening te bepalen. Bedenk hierbij steeds, dat deze ordening equivalent is met de natuurlijke ordening als we de permutaties opvatten als getallen in een $(n+1)$ -tallig stelsel.

Spoor eerst het element π_i op, dat zo ver mogelijk naar rechts staat en dat de eigenschap heeft $\pi_i < \pi_{i+1}$ ($0 \leq i < n$). Onderlinge permutatie van de elementen rechts van π_i levert slechts getallen op, die kleiner dan π zijn. Door π_i te verwisselen met een element rechts van π_i , dat groter is dan π_i , krijgt men een permutatie groter dan π . Hetzelfde gebeurt als we π_i verwisselen met een element links van π_i , dat kleiner is dan π_i . Echter zijn de permutaties, die dan ontstaan groter dan de eerder genoemde permutaties. De kleinste permutatie groter dan π (de opvolger van π) verkrijgt men door π_i te verwisselen met het kleinste element rechts van π_i , dat groter is dan π_i en door vervolgens de elementen $\pi_{i+1}, \dots, \pi_n$ (die in grootte afnemen) in omgekeerde volgorde te plaatsen.

Als voorbeeld beschouwen we het geval $n = 7$ en de permutatie

3 6 4 7 5 2 1. Via de tussenstap 3 6 5 7 4 2 1 vinden we de opvolger
3 6 5 1 2 4 7.

In algoritme 3.2 is deze redenering verwerkt om alle permutaties van de verzameling $\{1, 2, \dots, n\}$ te produceren in lexicografische volgorde.



Algoritme 3.2

Opmerkingen

1. De opdracht $a :=: b$ betekent a en b wisselen van waarde en is een compacte schrijfwijze voor een drietal gewone toekenningsoptdrachten. Voorts is commentaar steeds tussen vierkante haakjes geplaatst.

2. De variabele π_0 heeft slechts een formele betekenis in de algoritme en heeft steeds de waarde 0. Als tengevolge van de opdracht $i := i-1$ de waarde van i gelijk aan 0 is geworden, geldt zeker $\pi_0 = 0 < \pi_1$ en wordt de zolangopdracht niet meer uitgevoerd.
3. Zodra er in de opdracht zolang $\pi_i > \pi_j \dots$ geldt $\pi_i < \pi_j$ weten we, dat dit tevens de kleinste π_j is, rechts van π_i , met deze eigenschap. Was er namelijk een $\pi_{j'}$, met $\pi_i < \pi_{j'} < \pi_j$ en $i < j' < j$ dan was π_i niet het meest rechtse element geweest met $\pi_i < \pi_{i+1}$.

Een korrektheidsbewijs van algoritme 3.2 verloopt als volgt.

Eerst toont men aan dat de uitvoer begint met de permutatie $1 \ 2 \ \dots \ n$.

Vervolgens toont men aan dat uitgaande van een permutatie $\pi \neq n \ \dots \ 2 \ 1$

de opvolger van π in lexicografische zin wordt geproduceerd en dat de algoritme stopt als $\pi = n \ \dots \ 2 \ 1$. Op de details gaan we hier niet nader in.

3.2.4. Komplexiteit van algoritme 3.2

Onder de komplexiteit van een algoritme zullen wij verstaan het aantal bewerkingen, waarin de algoritme wordt uitgevoerd. Deze definitie is in hoge mate vaag. Immers wat moet men onder een bewerking verstaan?

Dit kan afhangen van de programmeertaal en de programmeerstijl, maar ook van allerlei implementatiedetails, zoals het ophogen van indices in een herhalingsopdracht, het berekenen van array-indices en van pointers in sommige datastructuren, etc.. In het uiterste geval zou men op machine-niveau het totale aantal instructies moeten tellen. Zo ver hoeft men echter niet te gaan als men alleen maar een schatting wil hebben van het aantal bewerkingen of het asymptotisch gedrag ervan wil weten of als men een aantal algoritmen met elkaar wil vergelijken. Het blijkt dan vaak voldoende te zijn, om een aantal basisbewerkingen te tellen, die karakteristiek zijn voor het type algoritme of voor het soort problemen, waar de algoritme betrekking op heeft. Zo zal bij het sorteren van getallen uit een lijst het vergelijken van twee lijstelementen een basisbewerking zijn. Bij matrixvermenigvuldiging zullen het vermenigvuldigen en het optellen van twee reële getallen als basisbewerkingen moeten worden beschouwd. In gedetailleerdere beschouwingen kan men dan weer onderscheid maken tussen deze beide bewerkingen, omdat vermenigvuldigen in het algemeen meer tijd kost dan optellen.

Wij zullen in dit diktaat aannemen - en in een enkel geval bewijzen - dat het totale aantal bewerkingen ruwweg evenredig is met het aantal basisbewerkingen. Vooreen goed gekozen stel basisbewerkingen zal dit, zeker voor grote waarden van de parameters van de algoritme, in het algemeen juist zijn. Voor een verdere discussie van deze problematiek raadplege men vooral hoofdstuk 1 van [7].

In het geval van algoritme 3.2 blijken de basisbewerkingen te zijn het verwisselen en het vergelijken van de getallen π_i . We merken verder op, dat in de lexicografisch geordende rij permutaties er een volledige deelrij is van de $k!$ permutaties van de k meest rechtse elementen, ($1 \leq k < n-1$) met de linker $n-k$ elementen vast. Dit maakt het mogelijk om recurrente betrekkingen af te leiden voor de getallen I_k en C_k die respectievelijk aanduiden het aantal verwisselingen $\pi_i := \pi_j$ en $\pi_r := \pi_s$ en het aantal "vergelijkingen" $\pi_i > \pi_{i+1}$ en $\pi_i > \pi_j$, gedaan om de eerste $k!$ permutaties te genereren. In het bijzonder geeft I_{n-1} het aantal verwisselingen aan, nodig om de $(n-1)!$ permutaties te genereren bij een vaste waarde van π_1 . Omdat π_1 de waarden 1 t/m n kan hebben geeft dit dus een bijdrage $n I_{n-1}$ tot het aantal verwisselingen. Bovendien moeten we nog de verwisselingen tellen, die worden uitgevoerd bij de overgang van de l^{de} deelrij van $(n-1)!$ permutaties naar de $(l+1)^{\text{ste}}$ deelrij ($1 \leq l \leq n-1$). Dit aantal bedraagt $1 + \lfloor \frac{n-1}{2} \rfloor = \lfloor \frac{n+1}{2} \rfloor$. (Eén keer voor $\pi_1 := \pi_j$ en $\lfloor \frac{n-1}{2} \rfloor$ keer voor paren elementen van de rij $\pi_2, \pi_3, \dots, \pi_n$.)

De recurrente betrekking wordt dus

$$I_n = n I_{n-1} + (n-1) \lfloor \frac{n+1}{2} \rfloor, \quad (3.6)$$

$$I_1 = 0. \quad (3.7)$$

Buiten beschouwing is hierbij gebleven het aantal verwisselingen dat gedaan wordt nadat de laatste permutatie is afgedrukt, dus in het $i = 0$ geval. Dit aantal is gelijk aan $1 + \lfloor \frac{n}{2} \rfloor = \lfloor \frac{n+2}{2} \rfloor$.

Analoog leiden we een betrekking af voor de getallen C_k . Voor elk van de n waarden van π_1 zijn C_{n-1} vergelijkingen nodig om de deelrijen met $(n-1)!$ permutaties te genereren. De overgang van de l^{de} deelrij ($\pi_1=1$) naar de $(l+1)^{\text{ste}}$ ($\pi_1=l+1$) deelrij kost $n-1$ vergelijkingen van het type $\pi_i > \pi_{i+1}$ en l vergelijkingen van het type $\pi_i > \pi_j$. Samen dus

$\sum_{l=1}^{n-1} (n-1+l) = \frac{1}{2} (n-1)(3n-2)$ vergelijkingen. (Bedenk dat bij overgang naar een volgende deelrij de elementen achter π_1 in afnemende grootte staan!)

De recurrente betrekking wordt:

$$C_n = n C_{n-1} + \frac{1}{2}(n-1)(3n-2), \quad (3.8)$$

$$C_1 = 0 \quad (3.9)$$

Voor het geval $i = 0$ worden er nog eens $n+1$ vergelijkingen gedaan, en wel

n keer $\pi_i > \pi_{i+1}$ en 1 keer $\pi_0 > \pi_j$.

Om (3.6) op te lossen voeren we de variabele $S_n = I_n + \lfloor \frac{n+1}{2} \rfloor$ in.

De betrekkingen (3.6) en (3.7) gaan dan over in

$$S_n = n S_{n-1} + n \epsilon_{n-1}, \quad \epsilon_n = \begin{cases} 0, & n \text{ is oneven} \\ 1, & n \text{ is even} \end{cases} \quad (3.10)$$

$$S_1 = 1. \quad (3.11)$$

Vervolgens definiëren we de exponentiële genererende functie

$$S(x) = \sum_{n=1}^{\infty} \frac{S_n}{n!} x^n.$$

Met (3.10) en (3.11) leiden we af dat

$$(1-x)S(x) = x + x \left(\frac{x^2}{2!} + \frac{x^4}{4!} + \dots \right) = x \cosh x$$

ofwel

$$S(x) = \frac{x \cosh x}{1-x}.$$

Wanneer we bedenken dat $\frac{1}{1-x}$ een "sommatie operator" is (zie 2.3.6) dan valt eenvoudig in te zien dat

$$S_n = n! \left(1 + \frac{1}{2!} + \frac{1}{4!} + \dots + \frac{1}{(2 \lfloor (n-1)/2 \rfloor)!} \right) \quad (3.12)$$

en dus

$$I_n = n! \sum_{j=0}^{\lfloor (n-1)/2 \rfloor} \frac{1}{(2j)!} - \lfloor \frac{n+1}{2} \rfloor. \quad (3.13)$$

We merken nog op dat (3.10) ook met gewone genererende functies is op te lossen. Men ga dat zelf na.

Samen met de $\lfloor n+2/2 \rfloor$ verwisselingen voor $i=0$ is dus het totale aantal verwisselingen in algoritme 3.2 gelijk aan

$$n! \sum_{j=0}^{\lfloor (n-1)/2 \rfloor} \frac{1}{(2j)!} + \epsilon_n.$$

Omdat

$$\sum_{j=0}^{\lfloor (n-1)/2 \rfloor} \frac{1}{(2j)!} \sim \cosh 1 \approx 1.54308$$

vinden we, dat de algoritme ongeveer $1.54308 n!$ verwisselingen doet, om de $n!$ permutaties te genereren, voor "grote" waarden van n .

Op analoge wijze kan men (3.8) oplossen (doe dit!).

Het resultaat luidt

$$C_n = n! \left(\frac{3}{2} \sum_{j=0}^n \frac{1}{j!} - 1 \right) - \frac{1}{2} (3n+1) . \quad (3.14)$$

Samen met de $n+1$ vergelijkingen voor $i = 0$ is dus het totale aantal vergelijkingen in algoritme 3.2 gelijk aan

$$n! \left(\frac{3}{2} \sum_{j=0}^n \frac{1}{j!} - 1 \right) - \frac{1}{2} (n-1) .$$

Daar

$$\sum_{j=0}^n \frac{1}{j!} \sim e \approx 2.71828$$

is het totale aantal vergelijkingen in de algoritme ongeveer gelijk aan $3.07742 n!$, voor "grote" waarden van n .

Om nu tot een totaal aantal basisbewerkingen te komen moeten we de twee hier ingevoerde soorten met elkaar vergelijken. Het lijkt redelijk om de verwisselingsbewerking een gewichtsfactor 3 te geven, omdat deze 3 toekenningen bevat. Verder geven we de vergelijkingsbewerking een gewichtsfactor 2, waarbij we er gemakshalve van uitgaan, dat het testen van $a > b$ wordt uitgevoerd via $a-b > 0$, zodat we met 1 aritmetische operatie en 1 testinstructie te maken hebben. We voeren hier dus als het ware 3 soorten elementaire bewerkingen in : toekenningen, aritmetische operaties en testinstructies. Wanneer we deze elementaire bewerkingen onderling als even zwaar beschouwen, hebben we een soort basiseenheid, waarin we de basisbewerkingen kunnen uitdrukken. Een verwisseling telt dan voor 3 basiseenheden en een vergelijking voor 2.

Het hoeft nauwelijks betoog, dat hier een element van willekeur is geïntroduceerd. Men moet daar echter niet te zwaar aan tillen. De bedoeling is slechts om een indruk te geven van de manier, waarop kompleksiteitsbeschouwingen kunnen verlopen.

De totale kompleksiteit t.g.v. verwisselen en vergelijken, uitgedrukt in basiseenheden is dus gelijk aan

$$3 \times 1.54308n! + 2 \times 3.07742! = 10.78408n! \quad (3.15)$$

Tenslotte tonen we nog aan, dat de kompleksiteit t.g.v. de basisbewerkingen bepalend is voor het totale aantal bewerkingen, dat door de algoritme wordt uitgevoerd. Daartoe is het voldoende om te laten zien, dat dit totale aantal bewerkingen evenredig is met (3.15), ofwel dat de niet-beschouwde bewerkingen ~~de~~ basisbewerkingen in grootte-orde niet overvleugelen. Is dat één keer aangetoond, dan hebben we dus achteraf een bewijs dat onze basisbewerkingen ook echte "basisbewerkingen" zijn.

Dergelijke bewijzen worden niet vaak gegeven en in dit diktaat zullen de bewijzen van stellingen (3.3) en (3.5) ook de enige in hun soort zijn. In de praktijk beperkt men zich er meestal toe bepaalde bewerkingen tot basisbewerkingen te verklaren, op grond van evidentie of ervaring.

Stelling (3.3)

Het totale aantal bewerkingen, uitgevoerd door algoritme 3.2 bij het genereren van alle n -permutaties in lexicografische volgorde is asymptotisch evenredig met $n!$ en dus is de algoritme lineair.

(*) Bewijs

(i) De voor-opdracht kan explicieter geschreven worden als

$j := 0 ;$
zolang $j \leq n$ doe $\begin{cases} \pi_j := j ; \\ j := j+1 \end{cases}$

Hieruit blijkt dat er $n+2$ vergelijkingen worden gedaan, $2n+3$ toekenningen en $n+1$ aritmetische operaties.

(ii) De vergelijking $i \neq 0$ wordt $(n! + 1)$ keer gedaan. De toekenningsopdrachten $i := n-1$, $j := n$, $r := n$ en $s := i+1$ worden elke keer uitgevoerd als de zolang-lus wordt doorlopen. Dit houdt dus in $4n!$ toekenningen en $2n!$ aritmetische operaties..

(iii) Het totale aantal toekenningsopdrachten $i := i-1$ en $j := j-1$ is gelijk aan het totale aantal vergelijkingen $C_n + n + 1$ minus $2n!$. Deze opdrachten houden dus in $(C_n + n + 1) - 2n!$ toekenningen en evenveel aritmetische operaties. ($2n!$ is het aantal keer dat de opdracht niet wordt uitgevoerd, omdat de voorwaarde niet meer geldt.)

(iv) De vergelijking $r > s$ in de laatste zolang-opdracht wordt

$((I_n + \lfloor \frac{n+2}{2} \rfloor) - n!)$ keer uitgevoerd. De twee opdrachten $r := r-1$ en $s := s+1$ samen betekenen $2((I_n + \lfloor \frac{n+2}{2} \rfloor) - n!)$ toekenningen en evenzoveel aritmetische operaties.

(v) We zagen reeds, dat de opdrachten $\pi_i := \pi_j$ en $\pi_r := \pi_s$ samen voor $3(I_n + \lfloor \frac{n+2}{2} \rfloor)$ toekenningen verantwoordelijk zijn.

(vi) Tenslotte gaven $\pi_i > \pi_{i+1}$ en $\pi_i > \pi_j$ aanleiding tot $C_n + n + 1$ vergelijkingen.

Tellen we dit allemaal op, dan vinden we de volgende aantallen.

toekenningen:

$$2n+3 + 4n! + (C_n + n + 1) - 2n! + 5(I_n + \lfloor \frac{n+2}{2} \rfloor) - 2n! =$$

$$C_n + 5I_n + 3n + 5\lfloor \frac{n+2}{2} \rfloor + 4 \sim 3.07742 n! + 5 \times 1.54308n! =$$

$$10.79282 n! ;$$

aritmetische operaties:

$$n+1 + 2n! + (C_n + n + 1) - 2n! + 2(I_n + \lfloor \frac{n+2}{2} \rfloor) - 2n! =$$

$$C_n + 2I_n - 2n! + 2n + 2 \lfloor \frac{n+2}{2} \rfloor + 2 \sim 3.07742 n! + 3.08616 n! - 2n! =$$

$$4.16358n!$$

vergelijkingen:

$$n + 2 + C_n + n + 1 + n! + 1 + I_n + \lfloor \frac{n+2}{2} \rfloor - n! \sim$$

$$3.07742n! + 1.54308n! = 4.62050n!$$

Het totale aantal bewerkingen, uitgedrukt in basiseenheden is dus inderdaad evenredig met het getal in (3.15) en daarom ook met $n!$

Daar het totale aantal af te drukken permutaties gelijk is aan $n!$ is de algoritme dus lineair. $\square$

Ondanks deze lineariteit is algoritme 3.2 toch niet de meeste efficiënte algoritme om alle permutaties van n getallen te genereren. Het berekenen van de lexicografische opvolger van een permutatie is nu eenmaal een tamelijk bewerkelijk proces. Kiest men in plaats van de lexicografische ordening een andere ordening, dan is het mogelijk een snellere algoritme te konstrueren. Dit is het onderwerp van de volgende paragraaf.

3.2.5. Genereren van permutaties door verwisseling nabuurelementen

Zoals reeds werd opgemerkt zal het vinden van efficiëntere algoritmen er vooral van afhangen of men er in slaagt een volgende permutatie snel uit zijn voorganger te berekenen. Het ligt voor de hand, om aan te nemen dat de hoeveelheid werk die hiermee is gemoeid minimaal is, wanneer de ene permutatie uit de andere volgt door slechts twee elementen van plaats te verwisselen. De volgende (rekursieve) methode, waarin slechts nabuurwisselingen plaatsvinden, dient zich dan aan.

Stel we hebben een rij van alle $(n-1)$ -permutaties, die uit elkaar ontstaan door nabuurtwisselingen. We nemen de eerste permutatie uit deze rij en voegen daaraan het element n toe, door dit achteraan te zetten. Vervolgens zetten we n tussen het laatste en het één na laatste element etc., totdat n vooraan is gekomen. Op deze wijze beweegt n dus van rechts naar links door de $(n-1)$ -permutatie. Daarna nemen we de volgende $(n-1)$ -permutatie (die volgens de inductie-aanname uit zijn voorganger ontstaat door slechts één nabuurtwisseling en handhaven n voorop. Dan laten we n van links naar rechts bewegen etc. Omdat we voor $n = 2$ alle permutaties kunnen ordenen met nabuurtwisselingen hebben we nu voor alle $n \geq 2$ een voorschrift om alle n -permutaties door nabuurtwisselingen te verkrijgen. Een formeel bewijs, dat we zo inderdaad alle n -permutaties krijgen laten we achterwege.

Voorbeeld 3

Als illustratie geven we de rij van 4-permutaties, die ontstaat door bovenstaand voorschrift achtereenvolgens toe te passen voor $n = 2$, $n = 3$, en $n = 4$.

$$n = 2 \quad 1 \ 2, \ 2 \ 1 \quad ;$$
$$n = 3 \quad 1\ 2\ 3, 1\ 3\ 2, 3\ 1\ 2, \\ 3\ 2\ 1, 2\ 3\ 1, 2\ 1\ 3 \quad ;$$

n = 4 1 2 3 4 , 1 2 4 3 , 1 4 2 3 , 4 1 2 3 ,
 4 1 3 2 , 1 4 3 2 , 1 3 4 2 , 1 3 2 4 ,
 3 1 2 4 , 3 1 4 2 , 3 4 1 2 , 4 3 1 2 ,
 4 3 2 1 , 3 4 2 1 , 3 2 4 1 , 3 2 1 4 ,
 2 3 1 4 , 2 3 4 1 , 2 4 3 1 , 4 2 3 1 ,
 4 2 1 3 , 2 4 1 3 , 2 1 4 3 , 2 1 3 4 □

Algoritme 3.3 is gebaseerd op deze beschouwing. In de algoritme wordt gebruik gemaakt van een drietal rijen variabelen.

In de eerste plaats gebruikt men een rij $\pi_1, \pi_2, \dots, \pi_n$ voor de "lopende" permutatie. In de tweede plaats is er een rij $p_1, p_2, \dots, p_n$, die de inverse permutatie voorstelt van $\pi := \pi_1 \pi_2 \dots \pi_n$ en waarin p_i de plaats van i aangeeft. In de derde plaats wordt er een rij $d_1, d_2, \dots, d_n$ gebruikt, om de "bewegingstoestand" van de elementen bij te houden en wel is $d_i = 1$ als i bezig is naar rechts te bewegen, $d_i = -1$ als i bezig is naar links te bewegen terwijl $d_i = 0$ als i geen mogelijkheid heeft om te bewegen. In feite is dit laatste

slechts het geval voor $i = 1$, zoals direkt zal blijken. Een element blijft bewegen totdat het een groter element ontmoet. Dan stopt het element, zijn bewegingsrichting wordt omgedraaid (d.w.z. d_i krijgt ander teken) en er wordt verder gegaan met het element dat één kleiner is dan het voorgaande. Zodra dit element niet verder in zijn bewegingsrichting blijkt te kunnen bewegen (tegen een groter element "opbotst") wordt ook zijn richting omgedraaid en wordt het volgende element genomen, etc. De hulpvariabelen π_0 en π_{n+1} krijgen de waarde $n+1$, om ervoor te zorgen, dat elementen niet links van de 1^{ste} plaats en niet rechts van de n^{de} plaats terecht komen.

voor $i := 1(1)n$	
	$\pi_i := p_i := i, d_i := -1$
$d_1 := 0$	
$\pi_0 := \pi_{n+1} := m := n+1$	
zolang $m \neq 1$	
	druk af $(\pi_1, \pi_2, \dots, \pi_n)$
	$m := n$
	zolang $\pi_{p_m + d_m} > m$ [als dit waar is staat links of rechts van m een groter element.]
	$d_m := -d_m$
	$m := m-1$
	[verwisseling van naburen]
	$\pi_{p_m} :=: \pi_{p_m + d_m}$
	[verwisseling van bijbehorende plaatsnummers]
	$p_{\pi_{p_m}} :=: p_m$

Algoritme 3.3 (Trotter, Johnson)

Ook nu geven we geen formeel korrektheidsbewijs van de algoritme maar schetsen slechts hoe een dergelijk bewijs zou verlopen. Allereerst bewijst men, dat als een permutatie $\pi := \pi_1 \pi_2 \dots \pi_n$ is afgedrukt de waarden van d_i , $1 \leq i < n$ precies dezelfde zijn als de korresponderende waarden, die ze zouden hebben als de permutatie $\pi-n$ (dit is de permutatie π waaruit n is weggelaten) zou worden afgedrukt. Deze eigenschap kan bewezen worden met inductie naar het aantal permutaties dat is gegenereerd. Vervolgens bewijst men de korrektheid van de algoritme met behulp van inductie naar n .

Algoritme 3.3 is een voorbeeld van een minimal change algorithm. De volgorde, waarin de permutaties worden gegenereerd is een andere dan de lexicografische volgorde uit 3.2.2. Duiden we het rangnummer van een n -permutatie π weer aan met $\text{ind}_n(\pi)$ en definiëren we $\text{ind}_n(1\ 2 \dots n) = 0$ dan kunnen we de volgende stelling afleiden.

Stelling (3.4)

Het rangnummer van de permutatie π , waarmee deze wordt gegenereerd door algoritme 3.3, wordt bepaald door de rekursieve betrekking

$$\text{ind}_n(\pi) = n \cdot \text{ind}_{n-1}(\pi-n) + \begin{cases} n-k & \text{als } \text{ind}_{n-1}(\pi-n) \text{ is even} \\ k-1 & \text{als } \text{ind}_{n-1}(\pi-n) \text{ is oneven,} \end{cases} \quad (3.16)$$

$$\text{ind}_1(1) = 0, \quad (3.17)$$

waarin k gedefinieerd wordt door $\pi_k = n$.

Bewijs

Beschouw een willekeurige n -permutatie π . Deze is ontstaan door in de $(n-1)$ -permutatie $\pi-n$ het element n op de k^{de} plaats van links te zetten. Eerder is het element n al in alle voorafgaande $(n-1)$ -permutaties gezet en wel in elke permutatie op n verschillende manieren. Er zijn $\text{ind}_{n-1}(\pi-n)$ aan $\pi-n$ voorafgaande $(n-1)$ -permutaties. Deze zorgen dus voor $n \cdot \text{ind}_{n-1}(\pi-n)$ aan π voorafgaande n -permutaties. Als $\text{ind}_{n-1}(\pi-n)$ even is, houdt dit in, dat n de permutatie $\pi-n$ van rechts naar links doorloopt en als $\text{ind}_{n-1}(\pi-n)$ oneven is van links naar rechts. (N.B. de permutatie $1\ 2 \dots n-1$ is de 0^{de} $(n-1)$ -permutatie).

In het even-geval gaan er dus nogeens $n-k$ permutaties vooraf aan π en in het oneven geval $k-1$ permutaties. $\square$

Ook voor deze nummering geven we een tweetal voorbeelden, die laten zien hoe men het rangnummer van een gegeven permutatie kan bepalen respectievelijk welke permutatie een gegeven rangnummer heeft.

Voorbeeld 4

Bepaal de index van de 9-permutatie

4 3 5 2 7 1 9 8 6

in de minimal change ordening.

Oplossing

Herhaaldelijk toepassen van (3.16) geeft achtereenvolgens:

$$\begin{aligned}
 \text{ind}_1(1) &= 0 & , \\
 \text{ind}_2(2,1) &= 2 \cdot 0 + 2 - 1 = 1 & , \\
 \text{ind}_3(3,2,1) &= 3 \cdot 1 + 1 - 1 = 3 & , \\
 \text{ind}_4(4,3,2,1) &= 4 \cdot 3 + 1 - 1 = 12 & , \\
 \text{ind}_5(4,3,5,2,1) &= 5 \cdot 12 + 5 - 3 = 62 & , \\
 \text{ind}_6(4,3,5,2,1,6) &= 6 \cdot 62 + 6 - 6 = 372 & , \\
 \text{ind}_7(4,3,5,2,7,1,6) &= 7 \cdot 372 + 7 - 5 = 2606 & , \\
 \text{ind}_8(4,3,5,2,7,1,8,6) &= 8 \cdot 2606 + 8 - 7 = 20849 & , \\
 \text{ind}_9(4,3,5,2,7,1,9,8,6) &= 9 \cdot 20849 + 7 - 1 = 187647 & \quad \square
 \end{aligned}$$

Voorbeeld 5

Bepaal de miljoenste 12-permutatie in de minimal change ordening.

(preciezer gezegd : bepaal π met $\text{ind}_{12}(\pi) = 10^6$)

Oplossing

Stel π is de gevraagde permutatie. Er geldt dan volgens (3.16)

$$\text{ind}_{12}(\pi) = 10^6 = 12 \cdot 83333 + 4,$$

met $k-1 = 4$, omdat 83333 oneven is.

Hieruit volgt $k = 5$ en dus $\pi_5 = 12$.

Analoog vinden we voor $\pi^{(1)} = \pi - 12$

$$\text{ind}_{11}(\pi^{(1)}) = 83333 = 11 \cdot 7575 + 8$$

en dus $k-1 = 8$, $\pi_9^{(1)} = 11$. Omdat $9 \geq 5$ is, staat in π het getal 11 rechts van 12, zodat $\pi_{10} = 11$.

Achtereenvolgens berekenen we nu

$$\pi^{(2)} = \pi^{(1)} - 11,$$

$$\text{ind}_{10}(\pi^{(2)}) = 7575 = 10 \cdot 757 + 5,$$

$$k-1 = 5, \pi_6^{(2)} = \pi_6^{(1)} = \pi_7 = 10;$$

$$\pi^{(3)} = \pi^{(2)} - 10,$$

$$\text{ind}_9(\pi^{(3)}) = 757 = 9 \cdot 84 + 1,$$

$$9-k = 1, \pi_8^{(3)} = \pi_9^{(2)} = \pi_{10}^{(1)} = \pi_{11} = 9;$$

$$\pi^{(4)} = \pi^{(3)} - 9,$$

$$\text{ind}_8(\pi^{(4)}) = 84 = 8 \cdot 10 + 4,$$

$$8-k = 4, \pi_4^{(4)} = \pi_4^{(3)} = \pi_4^{(2)} = \pi_4^{(1)} = \pi_4 = 8;$$

$$\pi^{(5)} = \pi^{(4)} - 8,$$

$$\text{ind}_7(\pi^{(5)}) = 10 = 7 \cdot 1 + 3,$$

$$k-1 = 3, \pi_4^{(5)} = \pi_5^{(4)} = \pi_5^{(3)} = \pi_5^{(2)} = \pi_5^{(1)} = \pi_6 = 7;$$

$$\pi^{(6)} = \pi^{(5)} - 7,$$

$$\text{ind}_6(\pi^{(6)}) = 1 = 6 \cdot 0 + 1,$$

$$6-k = 1, \pi_5^{(6)} = \pi_6^{(5)} = \pi_7^{(4)} = \pi_7^{(3)} = \pi_8^{(2)} = \pi_8^{(1)} = \pi_9 = 6;$$

$$\pi^{(7)} = \pi^{(6)} - 6,$$

$$\text{ind}_5(\pi^{(7)}) = 0 = 5 \cdot 0 + 0,$$

$$5-k = 0, \pi_5^{(7)} = \pi_6^{(6)} = \pi_7^{(5)} = \pi_8^{(4)} = \pi_9^{(3)} = \pi_{10}^{(2)} = \pi_{11}^{(1)} = \pi_{12} = 5,$$

en evenzo

$$\pi_4^{(7)} = 4, \pi_3^{(7)} = 3, \pi_2^{(7)} = 2, \pi_1^{(7)} = 1,$$

waaruit volgt

$$\pi_8 = 4, \pi_3 = 3, \pi_2 = 2, \pi_1 = 1.$$

De gevraagde permutatie is dus

$$\pi = 1 \quad 2 \quad 3 \quad 8 \quad 12 \quad 7 \quad 10 \quad 4 \quad 6 \quad 11 \quad 9 \quad 5 \quad \square$$

Opmerking

In plaats van direkt de plaats te bepalen van een getal in de oorspronkelijke permutatie π , kan men dit ook achteraf doen. Zodra men een volledige permutatie $\pi^{(i)}$ in handen heeft, kan men hieruit achtereenvolgens konstrueren $\pi^{(i-1)}, \pi^{(i-2)}, \dots, \pi$.

In voorbeeld 5 zou dit dus als volgt gaan:

$$\pi^{(7)} = 1 \quad 2 \quad 3 \quad 4 \quad 5$$

$$\pi^{(6)} = 1 \quad 2 \quad 3 \quad 4 \quad 6 \quad 5$$

$$\pi^{(5)} = 1 \quad 2 \quad 3 \quad 7 \quad 4 \quad 6 \quad 5$$

etc.

Er bestaat een snelle methode om direkt (zonder berekening van $\text{ind}_n(\pi)$) de opvolger van een gegeven permutatie π te bepalen. Zie hiervoor vraagstuk 9b van hoofdstuk 3.

Tenslotte geven we nog wat informatie betreffende de kompleksiteit van de algoritme in de vorm van een stelling.

Stelling (3.5)

Algoritme 3.3 is asymptotisch lineair.

(*) Bewijs

Als meest elementaire bewerkingen nemen we, evenals bij algoritme 3.2 de vergelijkingen, de toekenningen en de aritmetische operaties.

- (i) Allereerst tellen we het totale aantal vergelijkingen. De vergelijking van m met 1 in de buitenste zolang-opdracht gaat vooraf aan het afdrucken van een permutatie en wordt dus in totaal $n!+1$ keer uitgevoerd.

Beschouw nu de vergelijking $\pi_{p_m + d_m} > m$.

Stel C_n is het aantal keren dat deze wordt uitgevoerd. De vergelijking wordt minstens één keer uitgevoerd voor elke keer dat de buitenste lus wordt doorlopen. Hij wordt vaker dan één keer uitgevoerd dan en slechts dan als $\pi_{p_n + d_n} > n$. Dit laatste is alleen het geval als of

$p_n = n \wedge d_n = 1$ of als $p_n = 1 \wedge d_n = -1$. Eén van deze beide gebeurtenissen vindt plaats als π één der permutaties $\pi^{(1)}, \pi^{(2)}, \pi^{(3)} \dots$ is, waarbij $\pi^{(1)}, \pi^{(2)}, \dots, \pi^{((n-1)!)}$ een volledige rij $(n-1)$ -permutaties is. Het totale aantal extra keren dat de vergelijking $\pi_{p_m + d_m} > m$ wordt uitgevoerd voor deze $(n-1)!$ permutaties is gelijk aan C_{n-1} . Dus geldt $C_n = n! + C_{n-1}$, waaruit met $C_1 = 1$ volgt

$$C_n = \sum_{i=1}^n i! = n! + o(n!).$$

Het totaal aantal keren, dat er een vergelijking wordt uitgevoerd is dus $2n! + o(n!)$.

- (ii) Analoog als in het bewijs van stelling 3.3 blijkt dat het aantal toekenningen gelijk is aan

$$2C_n + n! + 3n! + 3n! = 9n! + o(n!).$$

(iii) Het aantal aritmetische operaties is gelijk aan

$$3n! + o(n!).$$

(Als operaties hebben we hier de bewerking $m-1$ en twee keer $p_m + d_m$ geteld).

In totaal vinden we dus dat er $13n! + o(n!)$ elementaire bewerkingen worden uitgevoerd. $\square$

Vergelijken we de aantallen uit bovenstaand bewijs met die in het bewijs van stelling (3.3), dan zien we, dat hoewel beide algoritmen asymptotisch lineair zijn, algoritme 3.3 toch sneller is dan algoritme 3.2.

Opmerking

Zowel in het bewijs van stelling (3.3) als van stelling (3.5) hebben we de af-drukopdrachten niet in de telling meegenomen. In beide algoritmen wordt er natuurlijk $n!$ keer een afdrukopdracht uitgevoerd.

3.3 Genereren van combinaties

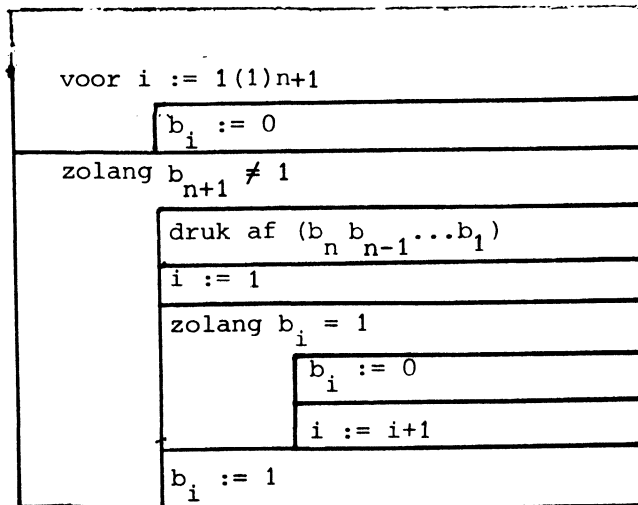
3.3.1 Genereren van alle combinaties uit n objecten

Zoals in 1.3.1 werd opgemerkt is een r -kombinatie uit n op te vatten als een deelverzameling ter grootte r van een verzameling ter grootte n . Het genereren van alle mogelijke combinaties, zonder te letten op de grootte, betekent dan het voortbrengen van alle deelverzamelingen van een verzameling met n objecten. Dit kan men doen door alle bitrijen $b_1 b_2 \dots b_n$ ter lengte n te genereren, ofwel alle n -bits woorden.

De n -bits woorden corresponderen één-éénduidig met de deelverzamelingen van een verzameling $V := \{a_1, a_2, \dots, a_n\}$. Als $b_i = 1$ betekent dit, dat a_i in de betreffende deelverzameling zit, terwijl dit niet het geval is als $b_i = 0$. Men noemt $(b_1, b_2, \dots, b_n)$ wel de karakteristieke vektor van de deelverzameling.

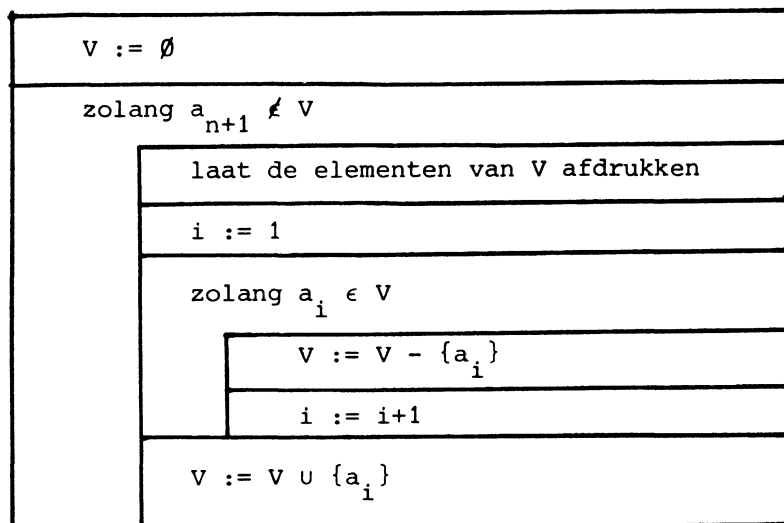
De meest voor de hand liggende manier om alle n -bits woorden voort te brengen is, om ze in het tweetallig stelsel te tellen van 0 tot en met $2^n - 1$.

Ga zelf na, dat dit in de volgende algoritme gebeurt.



Algoritme 3.4

De vertaling van deze algoritme in termen van verzamelingen is eenvoudig en luidt als volgt.



Algoritme 3.5

De wijze, waarop de verzameling V in deze algoritme gerepresenteerd moet worden laten we hier onbesproken.

De basisbewerking in bovenstaande algoritmen is de vergelijking $b_i = 1$ (resp. $a_i \in V$). Deze vergelijking wordt getest elke keer nadat er een bitrij $b_n b_{n-1} \dots b_1$ is afgedrukt, dus 2^n keer. Hij wordt nog eens getest voor elke keer dat $b_1 = 1$, dus 2^{n-1} keer, etc. Het totale aantal keren dat de vergelijking wordt getest is dus $2^n + 2^{n-1} + \dots + 1 = 2^{n+1} - 1 \sim 2 \cdot 2^n$.

De algoritmen zijn dus lineair.

Algoritme 3.4 drukt de getallen $0, 1, 2, \dots, 2^n - 1$ in binaire vorm af in hun natuurlijke volgorde.

Voorbeeld 6

In het geval $n = 3$ worden respectievelijk de volgende bitrijen afgedrukt:

000, 001, 010, 011, 100, 101, 110, 111 □

Hieraan is te zien, dat de algoritmen niet tot de categorie der algoritmen met minimale verandering behoren.

Algoritmen voor het afdrukken van alle n -bits woorden (ofwel voor alle combinaties uit n objecten), die wèl tot deze categorie behoren, zijn het onderwerp van de volgende paragrafen.

3.3.2 Gray codes

Als hulpmiddel gebruiken we het begrip Gray kode.

Definitie

Een n -bits Gray kode is een rij van alle n -bits woorden, zodanig geordend, dat elke twee opeenvolgende woorden in slechts één (bit)positie verschillen.

De 2^n woorden van een Gray kode worden in dit verband codewoorden genoemd. Als bovendien het eerste codewoord met het laatste codewoord ook in slechts één positie verschilt spreken we van een cyklische Gray kode.

Voorbeeld 7

Voor $n = 3$ bestaat er de volgende (cyklische) Gray kode:

000, 001, 011, 010, 110, 111, 101, 100 □

Vergelijk bovenstaande volgorde met die uit voorbeeld 6.

Er zijn, voor vaste n , verschillende Gray codes. Wij zullen ons hier tot een speciaal type beperken, namelijk de binair-gespiegelde Gray kode (binary-reflected Gray-code). We definiëren deze kode rekursief.

Stel dat we een n -bits Gray kode voorstellen door een $(2^n \times n)$ -matrix van nullen en enen, zodanig dat de i^{de} rij het i^{de} codewoord voorstelt ($0 \leq i \leq 2^n - 1$):

$$G(n) = \begin{pmatrix} G_0 \\ G_1 \\ \cdot \\ \cdot \\ G_{2^n-1} \end{pmatrix} \quad (3.18)$$

De matrix

$$G(n+1) = \begin{pmatrix} 0 & G_0 \\ 0 & G_1 \\ \cdot & \\ \cdot & \\ \cdot & \\ 0 & G_{2^n-1} \\ 1 & G_{2^n-1} \\ 1 & G_{2^n-2} \\ \cdot & \\ \cdot & \\ \cdot & \\ 1 & G_0 \end{pmatrix} \quad (3.19)$$

stelt dan een (n+1)-bits Gray kode voor, zoals men direkt kan nagaan. Door uit te gaan van de triviale 1-bits kode

$$G(1) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (3.20)$$

hebben we daarmee voor alle $n \geq 1$ een Gray kode vastgelegd.

We kunnen de hierboven ingevoerde kodes ook op een andere wijze vastleggen.

Daartoe voeren we het begrip overgangsrij in.

Definitie

Nummert men de kolommen van een Gray kode van rechts naar links met de getallen 1 tot en met n en geeft $t_i, 0 \leq i \leq 2^n-2$, de positie aan in het i^{de} woord, waar verandering optreedt bij overgang naar het volgende woord, dan heet de rij

$$T_n = t_0 t_1 \dots t_{2^n-2}$$

de overgangsrij van de kode.

Voorbeeld 8

De overgangsrij van de kode uit voorbeeld 7 luidt

$$T_3 = 1 \ 2 \ 1 \ 3 \ 1 \ 2 \ 1 \quad \square$$

Stelling (3.6)

De overgangsrij van de binair-gespiegelde Gray kode wordt gegeven door

$$\begin{aligned} T_1 &= 1 \\ T_{n+1} &= T_n(n+1)T_n \end{aligned} \quad .$$

Bewijs

Voor $n = 1$ is de bewering juist.

Als de bewering geldt voor zekere $n \geq 1$, dan geldt hij ook voor $n+1$ zoals men gemakkelijk vaststelt met behulp van de definitie van de binair-gespiegelde Gray kode. Met inductie volgt nu de juistheid van de stelling. $\square$

Opmerkingen

1. De overgangsrij van de binair-gespiegelde Gray kode is altijd symmetrisch onder spiegeling van links naar rechts.
2. In de vraagstukken 11 en 12 wordt een methode afgeleid, om direkt (dus op niet-rekursieve wijze) voor een gegeven waarde van i het i^{de} kodewoord van $G(n)$ te berekenen. Het resultaat geven we hier in de vorm van een stelling.

Stelling (3.7)

Zij $G_i = g_n g_{n-1} \dots g_1$ het i^{de} kodewoord van $G(n)$, $0 \leq i \leq 2^n-1$, en zij $i = b_{n+1} b_n \dots b_1$ de binaire representatie van het getal i ($b_{n+1}=0$). Er geldt dan dat

$$g_j = b_{j+1} + b_j \pmod{2}, \quad 1 \leq j \leq n.$$

De oplossing van het inverse probleem wordt gegeven door de volgende stelling (zie vraagstuk 18):

Stelling (3.8)

Zij i en G_i zoals gedefinieerd in de vorige stelling, dan geldt:

$$b_j = g_n + g_{n-1} + \dots + g_j \pmod{2}, \quad 1 \leq j \leq n.$$

3.3.3 Genereren van de binair-gespiegelde Gray kode

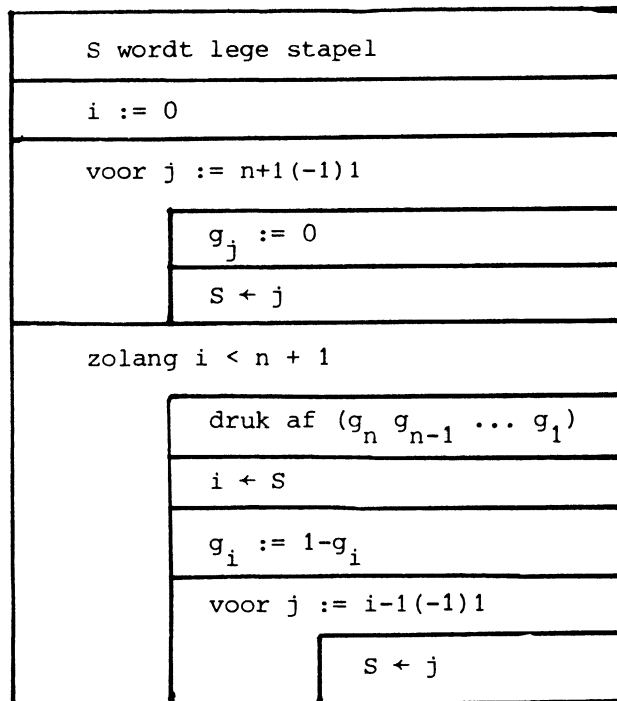
Uit de vorige paragraaf zal duidelijk zijn geworden, dat als men de beschikking heeft over een algoritme, die de woorden van $G(n)$ genereert, deze ook opgevat kan worden als een minimal change generator voor alle combinaties uit n objecten. Hieronder presenteren we een dergelijke algoritme. Er wordt daarin gebruik gemaakt van een stapel (zie [6]). Stelt S een stapel voor dan zullen we met de opdracht

$S \leftarrow x$

bedoelen: zet de waarde van x op de stapel (de push-operatie) en met

$x \leftarrow S$

bedoelen we: geef x de waarde van de top van de stapel en verwijder die top van S (de pop-operatie).



Algoritme 3.6

Men kan als volgt inzien dat algoritme 3.6 de kode $G(n)$ genereert.

De stapel bevat aanvankelijk van boven naar beneden de getallen $1, 2, \dots, n+1$. Bij elke doorloop van de zolang-opdracht wordt de top i van de stapel verwijderd en wordt het i^{de} bit van rechts veranderd. Vervolgens worden de getallen $i-1, i-2, \dots, 1$ (in deze volgorde) op de stapel gezet als $i > 1$ is, terwijl er niets op de stapel wordt gezet als $i = 1$.

De korrekte werking van de algoritme steunt nu op de eigenschap, dat de top van de stapel achtereenvolgens alle waarden aanneemt van de elementen van de overgangsrij T_n . Men bewijst dit met inductie. Voor $n = 1$ gaat men, door de algoritme uit te voeren, eenvoudig na dat deze bewering waar is.

Stel de bewering is waar voor $n-1$ ($n \geq 1$). Begin met de situatie dat de stapel $1, 2, \dots, n, n+1$ bevat. De top van de stapel krijgt dan achtereenvolgens de waarden van de elementen van T_{n-1} , (terwijl $g_n = 0$ blijft), volgens onze inductieveronderstelling.

Als het laatste woord $0 G_{2^{n-1}}$ is afgedrukt bestaat de stapel S uit de getallen $n, n+1$. Dan krijgt i de waarde n , het n^{de} bit wordt veranderd en S krijgt de gedaante $1, 2, \dots, n-1, n+1$. Omdat de eerste $n-1$ elementen van de stapel nu evenals in het begin de getallen $1, 2, \dots, n-1$ bevatten, doorloopt de top van de stapel opnieuw de rij T_{n-1} .

Tenslotte wordt de top en dus ook i gelijk aan $n+1$ en stopt de algoritme.

We geven nog een tweede versie van algoritme 3.6 waarbij we de stapel S implementeren met behulp van een array $(\tau_n, \tau_{n-1}, \dots, \tau_0)$. Het element τ_0 is steeds gelijk aan de top van S en voor $j \geq 1$ is τ_j gelijk aan het getal onder j op de stapel, als j op de stapel zit terwijl $\tau_j = j+1$ als j niet op de stapel zit.

Stel $\tau_0 = i$, dan is τ_i gelijk aan het getal onder i .

Als we i verwijderen voegen we $i-1, i-2, \dots, 1$ aan de stapel toe. Alle waarden van $\tau_{i-2}, \tau_{i-3}, \dots, \tau_1$ zijn dan al korrekt. Immers $i-2, i-3, \dots, 1$ kwamen niet op de stapel voor en dus gold $\tau_{i-2} = i-1, \tau_{i-3} = i-2, \dots, \tau_1 = 2$. De enige aanpassingen die we moeten maken zijn:

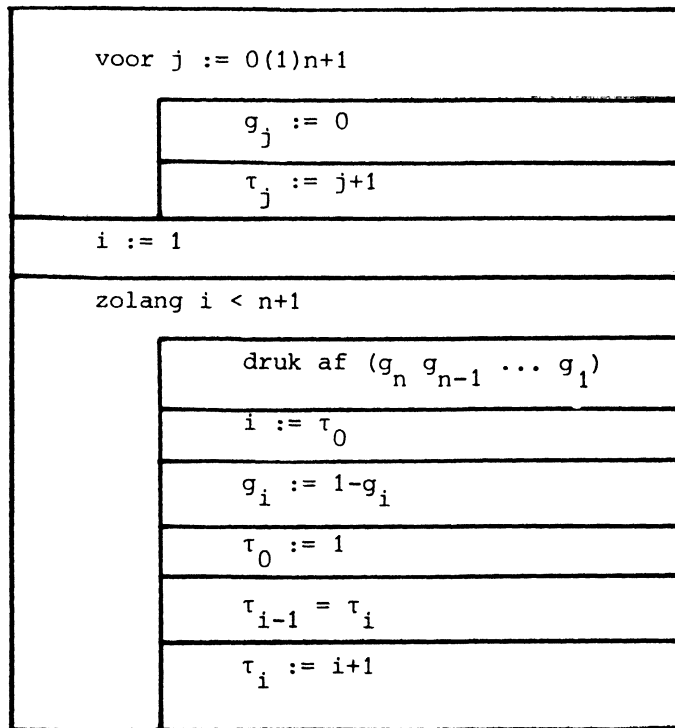
τ_{i-1} moet de waarde krijgen van τ_i ;

τ_i zelf moet gelijk gesteld worden aan $i+1$, want i komt niet meer voor;

als $i \neq 1$ is moet τ_0 gelijk aan 1 worden;

als $i = 1$ moet τ_0 de waarde τ_1 krijgen (op de vorige bladzijde was afgesproken, dat er in dit geval niets op de stapel gezet zou worden).

We komen zo tot de volgende algoritme (Bitner, Ehrlich, Reingold).



Algoritme 3.7

Opmerkingen

1. De inkorrekte waarde, die τ_0 krijgt als $i = 1$ wordt onmiddellijk gecorrigeerd door de opdracht $\tau_{i-1} := \tau_i$.
2. De implementatie van de stapel S , zoals die hier boven is gebeurd, zorgt ervoor dat de binnenste herhalingsopdracht uit algoritme 3.6 verdwenen is. Elke overgang naar een volgend woord kost nu dezelfde hoeveelheid tijd.

3.3.4 Genereren van k-kombinaties in lexicografische volgorde

De $\binom{n}{k}$ k-kombinaties van de getallen $\{1, 2, \dots, n\}$ kunnen evenals de in 3.2.2 beschouwde permutaties lexicografisch geordend worden. De definitie van deze ordening is geheel analoog aan de in 3.2.2 gegeven definitie.

Voorbeeld 9

De $\binom{5}{3}$ 3-kombinaties van de getallen $\{1, 2, 3, 4, 5\}$ zijn in lexicografische volgorde

123, 124, 125, 134, 135,

145, 234, 235, 245, 345 $\square$

Evenals we dat in 3.2.2 deden kunnen we de combinaties nummeren. Beschouw de combinatie $a := a_1 a_2 \dots a_k$, met $1 \leq a_1 < a_2 < \dots < a_k \leq n$ (zie 1.3). Het rangnummer van a in de lexicografisch geordende lijst van k-kombinaties uit $\{1, 2, \dots, n\}$ duiden we aan met $\text{ind}_{n,k}(a)$. Voor $a = 1 \ 2 \ \dots \ k$ definiëren we $\text{ind}_{n,k}(a) = 1$ (let op het verschil met het rangnummer van de eenheidspermutatie dat zowel in de lexicografische ordening als in de minimal change ordening gelijk aan 0 was gekozen).

Stelling(3.9)

Het lexicografische rangnummer van de combinatie a wordt gegeven door

$$\text{ind}_{n,k}(a) = \binom{n}{k} - \sum_{j=1}^k \binom{n-a_j}{k-j+1}.$$

Bewijs

We bewijzen bovenstaande uitdrukking door het aantal combinaties te tellen, die na de combinatie a komen in de lexicografische ordening.

Allereerst zijn dat die combinaties die beginnen met a_1+1 of met a_1+2 , etc. Hiervan zijn er $\binom{n-a_1}{k}$, omdat $n-a_1$ het totale aantal getallen groter dan a_1 is. Verder volgen op a , die combinaties die beginnen met a_1 en waarvan het tweede getal groter dan a_2 is. Hiervan zijn er $\binom{n-a_2}{k-1}$. Eveneens volgen op a die kombintaties die beginnen met $a_1 a_2$ en waarvan het derde getal groter is dan a_3 . Hiervan zijn er $\binom{n-a_3}{k-2}$. Enzovoort.

Omdat het totale aantal combinaties gelijk is aan $\binom{n}{k}$ is hiermee de formule bewezen. $\square$

Opmerking

Men kan ook een uitdrukking voor $\text{ind}_{n,k}(a)$ afleiden door het aantal combinaties voorafgaande aan a te tellen. Ga na dat deze uitdrukking minder eenvoudig is dan die gegeven in stelling (3.9).

Voorbeeld 10

Beschouw alle strikt monotone woorden van lengte 4 over het gewone latijnse alfabet. Wat is het rangnummer van het woord *klop* in lexicografische ordening?

Oplossing

Het antwoord volgt direkt door stelling (3.9) toe te passen met $n = 26$ en $k = 4$ en luidt :

$$\binom{26}{4} - \binom{15}{4} - \binom{14}{3} - \binom{11}{2} - \binom{10}{1} = 13156. \quad \square$$

Opmerking

Men zou in plaats van alle k -kombinaties uit n ook alle k -permutaties uit n lexicografisch kunnen ordenen. Voor het bijzondere geval $k = n$ deden we dat reeds in 3.2.2.

Bijvoorbeeld kunnen we alle woorden van lengte 4 beschouwen, die elke letter uit het latijnse alfabet hoogstens één keer bevatten. Het woord *klop* (zie voorbeeld 10) wordt dan voorafgegaan door $10 \cdot P(25,3)$ woorden die beginnen met één van de letters $a, b, \dots, j$, alsmede door $10 \cdot P(24,2)$ woorden die beginnen met $ka, kb, \dots, kj$, en door $12 \cdot P(23,1)$ woorden die beginnen met $kla, klb, \dots, kln$. Tenslotte gaan aan *klop* nog vooraf de 12 woorden $kloa, klob, \dots, klon$. Het rangnummer is dus

$$10 \cdot P(25,3) + 10 \cdot P(24,2) + 12 \cdot P(23,1) + 12 + 1 = 1132 \cdot 23! + 13.$$

Een algemene formule voor dit rangnummer geven we niet.

In toepassingen van de uitdrukking uit stelling (3.9) kan men soms met voordeel gebruik maken van de volgende identiteit voor binomiaalkoefficienten

$$\sum_{j=1}^k \binom{p-j}{q} + \binom{p-k}{q+1} = \binom{p}{q+1}. \quad (3.21)$$

Deze betrekking geldt voor alle $p \geq 1$, $q \geq 0$ en $1 \leq k \leq p$. Een bewijs is eenvoudig te geven met volledige inductie naar k . Ga dit na.

Ondermeer zal (3.21) gebruikt worden bij de kompleksiteitsanalyse van algoritme 3.8.

Algoritme 3.8 genereert alle k -kombinaties uit $\{1, 2, \dots, n\}$. Uitgaande van de "lopende" combinatie wordt het meest **rechtse** element opgezocht, dat nog niet zijn maximum bereikt heeft. Bedenk dat de maximale waarde van g_j in de k -combinatie $g_1 g_2 \dots g_k$ gelijk is aan $n-k+j$. Dit element wordt dan met 1 opgehoogd en de elementen rechts ervan krijgen hun kleinst mogelijke waarden, waarbij er van links naar rechts wordt gewerkt. Door te beginnen met de combinatie $1\ 2 \dots k$ krijgt men zo alle k -kombinaties.

$g_0 := -1$
voor $i := 1(1)k$
$g_i := i$
$j := 1$
zolang $j \neq 0$
druk af $(g_1, g_2, \dots, g_k)$
[we gaan het meest rechtse element zoeken dat nog opgehoogd kan worden]
$j := k$
zolang $g_j = n-k+j$
$j := j-1$
$g_j := g_j + 1$
[de elementen rechts van g_j krijgen hun minimale waarde]
voor $i := j+1(1)k$
$g_i := g_{i-1} + 1$

Algoritme 3.8

Opmerking

De opdracht $g_0 := -1$ zorgt ervoor, dat de binnenste zolang-opdracht stopt als $j = 0$ is geworden.

Voor een kompleksiteitsbeschouwing nemen we aan, dat de basisbewerkingen zijn de vergelijkingen $j \neq 0, g_j = n-k+j$ en de toekenningsoopdracht $g_i := g_{i-1} + 1$. Het aantal opdrachten in de zolang-opdrachten en het aantal handelingen, dat gepaard gaat met de herhaling $i := j+1(1)k$ zijn beide evenredig met de genoemde basisbewerkingen.

Stelling (3.10)

In het geval dat $k = o(n)$ is algoritme 3.8 asymptotisch lineair in het aantal combinaties, dat gegenereerd wordt.

Bewijs

- (i) Het aantal keren, dat de vergelijking $j \neq 0$ wordt getest is gelijk aan $\binom{n}{k} + 1$;
- (ii) De vergelijking $g_j = n - k + j$ wordt getest voor elke k -combinatie. Hij wordt nog eens getest voor elke k -combinatie met $g_k = n$, nog eens voor elke k -combinatie met $g_k = n$ en $g_{k-1} = n-1$ etc. Het totaal aantal keren dat deze vergelijking wordt getest is dus gelijk aan
- $$\sum_{j=0}^k \binom{n-j}{k-j} = \sum_{j=0}^k \binom{n-j}{n-k} = \binom{n+1}{k},$$
- waarbij (3.21) is gebruikt.
- (iii) Uit (i) en (ii) volgt nu eenvoudig het aantal keren, dat de opdracht $g_i := g_{i-1} + 1$ wordt uitgevoerd. Elk element g_j waarvoor gold $g_j = n - k + j$ krijgt namelijk in de voor-opdracht zijn minimale waarde. Het totale aantal keren dat deze gelijkheid waar blijkt te zijn is het totale aantal keren dat de gelijkheid getest wordt minus het aantal keren, dat hij niet waar blijkt te zijn. Dit laatste aantal is gelijk aan het aantal afgedrukte combinaties, dus aan $\binom{n}{k}$. Het gezochte aantal is daarom
- $$\binom{n+1}{k} - \binom{n}{k} = \binom{n}{k-1}.$$

Er geldt

$$\binom{n+1}{k} = \binom{n}{k} \frac{n+1}{n-k+1} = \binom{n}{k} \left(1 + \frac{k}{n-k+1}\right) = \binom{n}{k} \left[1 + \frac{k}{n} \left(1 + o\left(\frac{k}{n}\right)\right)\right]$$

$$\text{en } \binom{n}{k-1} = \binom{n}{k} \frac{k}{n-k+1} = \binom{n}{k} \frac{k}{n} \left(1 + o\left(\frac{k}{n}\right)\right).$$

Uit een en ander volgt, dat als $k = o(n)$ het aantal bewerkingen evenredig is met $\binom{n}{k}$ voor grote waarden van n . $\square$

Opmerking

Als k niet klein is ten opzichte van n is algoritme 3.8 niet meer lineair. In dat geval kan men de algoritme efficiënter benutten door er $(n-k)$ -combinaties mee te genereren.

3.2.5 Genereren van k-kombinaties door middel van minimale veranderingen

Evenals voor het genereren van alle combinaties uit n objecten, zullen we nu een algoritme met minimale veranderingen afleiden voor het genereren van alle k -kombinaties uit n . In termen van bitrijen ofwel van binaire woorden betekent dit dat we alle n -bits woorden met precies k enen willen laten afdrukken.

Zoals een eenvoudige beschouwing leert, zal het aantal te veranderen bits bij de overgang naar een volgend woord 2 of meer bedragen. Het beste wat men mag verwachten is dus een rij n -bits woorden, waarbij elk woord uit zijn voorganger ontstaat door 2 bits te wijzigen. Een dergelijke rij is inderdaad te konstrueren en wel met behulp van de in 3.3.2 besproken binair-gespiegelde Gray kode $G(n)$.

Definitie

We definiëren als $G(n,k)$ de deelrij van kodewoorden van $G(n)$, die precies k enen bevatten. Op grond van (3.19) kunnen we ook nog de volgende formele definitie geven:

$$G(n,k) = \begin{pmatrix} 0 & G(n-1, k) \\ 1 & G(n-1, k-1)^R \end{pmatrix}, \quad n > k > 0, \quad (3.22)$$

$$G(n,0) = 0^n, \quad (3.23)$$

$$G(n,n) = 1^n. \quad (3.24)$$

Met de notatie $G(n,k)^R$ bedoelen we hier de kodewoorden van $G(n,k)$, waarbij de volgorde der kodewoorden is omgedraaid ten opzichte van die van $G(n,k)$.

De schrijfwijze $0^1 1^m$ betekent in dit verband de konkateneratie van 1 nullen en m enen.

Stelling (3.11)

Opeenvolgende kodewoorden in $G(n,k)$ verschillen in precies 2 bits, voor $n \geq 2$.

Bewijs

Zoals eenvoudig valt af te leiden uit de rekursieve definitie (3.19) is het eerste kodewoord in $G(n)$ met precies k enen het woord $0^{n-k} 1^k$. Het laatste kodewoord met k enen is $1 0^{n-k} 1^{k-1}$ in het geval $k \geq 1$ en 0^n in het geval $k = 0$. De stelling volgt nu eenvoudig met inductie naar n . Voor $n = 2$ ($k = 0, 1$ of 2) is de stelling triviaal. Stel vervolgens, dat de stelling juist is voor zekere $n \geq 2$ en voor alle k met $0 \leq k \leq n$. We moeten dan aantonen, dat de stelling ook juist is voor $n+1$ en voor alle k met $0 \leq k \leq n+1$. Nu is voor $k = 0$ en $k = n+1$ de stelling opnieuw triviaal. Voor $1 \leq k \leq n$ geldt de stelling volgens de in-

duktieveronderstelling als de twee opeenvolgende woorden beide in $0 G(n)$ liggen of beide in $1 G(n)^R$. Het enige geval dat overblijft om te bewijzen is, dat het laatste kodewoord in $0 G(n)$ met k bits in precies 2 posities verschilt van het eerste kodewoord in $1 G(n)^R$ met k bits. Nu is, zoals we hierboven opmerkten, het eerstgenoemde woord gelijk aan $010^{n-k}1^{k-1} = 010^{n-k}1^{k-2}$ en het tweede woord $110^{n-k+1}1^{k-2} = 110^{n-k}01^{k-2}$ als $k \geq 2$. In het geval $k = 1$ zijn die woorden respectievelijk 010^{n-1} en 100^{n-1} . In beide gevallen verschillen de woorden in precies 2 bits.

Voorbeeld 10

Uit de Gray kode $G(4)$ leidt men direkt af de deelrij $G(4,2)$ van woorden die precies 2 enen bevatten:

0	0	0	0		1	1	0	0	(*)
0	0	0	1		1	1	0	1	
0	0	1	1	(*)	1	1	1	1	
0	0	1	0		1	1	1	0	
0	1	1	0	(*)	1	0	1	0	(*)
0	1	1	1		1	0	1	1	
0	1	0	1	(*)	1	0	0	1	(*)
0	1	0	0		1	0	0	0	

Men kan verifiëren dat de met (*) aangegeven woorden inderdaad uit elkaar volgen door iedere keer precies 2 bits te veranderen. $\square$

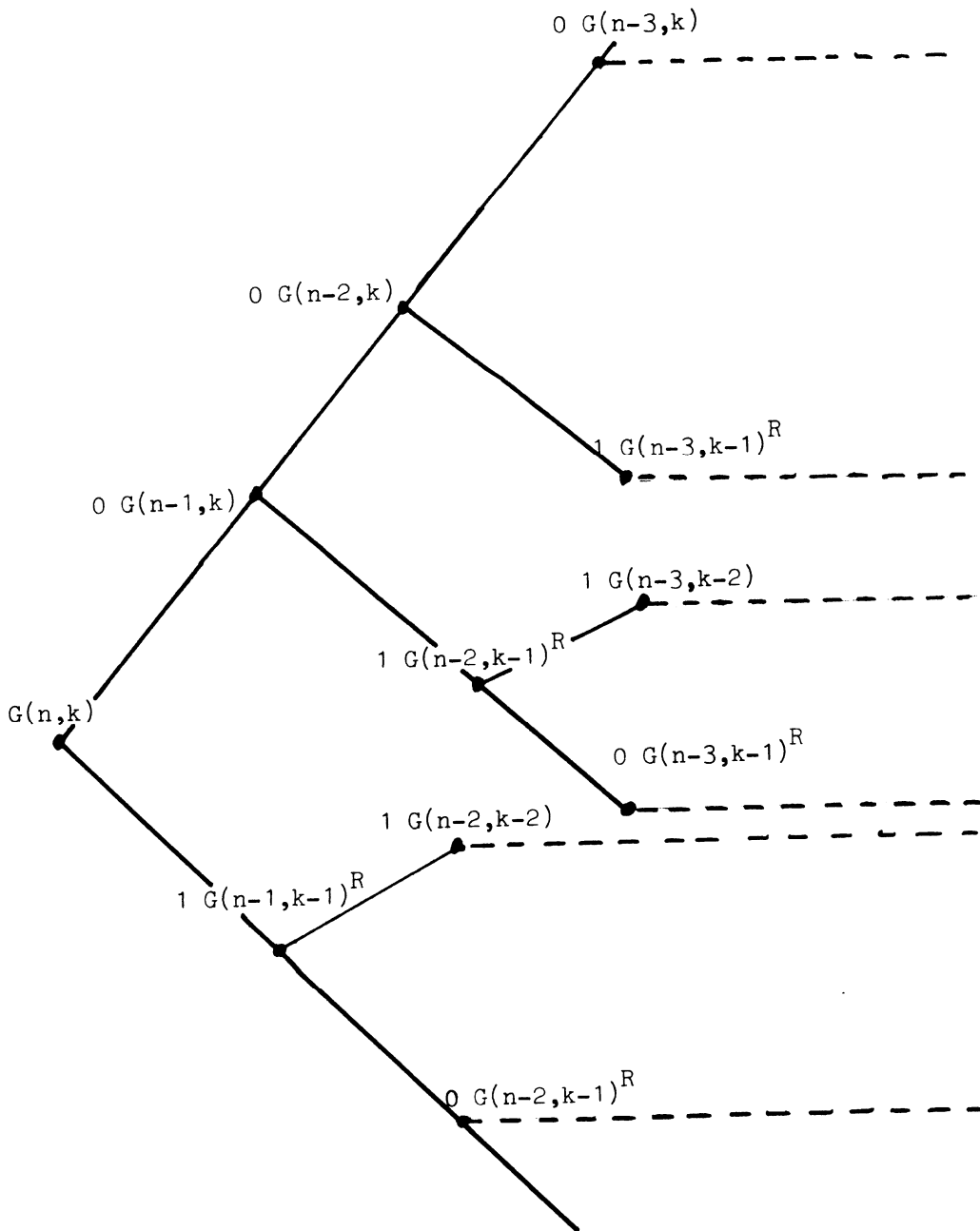
We zullen het rekursieve karakter van $G(n,k)$ zoals is vastgelegd in (3.22)-(3.24), illustreren met behulp van een binaire boom (zie 2.3.3). De knopen van de boom zijn kodewoorden van de "gedaante" $g_i G(i-1,t)$ (of, afhankelijk van de konstruktie, $g_i G(i-1,t)^R$), met $g_i \in \{0,1\}$ en $n > i \geq t+1 \geq 1$. Er is een tak tussen twee kodewoorden als de één uit de ander ontstaat door toepassing van (3.22). Omdat daar op twee manieren nieuwe kodewoorden worden gekonstrueerd, betekent dit steeds twee takken naar knopen op een "hoger" niveau. Omdat we de boom horizontaal tekenen, met de wortel aan de linker kant, spreken we in dit verband ook over boven- en ondertakken. Denk er verder aan, dat met (3.22) correspondeert

$$G(n,k)^R = \begin{pmatrix} 1 G(n-1,k-1) \\ 0 G(n-1,k)^R \end{pmatrix}, \quad n > k > 0. \quad (3.25)$$

De bladeren van de boom komen overeen met woorden $0 G(m,m)$ of $1 G(m,0)^R$.

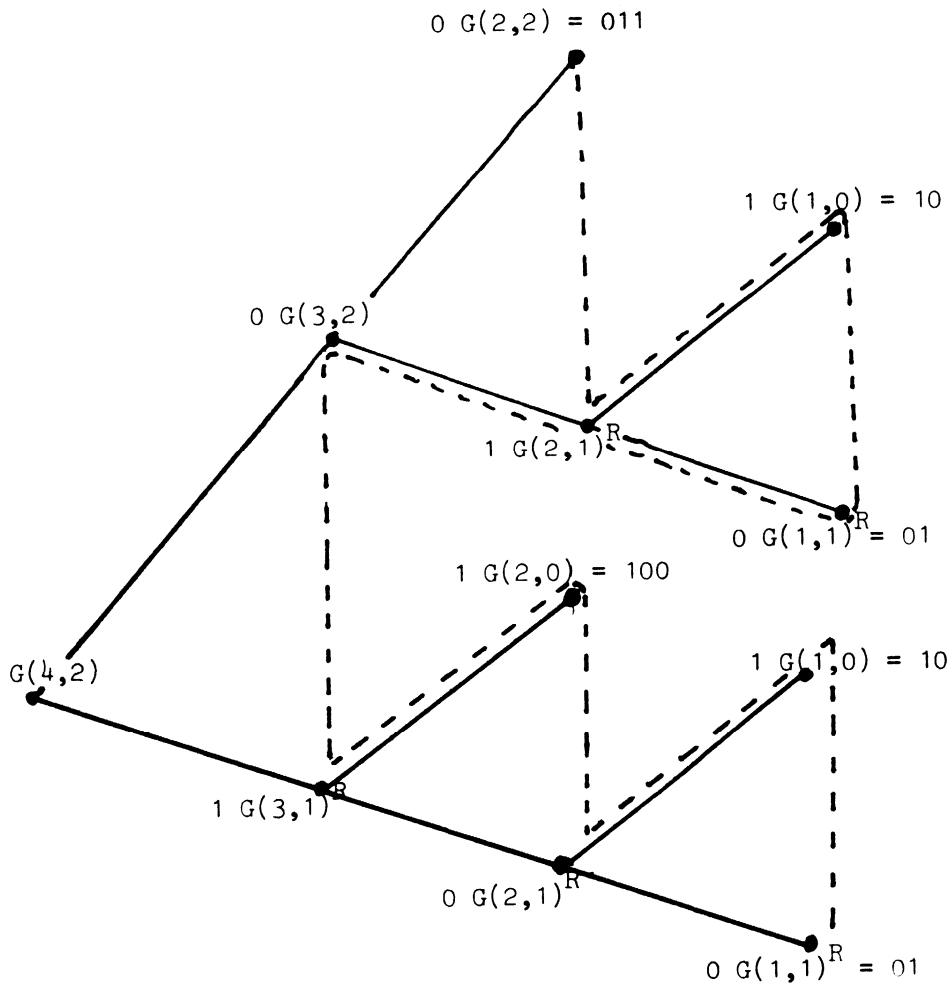
De gedaante hiervan laat zich onmiddellijk opschrijven met (3.23) en (3.24).

We tekenen nu een gedeelte van de boom van $G(n,k)$.



Vanuit elk blad is er één pad (zie ook hoofdstuk 4) naar de wortel $G(n,k)$. Dit pad correspondeert met (de konstruktie van) een kodewoord. Een algoritme, die alle kodewoorden genereert zal dus achtereenvolgens alle bladeren moeten aflopen en de bijbehorende woorden moeten afdrukken. Dit aflopen kan, door uitgaande van het lopende blad (kodewoord) in de boom naar links te bewegen, daarbij steeds ondertakken volgend, totdat men komt in een knoop, die zelf eindpunt is van een boventak. Ga dan over op de knoop, die eindpunt is van de bijbehorende ondertak (vertikaal naar beneden) en beweeg vervolgens naar rechts, daarbij steeds boventakken volgend.

Voorbeeld 11



□

Het pad van de wortel naar het "lopende" blad zullen we bijhouden door middel van het bijbehorende kodewoord $g_n g_{n-1} \dots g_1$.

Tevens houden we een stapel bij voor de knopen $g_i G(i-1, t)$, waar we naar terug moeten keren, wanneer we ons naar links bewegen in de boom. (Knopen van het type $g_i G(i-1, t)^R$ zijn altijd eindpunten van ondertakken en zullen slechts vertikaal bereikt worden). Voldoende is om de waarde van i op die stapel bij te houden. De bijbehorende waarde van t in $g_i G(i-1, t)$ is gelijk aan het aantal enen in de rij $g_{i-1}, g_{i-2}, \dots, g_1$ en kan dus eenvoudig berekend worden.

Elke keer als we vertikaal bewegen in de boomstructuur gaan we over naar het volgende kodewoord. Volgens de theorie moeten er dan twee bits veranderd worden. Deze veranderingen hangen af van de waarden van g_i en t . We onderscheiden daar-

bij 4 gevallen. De aard der veranderingen volgt gemakkelijk uit de beweringen gemaakt in het begin van het bewijs van stelling (3.10). We geven hier een schematisch overzicht. Het beginbit in onderstaande woorden is steeds g_i .

1. $g_i = 0, t > 1$

$$\begin{array}{ccccccc} \text{.....} & 0 & 1 & 0^{i-t-1} & 1 & 1^{t-2} \\ & & & \downarrow & & \\ \text{.....} & 1 & 1 & 0^{i-t-1} & 0 & 1^{t-2} \end{array} ;$$

g_i en g_{t-1} veranderen.

2. $g_i = 0, t = 1$

$$\begin{array}{ccccccc} \text{.....} & 0 & 1 & 0^{i-2} \\ & & & \downarrow & & \\ \text{.....} & 1 & 0 & 0^{i-2} \end{array} ;$$

g_i en g_{i-1} veranderen.

3. $g_i = 1, t > 0$

$$\begin{array}{ccccccc} \text{.....} & 1 & 1 & 0^{i-t-2} & 0 & 1^{t-1} \\ & & & \downarrow & & \\ \text{.....} & 0 & 1 & 0^{i-t-2} & 1 & 1^{t-1} \end{array} ;$$

g_i en g_t veranderen.

4. $g_i = 1, t = 0$

$$\begin{array}{ccccccc} \text{.....} & 1 & 0 & 0^{i-2} \\ & & & \downarrow & & \\ \text{.....} & 0 & 1 & 0^{i-2} \end{array} ;$$

g_i en g_{i-1} veranderen.

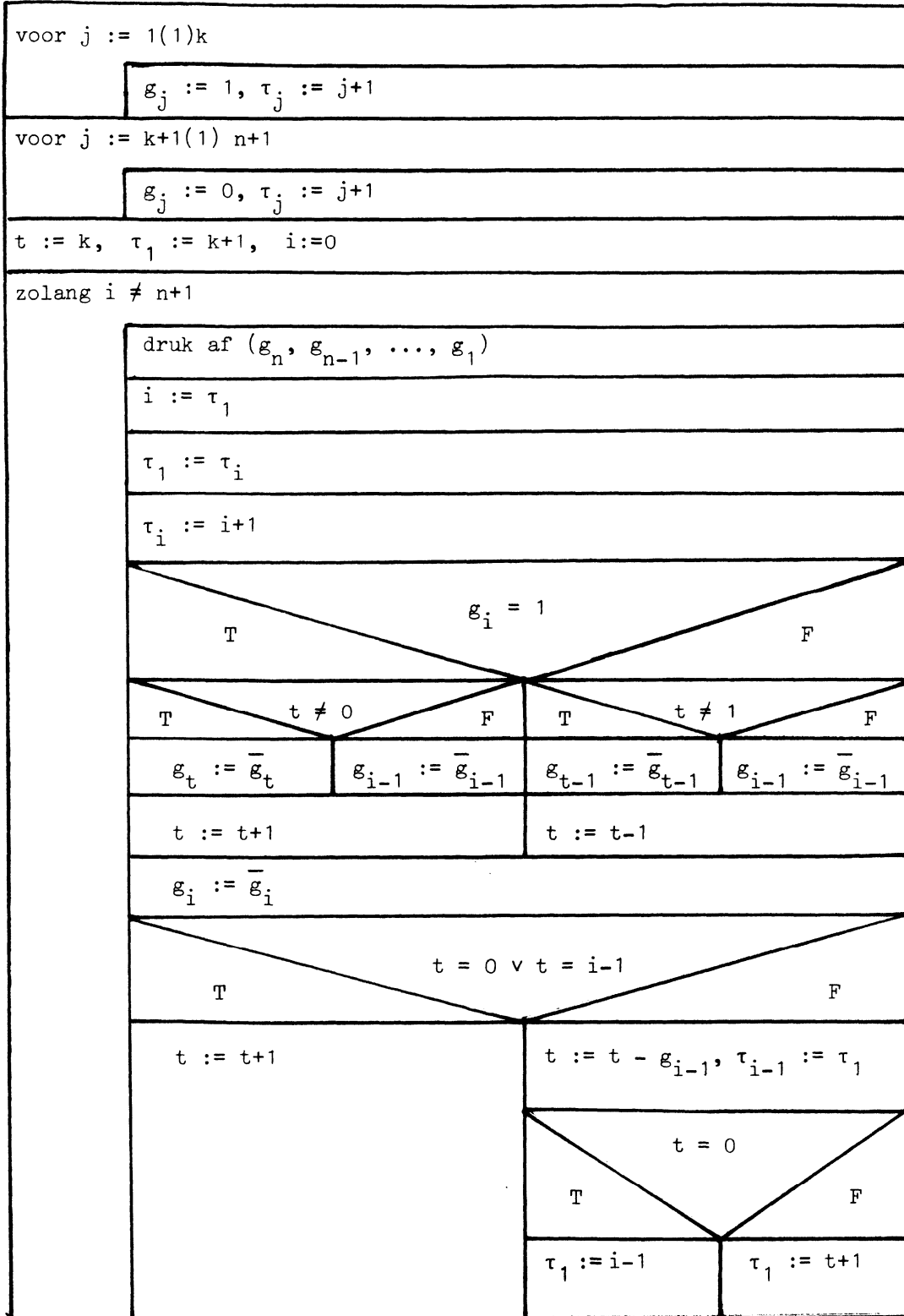
Bij elk van deze transformaties verandert het aantal enen rechts van g_i en dus de waarde van t . Voor $g_i = 0$ moeten we daarom t met 1 verminderen en voor $g_i = 1$ met 1 vermeerderen. Nadat één der vier transformaties is uitgevoerd hebben we een nieuw kodewoord, dat kan worden afgedrukt.

Echter moet de stapel nog worden aangepast. De top van de stapel wijst - zoals gezegd - steeds de i -waarde aan van het bit, dat in de volgende herhaling de rol

van g_i zal vervullen. De waarde van $i-1$ geeft dus de "terugkeerknoop" $G(i-1, t)$ aan, dat wil zeggen, de knoop van waaruit we ons in de boom bij de volgende herhaling vertikaal naar beneden zullen bewegen, om het volgende woord te bereiken. Als de knoop waarin we, na één der vier transformaties toegepast te hebben, zijn terechtgekomen een blad is ($t = 0$ of $t = i-1$), dan hoeft de top niet veranderd te worden. Is $t \neq 0$ en $t \neq i-1$, dan kunnen we in de boomstructuur naar rechts bewegen langsboventakken. In dit geval moeten de waarden $i-1, i-2, \dots, t+1$ op de stapel worden gezet, tenzij $g_{i-2} = \dots = g_1 = 0$, in welk geval men kan volstaan met $i-1$ op de stapel te zetten. (Een voorbeeld van deze laatste situatie is de knoop $G(3, 1)^R$ in voorbeeld 6). In alle gevallen moet ook steeds de nieuwe t -waarde worden uitgerekend. Voor $t = 0$ of $t = i-1$ wordt de nieuwe t -waarde gelijk aan $t+1$ (ga na, dat er één 1-bit bij is gekomen rechts van g_i). Als $t \neq 0$ en $t \neq i-1$ dan wordt t verminderd met het aantal 1-bits in de verzameling $\{g_{i-1}, g_{i-2}, \dots, g_{t+1}\}$. Het enige bit, dat mogelijk gelijk is aan 1 is g_{i-1} , dus $t := t - g_{i-1}$. Als nu geldt $t = 0$, dan zitten we kennelijk in de situatie $g_{i-2} = \dots = g_1 = 0$. Een en ander is verwerkt in algoritme 3.9 (Bitner, Ehrlich, Reingold). De stapel is op dezelfde wijze geïmplementeerd als in algoritme 3.7. Bitsveranderingen $g_j := 1 - g_j$ zijn eenvoudigheidshalve genoteerd als $g_j := \bar{g}_j$.

Opmerkingen

1. Algoritme 3.9 is korrekt voor $n \geq k > 0$. Een bewijs blijft achterwege.
2. Voor een kompleksiteitsbeschouwing van de algoritme wordt de lezer verwezen naar hoofdstuk 5 van [1].



Algoritme 3.9

4. GRAFEN EN ALGORITMEN

4.1. Inleiding

In de eerste twee paragrafen van dit hoofdstuk geven we een aantal definities en enige eenvoudige eigenschappen betreffende grafen. Bovendien bespreken we enkele manieren om grafen in computerprogramma's te representeren.

4.1.1 Definities

Een graaf bestaat in het algemeen uit een aantal punten (knopen) en een aantal verbindingen tussen die punten (takken). De grafentheorie is een belangrijk zelfstandig wiskundig onderwerp en wordt tevens in vele andere gebieden toegepast, zoals de topologie, de kombinatorische wiskunde, de operationele analyse, de informatica, de natuurkunde en de scheikunde.

Voorbeelden van grafen zijn meetkundige figuren met hun hoekpunten en ribben, een stadsplattegrond, een kommunikatienet, een chemische structuurformule, een elektrisch schakelschema, een prioriteitenschema van processen in een bedrijfssysteem enzovoort.

Het begrip graaf wordt wiskundig gepreciseerd door de volgende abstracte definitie.

Definitie

Een eindige graaf $G = (V, E)$ bestaat uit een eindige verzameling knopen $V = \{v_1, v_2, \dots, v_n\} (\neq \emptyset)$ en een eindige verzameling takken $E = \{e_1, e_2, \dots, e_m\}$ ($V \cap E = \emptyset$), zodanig dat met elke tak een geordend paar knopen (v_i, v_j) correspondeert.

Opmerkingen

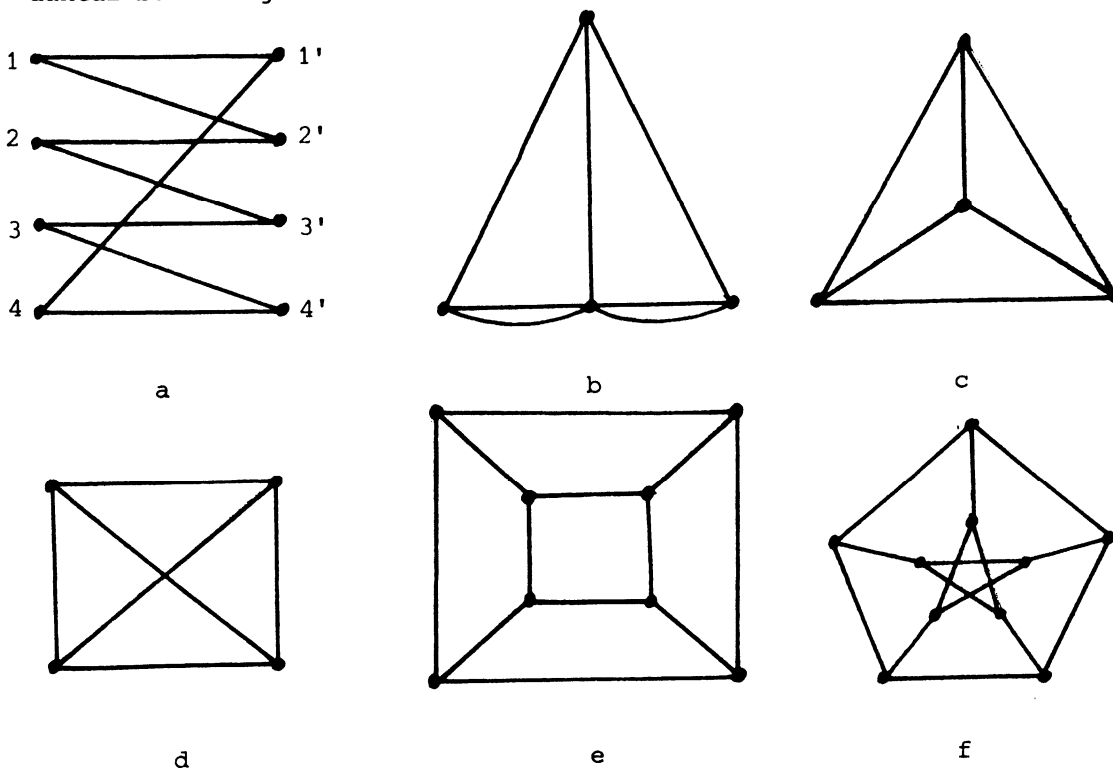
1. Op grond van bovenstaande definitie kan men ook zeggen, dat een graaf een afbeelding $\phi : E \rightarrow V \times V$ is.
2. In de Engelstalige literatuur wordt over vertices en edges gesproken. Vandaar de notatie V en E voor respectievelijk knopen en takken. Ook schrijft men wel VG en EG voor deze verzamelingen.

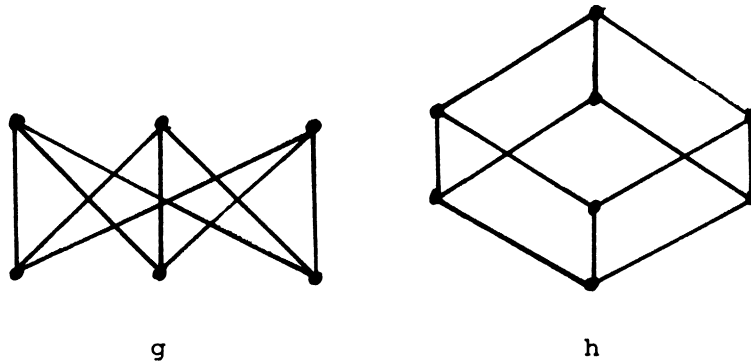
Een tak (a, b) heet incident met de knopen a en b . Knoop a heet de beginknoop en b heet de eindknoop. In een diagram voorziet men de tak (a, b) van een pijlrichting van a naar b . In dit verband spreekt men ook wel van een gerichte graaf of van een digraaf. Kent men geen richting toe aan de takken, dan spreekt men over een ongerichte graaf.

Het paar knopen (v_i, v_j) uit de definitie moet dan worden opgevat als een ongeordend paar $\{v_i, v_j\}$. Spreekt men zonder meer over een graaf dan bedoelt men meestal een ongerichte graaf. Twee knopen, die incident zijn met dezelfde tak heten naburig. Een knoop a , kan naburig zijn met zichzelf door middel van een tak (a, a) . Een dergelijke tak heet een lus (loop). Een knoop die met geen enkele tak incident is heet een geïsoleerde knoop. De definitie sluit niet uit dat een paar knopen (a, b) incident is met meer dan één tak. Zo'n stel takken heet parallel. Ook spreekt men wel over een meervoudige verbinding. In een gerichte graaf heten takken (a, b) en (b, a) anti-parallel. Een graaf heet enkelvoudig (simple) als hij géén lussen en géén parallelle of anti-parallelle takken bevat. Een graaf heet gelabeld als men aan elke knoop of aan elke tak een element toevoegt uit een of andere verzameling (Vaak is die verzameling $\mathbb{R}$ of $\mathbb{Z}$.) De elementen van die verzameling heten labels. De aan takken toegevoegde labels heten ook wel gewichten.

In communicatie- of transportnetwerken stellen zulke gewichten vaak fysische grootheden voor, zoals kosten, weglengte, capaciteit, enzovoort.

Alvorens verder te gaan laten we eerst de diagrammen volgen van een aantal bekende grafen.



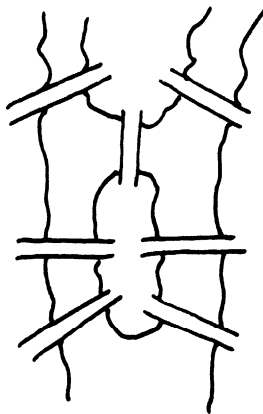


Figuur 4.1

Graaf a wordt wel gebruikt in de informatietheorie.

Vier signalen (de knopen 1,2,3 en 4) worden uitgezonden via een communicatiekanaal en vier korresponderende signalen (1',2',3' en 4') worden opgevangen. Wegens storingen kan een signaal ook verkeerd worden geïnterpreteerd. Zo kan het signaal 1 opgevangen worden als 1' of als 2' en analoog voor de overige signalen.

Graaf b is de abstrahering van het beroemde Königsberger bruggenprobleem (zie figuur 4.2)



Figuur 4.2

Euler bewees, dat het onmogelijk is, om een wandeling te maken, zó dat men elk der zeven bruggen slechts éénmaal passeert (een Eulerpad).

In graaf c is elk van de vier knopen verbonden met elke andere knoop.

Dit is een voorbeeld van een volledige graaf K_n , waarin deze eigenschap geldt voor alle n knopen. Ook is graaf c op te vatten als de graaf van een tetraëder.

Graaf d is eveneens een realisatie van K_4 , hoewel zijn uiterlijke gedaante verschilt met die uit c. Men zegt, dat de grafen c en d isomorf zijn.

In het algemeen zijn twee grafen G_1 en G_2 isomorf als er bijekties $f : VG_1 \rightarrow VG_2$ en $g : EG_1 \rightarrow EG_2$ bestaan zodanig, dat voor elke tak $e \in EG_1$ met begin- en eindknoop u respectievelijk v geldt, dat $f(u)$ en $f(v)$ begin- en eindknoop zijn van $g(e)$. Men kan ook zeggen, dat de bijekties f en g de incidenties van knopen en takken invariant laten.

Graaf e is de graaf die hoort bij de kubus.

Graaf f is de zogenaamde Petersengraaf. Deze speelt een grote rol in de algebraïsche grafentheorie.

Graaf g is de utiliteitsgraaf en treedt op in allerlei puzzels. De puzzel komt er altijd op neer, dat men moet nagaan of er een isomorfe graaf bestaat met de eigenschap, dat takken elkaar slechts snijden in knopen.

Graaf h is isomorf met graaf e .

Opmerkingen

1. Graaf c en e zijn voorbeelden van een Platonische graaf. De knopen van een dergelijke graaf zijn de hoekpunten en de takken zijn de ribben van één der vijf regelmatige veelvlakken in R_3 (de tetraëder, de kubus, de octaëder, de dodecaëder en de icoosaëder).
2. Graaf g is een voorbeeld van een bipartite graaf. Zo'n graaf wordt gedefinieerd als een graaf met $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$, terwijl elke tak e incident is met een knoop uit V_1 en een knoop uit V_2 en men noteert $G(V_1, V_2)$.
Is $G(V_1, V_2)$ enkelvoudig en is elke knoop in V_1 naburig met elke knoop in V_2 , dan is $G(V_1, V_2)$ een volledige bipartite graaf. Als $|V_1| = p$ en $|V_2| = q$, dan noteert men $G(V_1, V_2)$ in dat geval met $K_{p,q}$. Graaf g beeldt dus $K_{3,3}$ af.

Definitie

Een graaf G' is een partiële deelgraaf van de graaf G als geldt:

- (i) $VG' \subset VG$ en $EG' \subset EG$;
- (ii) elke tak van EG' is slechts incident met knopen van VG' .

Een bijzondere deelgraaf krijgt men door een deelverzameling $\bar{V} \subset VG$ te nemen en alle takken uit EG , die slechts incident zijn met knopen uit $\bar{V}$. Zo'n graaf heet een knoopdeelgraaf van G . Analoog definieert men een takdeelgraaf.

Tenslotte geven we in deze paragraaf een definitie van het begrip graad of valentie van een knoop.

Definitie

In een gerichte graaf is de ingraad $\rho^-(v)$ van een knoop v het aantal takken, dat die knoop als eindknoop heeft, terwijl de uitgraad $\rho^+(v)$ het aantal takken is met v als beginknoop.

In een ongerichte graaf is $\rho^+(v) = \rho^-(v)$ en spreken we kortweg over de graad $\rho(v)$.

Hebben alle knopen van een ongerichte graaf dezelfde graad, dan heet de graaf regulier.

Zonder bewijs delen we nog onderstaande eigenschap mee.

Stelling (4.1) (handshaking - theorema)

De som van de graden van de knopen in een ongerichte graaf is gelijk aan twee maal het aantal takken in die graaf.

4.1.2. Paden, afstanden en componenten

Allereerst definiëren we het begrip pad.

Definitie

Een pad van a naar b in een graaf G ($a, b \in VG$) is een rij takken $(v_0, v_1), (v_1, v_2), \dots, (v_{n-2}, v_{n-1}), (v_{n-1}, v_n)$, waarin $v_0 = a$ en $v_n = b$.

Definitie

Een pad heet een enkelvoudig (simple) pad, als elke tak van G er hoogstens één keer in voorkomt.

Definitie

Een pad heet een elementair pad of keten, als elke knoop van G er hoogstens één keer in voorkomt.

Definitie

Een circuit is een pad, waarvan de beginknoop dezelfde is als de eindknoop. Een circuit heet enkelvoudig als elke tak van G er hoogstens één keer in voorkomt en elementair als elke knoop er hoogstens één keer in voorkomt, behalve de beginknoop.

Opmerkingen

1. Bovenstaande definities gelden voor zowel gerichte als ongerichte grafen. Is de graaf ongericht, dan mag men een tak $\{v_i, v_j\}$ zowel als (v_i, v_j) als wel als (v_j, v_i) in een pad of in een circuit opnemen. Is het pad of circuit enkelvoudig, dan mag maar één van beide mogelijkheden optreden. In alle gevallen heeft het pad (of het circuit) een richting, zoals de termen begin- en eindknoop al aangeven.

2. In een circuit kan men elke knoop als beginknoop nemen, om toch dezelfde "cykel" van takken te verkrijgen. Hebben twee circuits dezelfde takken in dezelfde cyclische volgorde, dan beschouwen we ze als identiek, ook al hebben ze verschillende beginknopen.

3. Vanzelfsprekend is elk elementair pad (circuit) ook enkelvoudig.

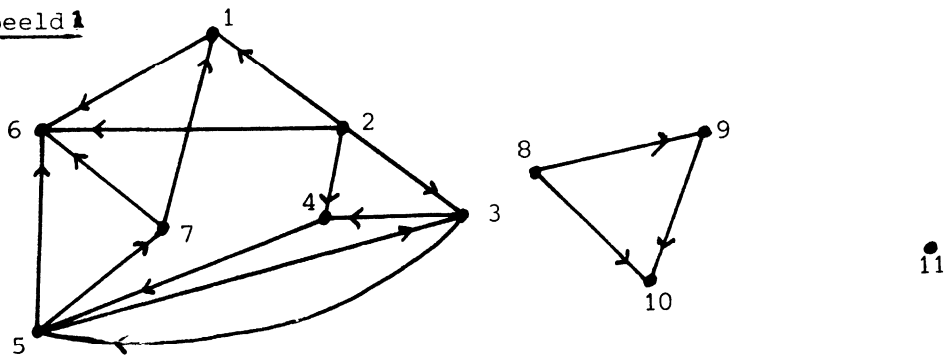
Definitie

Een Eulerpad (Eulercircuit) (zie 4.1.1) is een enkelvoudig pad (circuit), dat alle takken van een graaf bevat.

Definitie

De lengte van een pad is het aantal takken van dat pad. Als een pad niet enkelvoudig is wordt elke tak zo vaak geteld als hij in het pad voorkomt.

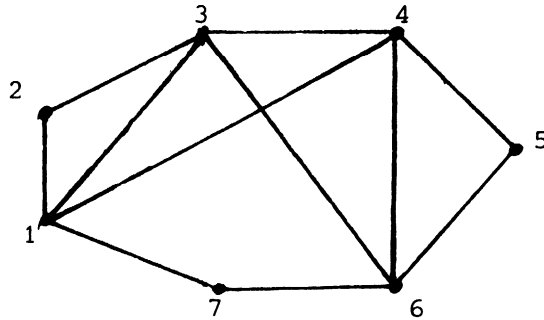
Voorbeeld 1



In bovenstaande digraaf is de rij takken $(v_3, v_4), (v_4, v_5), (v_5, v_7), (v_7, v_1)$, die we korthedshalve weergeven als $(v_3, v_4, v_5, v_7, v_1)$, een enkelvoudig en zelfs een elementair pad ter lengte 4 van v_3 naar v_1 . De rij (v_3, v_5, v_3) is een pad ter lengte 2 van v_3 naar v_3 en is bovendien een voorbeeld van een enkelvoudig (en elementair) circuit. De rij $(v_3, v_5, v_3, v_5, v_3, v_4)$ is een pad van v_3 naar v_4 ter lengte 5. De rij (v_3, v_4, v_5, v_3) is een elementair circuit van lengte 3. De rij (v_4, v_5, v_3, v_4) is hetzelfde circuit.

Merk op, dat er geen pad is van v_1 naar v_8 , en ook niet van v_1 naar v_2 of van v_1 naar v_3 ! $\square$

Voorbeeld 2



In deze ongerichte graaf stellen

$$(v_1, v_3, v_4, v_1, v_2, v_3, v_6, v_7, v_1)$$

en

$$(v_2, v_3, v_6, v_7, v_1, v_3, v_4, v_1, v_2)$$

hetzelfde enkelvoudige circuit voor.

Het enkelvoudige circuit

$$(v_1, v_3, v_4, v_1, v_7, v_6, v_3, v_2, v_1)$$

bevat dezelfde takken als het eerste circuit, maar in een andere volgorde. Volgens onze definities is dit dus een ander circuit.

De circuits

$$(v_2, v_3, v_4, v_1, v_2)$$

en

$$(v_2, v_1, v_4, v_3, v_2)$$

zijn verschillende elementaire circuits van lengte 4.

Het circuit

$$(v_2, v_3, v_1, v_4, v_3, v_2)$$

is niet enkelvoudig (zie opmerking 1 van deze paragraaf).

De circuits

$$(v_2, v_3, v_4, v_5, v_6, v_7, v_1, v_3, v_6, v_4, v_1, v_2)$$

en

$$(v_1, v_3, v_6, v_7, v_1, v_2, v_3, v_4, v_6, v_5, v_4, v_1)$$

zijn twee verschillende Eulercircuits. $\square$

Definitie

De afstand van knoop a naar knoop b in een graaf is de lengte van het kortste pad van a naar b , als $a \neq b$ is. Wanneer $a = b$ dan is de afstand 0. We noteren de afstand van a naar b met $d(a,b)$.

In voorbeeld 1 is de afstand van v_2 naar v_5 gelijk aan 2 en van v_2 naar v_6 gelijk aan 1. De afstand van v_1 naar v_2 is niet gedefinieerd, maar wordt bij praktische toepassingen vaak gelijkgesteld aan ∞ .

Bij elke digraaf hoort een ongerichte graaf, die men verkrijgt door in plaats van de geordende paren knopen (v_i, v_j) de ongeordende paren $\{v_i, v_j\}$ als takken te beschouwen. (N.B. als de digraaf circuits ter lengte 2 heeft, bezit de ongerichte graaf tweevoudige takken.)

In een diagram laat men dan eenvoudig de pijlrichtingen weg. Doet men dit met de graaf uit voorbeeld 1, dan is de afstand van v_1 naar v_2 wel gedefinieerd en gelijk aan 1. Omdat richtingen nu geen rol meer spelen, spreekt men eenvoudig over de afstand tussen v_1 en v_2 . Ook nu is er geen pad van v_1 naar v_8 of van v_8 naar v_1 .

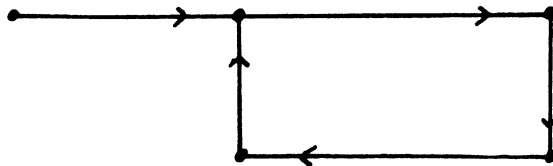
In een graaf (gericht of ongericht) heten twee knopen verbonden (connected) als er een pad van de ene naar de andere is. Een ongerichte graaf heet samenhangend (connected) als elk tweetal knopen verbonden is. De graaf, die bestaat uit één enkele knoop is per definitie ook samenhangend. Een gerichte graaf heet samenhangend, als de bijbehorende ongerichte graaf samenhangend is.

Een gerichte graaf heet sterk samenhangend (strongly connected) als er voor elk tweetal knopen a en b een pad is van a naar b en ook van b naar a .

Zij $\bar{G}$ een deelgraaf van G ($\bar{G} \subseteq G$). We spreken van een echte deelgraaf als $\bar{G} \neq G$, en schrijven $\bar{G} \subset G$. Als $\bar{G}$ samenhangend is en als er geen

samenhangende deelgraaf $\overline{\overline{G}}$ van G bestaat met $\overline{G} \subset \overline{\overline{G}} \subset G$, dan heet $\overline{G}$ maximaal samenhangend. Een dergelijke deelgraaf noemt men ook wel komponent van G . Uit de definitie van samenhangende graaf volgt, dat een niet-samenhangende graaf uit twee of meer componenten bestaat. De graaf uit voorbeeld 1 bestaat uit drie componenten.

Voorbeeld 3



Deze digraaf is samenhangend, maar niet sterk samenhangend. □

Zonder bewijs geven we nog enkele elementaire stellingen betreffende paden in grafen.

Stelling (4.2)

In een (gerichte of ongerichte) graaf met $|VG| = n$ is de afstand van een knoop naar een daarmee verbonden knoop hoogstens gelijk aan $n-1$.

Definitie

Het werkelijke maximum van de afstanden tussen twee knopen in een ongerichte graaf G heet de diameter van G .

Stelling (4.3)

Een ongerichte samenhangende graaf G bezit een Eulerpad dan en slechts dan als G 0 of 2 knopen heeft van oneven graad.

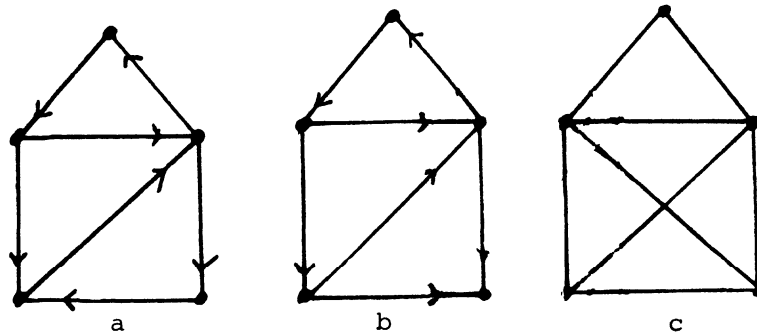
Stelling (4.4)

Een gerichte samenhangende graaf G bezit een Eulerpad dan en slechts dan als $\rho^+(v) = \rho^-(v)$ voor alle $v \in VG$, met een mogelijke uitzondering voor precies twee knopen v_1 en v_2 . Hiervoor moet dan gelden $\rho^+(v_1) = \rho^-(v_1)+1$ en $\rho^-(v_2) = \rho^+(v_2)+1$.

Opmerking

Als G géén knopen van oneven graad heeft in stelling (4.3) of als $\rho^+(v) = \rho^-(v)$ voor alle $v \in VG$ in stelling (4.4) vallen begin- en eindknoop van een Eulerpad samen en heeft men dus een Eulercircuit.

Voorbeeld 4

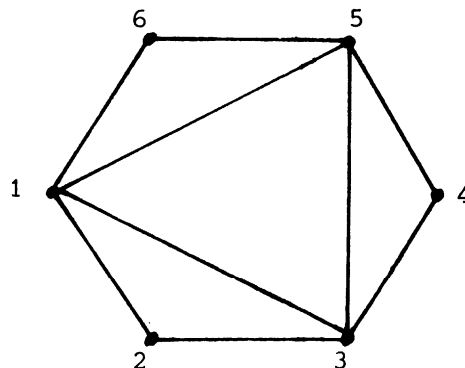


De digraaf in a heeft een Eulerpad, maar de digraaf in b heeft dat niet. De ongerichte graaf in c heeft geen Eulercircuit, maar wel een Eulerpad. $\square$

4.1.3. Circuits in enkelvoudige ongerichte grafen

In voorbeeld 2 zagen we enige enkelvoudige en elementaire circuits in een enkelvoudige ongerichte graaf. In deze paragraaf geven we hiervan nog een paar voorbeelden, waarbij de vraag centraal staat hoeveel verschillende circuits van een bepaald type mogelijk zijn.

Voorbeeld 5



In bovenstaande graaf zijn

$(v_1, v_2, v_3, v_4, v_5, v_6, v_1)$ en $(v_1, v_6, v_5, v_4, v_3, v_2, v_1)$ verschillende elementaire circuits van lengte 6. Het is direkt duidelijk dat er niet nog meer elementaire circuits van lengte 6 zijn.

Vragen we naar enkelvoudige circuits van lengte 6, dan hebben we behalve de twee reeds genoemde circuits nog :

$$(v_1, v_6, v_5, v_1, v_2, v_3, v_1) \quad ,$$

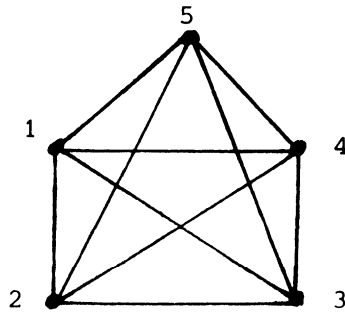
$$(v_1, v_6, v_5, v_1, v_3, v_2, v_1) \quad ,$$

$$(v_1, v_5, v_6, v_1, v_2, v_3, v_1) \quad ,$$

$$(v_1, v_5, v_6, v_1, v_3, v_2, v_1) \quad ,$$

plus nog eens tweemaal vier circuits betrekking hebbende op de paren driehoeken $\{v_1, v_2, v_3\}$ $\{v_3, v_4, v_5\}$ respektievelijk $\{v_1, v_5, v_6\}$, $\{v_3, v_4, v_5\}$. Er zijn dus in totaal $2 + 3 \times 4 = 14$ enkelvoudige circuits van lengte 6. $\square$

Voorbeeld 6



Gegeven is de volledige graaf K_5 .

Gevraagd wordt de diameter van deze graaf te bepalen en het totale aantal elementaire circuits van lengte 4.

Oplossing

Aangezien twee verschillende knopen altijd op afstand 1 van elkaar liggen is de maximale afstand, dus de diameter, ook gelijk aan 1.

Een elementair circuit van lengte 4 bevat 4 knopen, zeg v_1, v_2, v_3 en v_4 .

Uit de definities aan het begin van 4.1.2 volgt dat de volgorde van de takken in een circuit relevant is. Dus horen bij de knopen v_1, v_2, v_3 en v_4 de volgende zes verschillende circuits:

$(v_1, v_2) , (v_2, v_3) , (v_3, v_4) , (v_4, v_1) ;$

$(v_1, v_2) , (v_2, v_4) , (v_4, v_3) , (v_3, v_1) ;$

$(v_1, v_3) , (v_3, v_2) , (v_2, v_4) , (v_4, v_1) ;$

$(v_1, v_3) , (v_3, v_4) , (v_4, v_2) , (v_2, v_1) ;$

$(v_1, v_4) , (v_4, v_2) , (v_2, v_3) , (v_3, v_1) ;$

$(v_1, v_4) , (v_4, v_3) , (v_3, v_2) , (v_2, v_1) ;$

Uit 5 knopen kunnen we op $\binom{5}{4}$ manieren 4 verschillende knopen kiezen. Het gevraagde aantal is dus $6\binom{5}{4} = 30$.

Een andere manier om het gevraagde aantal circuits uit te rekenen is de volgende.

Zij v_1 een bepaalde knoop. De verzameling elementaire circuits van lengte 4 valt uiteen in twee deelverzamelingen, nl. die welke v_1 wél bevatten en die welke v_1 niet bevatten.

Beschouw de eerste deelverzameling. Uitgaande van v_1 kan men op 4 manieren de "eerste" tak van het circuit kiezen, op 3 manieren de tweede tak en op 2 manieren de derde tak. De laatste tak ligt nu vast. Het totale aantal van dit soort circuits is dus $4 \times 3 \times 2 = 24$.

Het totale aantal circuits die v_1 niet bevatten is volgens een analoge redenering (zie ook de eerste oplossing) gelijk aan $3 \times 2 = 6$. Volgens de somregel (zie hoofdstuk 1) is het gevraagde aantal gelijk aan $24 + 6 = 30$.

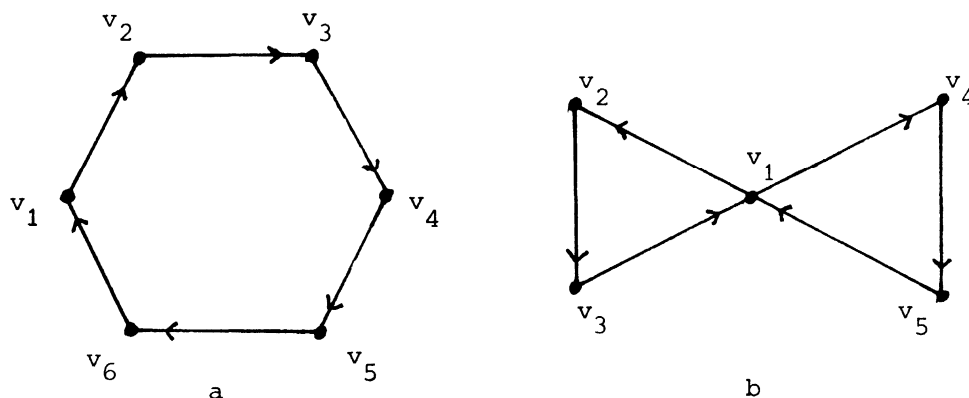
Was er gevraagd om het aantal elementaire circuits van lengte 4 te berekenen, zonder op de omlooprichting te letten, dan was het antwoord vanzelfsprekend $\frac{1}{2} \times 30 = 15$ geweest. $\square$

Voorbeeld 7

Bepaal het totale aantal enkelvoudige circuits van lengte 6 in de volledige graaf K_6 .

Oplossing

Het zal direkt duidelijk zijn dat de bedoelde circuits twee verschillende gedaanten kunnen hebben. Schematisch geven we deze typen weer als :



Circuits van type a zijn elementair, die van type b zijn dat niet. De knopen $v_1, v_2, \dots, v_6$ nemen waarden aan uit de verzameling $VK_6 := \{1, 2, \dots, 6\}$.

Om het totale aantal circuits van type a te tellen gaan we als volgt te werk. We kiezen eerst een waarde voor v_1 . Dit kan op 6 manieren. Vanuit v_1 kunnen we op 5 manieren een tak (v_1, v_2) kiezen. Vanuit v_2 kunnen we op 4 manieren een tak (v_2, v_3) kiezen, etc. Het totale aantal manieren om $v_1, v_2, \dots, v_6$ te kiezen is dus $6!$. Omdat in een circuit van type a alle knopen een "gelijkwaardige rol" spelen zullen alle circuits echter niet verschillend zijn. Zo zijn de circuits $(4, 1, 2, 6, 5, 3)$ en $(1, 2, 6, 5, 3, 4)$ identiek volgens opmer-

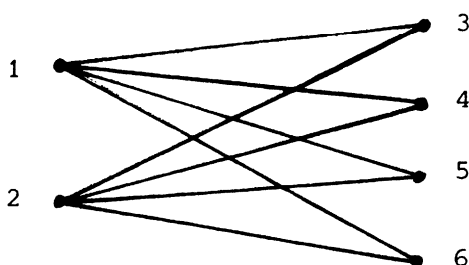
king 2 op p. 91. Elke cyclische permutatie van een rij waarden voor $v_1, v_2, \dots, v_6$ geeft aanleiding tot hetzelfde circuit. Dus is het totale aantal circuits van type a gelijk aan $\frac{1}{6} \cdot 6! = 5!$

Om een circuit van type b te konstrueren kiezen we eerst 5 knopen uit VK_6 . Dit kan op $\binom{6}{5}$ manieren. Voor v_1 hebben we nu 5 mogelijkheden. De tak (v_1, v_2) is vervolgens op 4 manieren te kiezen en (v_2, v_3) op 3 manieren. De tak (v_3, v_1) ligt nu vast. Tenslotte kunnen we nog op 2 manieren de tak (v_1, v_4) kiezen, waarna (v_4, v_5) en (v_5, v_1) weer vastliggen. Het totale aantal keuzen is dus $\binom{6}{5} \cdot 5 \cdot 4 \cdot 3 \cdot 2 = 6!$. Echter moeten we ook hier rekening houden met de symmetrie van de circuits. De knopen v_2 en v_4 spelen dezelfde rol evenals v_3 en v_5 . Zo zijn de circuits $(4, 1, 2, 4, 6, 5, 4)$ en $(4, 6, 5, 4, 1, 2, 4)$ identiek. Het totale aantal circuits van type b is dus $\frac{1}{2} \cdot 6!$

Het gevraagde aantal is $5! + \frac{1}{2} \cdot 6! = 480$. $\square$

Voorbeeld 8

Bepaal het totale aantal Eulercircuits van $K_{2,4}$.



Oplossing

Omdat een Eulercircuit elke tak van de graaf bevat, zal de tak tussen 1 en 3 zeker voorkomen en wel òf als $(1, 3)$ of als $(3, 1)$. We tellen nu eerst het aantal Eulercircuits die $(1, 3)$ bevatten. We gaan het circuit doorlopen, beginnende met deze tak. Vanuit 3 kunnen we slechts naar 2. Vandaar uit zijn er 3 takken mogelijk, waarna we weer naar 1 moeten, etc. Er zijn dus $3!$ Eulercircuits van dit type. Gaan we uit van $(3, 1)$ dan vinden we ook $3!$ mogelijkheden. Het totale aantal is dus $2 \cdot 3! = 12$. $\square$

Opmerkingen

1. De in de voorafgaande voorbeelden gevolgde methoden zijn niet eenvoudig te generaliseren voor willekeurige grafen.

2. Het tellen wordt aanzienlijk eenvoudiger als de volgorde van de takken er niet toe doet en het er alleen maar om gaat welke takken er in het circuit zitten.

Ga zelf na, dat het aantal circuits in voorbeeld 7 dan 240 wordt en in voorbeeld 8 vanzelfsprekend 1.

In 4.6.1. komt deze vraag in een ander verband opnieuw aan de orde.

4.1.4. Bomen

In deze paragraaf is G steeds een enkelvoudige graaf met k componenten. Verder stellen we $|VG| = n$ en $|EG| = m$. De aantallen k , n en m voldoen aan de volgende betrekkingen.

Stelling (4.5)

Voor een ongerichte, enkelvoudige graaf G geldt:

$$n - k \leq m \leq \frac{1}{2}(n-k)(n-k+1).$$

Bewijs

(i) De linker ongelijkheid bewijzen we eerst voor samenhangende grafen, dus voor het geval $k = 1$.

De relatie $m \geq n-1$ geldt zeker als $n = 1$, omdat er dan geen takken zijn. Stel de relatie geldt voor zekere $n \geq 1$.

Beschouw nu een graaf G met $n+1$ knopen. We verwijderen uit die graaf 1 knoop en alle takken incident met die knoop. Als dit aantal p is, geldt $p \geq 1$, want G was samenhangend. De nieuwe graaf heeft n knopen en m' takken, waarvoor geldt $m' \geq n-1$. Dan geldt echter ook $m = m' + p \geq n-1+p \geq n-1+1 = n$.

Door inductie naar n toe te passen is het bewijs hiermee geleverd voor het geval $k = 1$.

Het algemene geval $k \geq 1$ volgt nu door het bewezene toe te passen op elke component. Er geldt $n_i - 1 \leq m_i$ voor $1 \leq i \leq k$. Sommeren over i levert $n-k \leq m$.

(ii) Om de rechter ongelijkheid af te leiden bedenken we, dat één component een maximaal aantal takken heeft als hijzelf een volledige graaf is met $\frac{1}{2}n'(n'-1)$ takken, als n' het aantal knopen in deze component voorstelt. Voor $k = 1$ is hiermee de ongelijkheid aangetoond.

Stel vervolgens dat $k > 1$ is en beschouw de componenten C_i en C_j met respectievelijk n_i en n_j knopen. Neem aan dat $n_i \geq n_j > 1$ (als er niet zo'n component C_j te vinden is is de uitspraak triviaal). We gaan nu, bij vaste n en k , het aantal takken vergroten. Daartoe vervangen we allereerst C_i en C_j door de volledige grafen K_{n_i} en K_{n_j} .

Vervolgens brengen we van C_j één knoop over naar C_i en verbinden deze met alle overige knopen van C_i . Het totale aantal knopen blijft dan gelijk aan n , maar het aantal takken neemt toe met

$$\frac{1}{2}\{(n_i+1)n_i - n_i(n_i-1)\} - \frac{1}{2}\{n_j(n_j-1) - (n_j-1)(n_j-2)\} = n_i - n_j + 1 > 0.$$

De graaf met het maximale aantal takken (bij vaste n en k) bestaat dus uit een volledige graaf met $n-k+1$ knopen en verder nog $k-1$ geïsoleerde knopen. Het aantal takken in deze graaf is gelijk aan $\frac{1}{2}(n-k+1)(n-k)$. $\square$

Opmerking

Stelling (4.5) geldt ook voor gerichte enkelvoudige grafen G . Het bewijs van stelling (4.5) blijft namelijk van kracht voor de bij G behorende ongerichte graaf, die op grond van de definitie (zie 4.1.1) ook enkelvoudig is.

Zoals we zagen in het bewijs van stelling (4.5) bestaan er grafen, waarin m zijn bovengrens aanneemt. Er zijn ook voorbeelden van grafen waarin m zijn ondergrens aanneemt, zoals hieronder blijkt.

Definitie

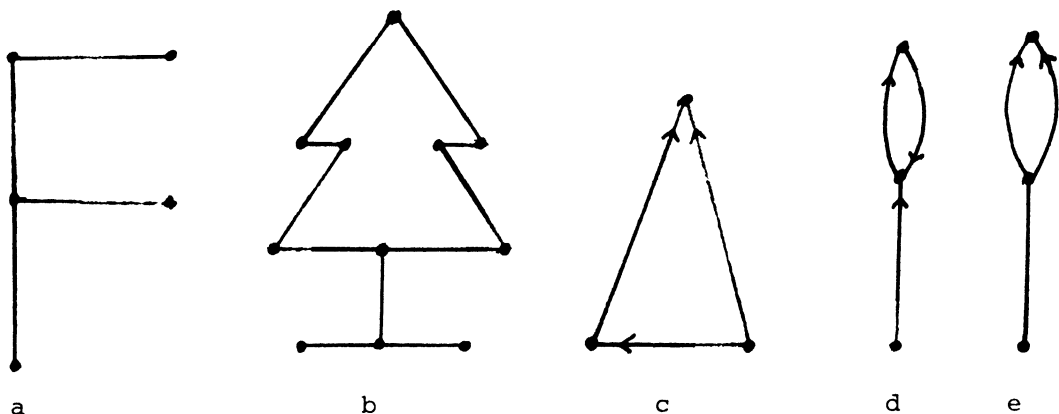
Een enkelvoudige, samenhangende graaf met n knopen en $n-1$ takken heet een boom. Een niet-samenhangende graaf, waarvan elke component een boom is noemen we een bos.

Stelling (4.6)

Een enkelvoudige, samenhangende graaf is dan en slechts dan een boom als de bijbehorende ongerichte graaf geen circuits bevat.

Bewijs deze eigenschap zelf.

Voorbeeld 9

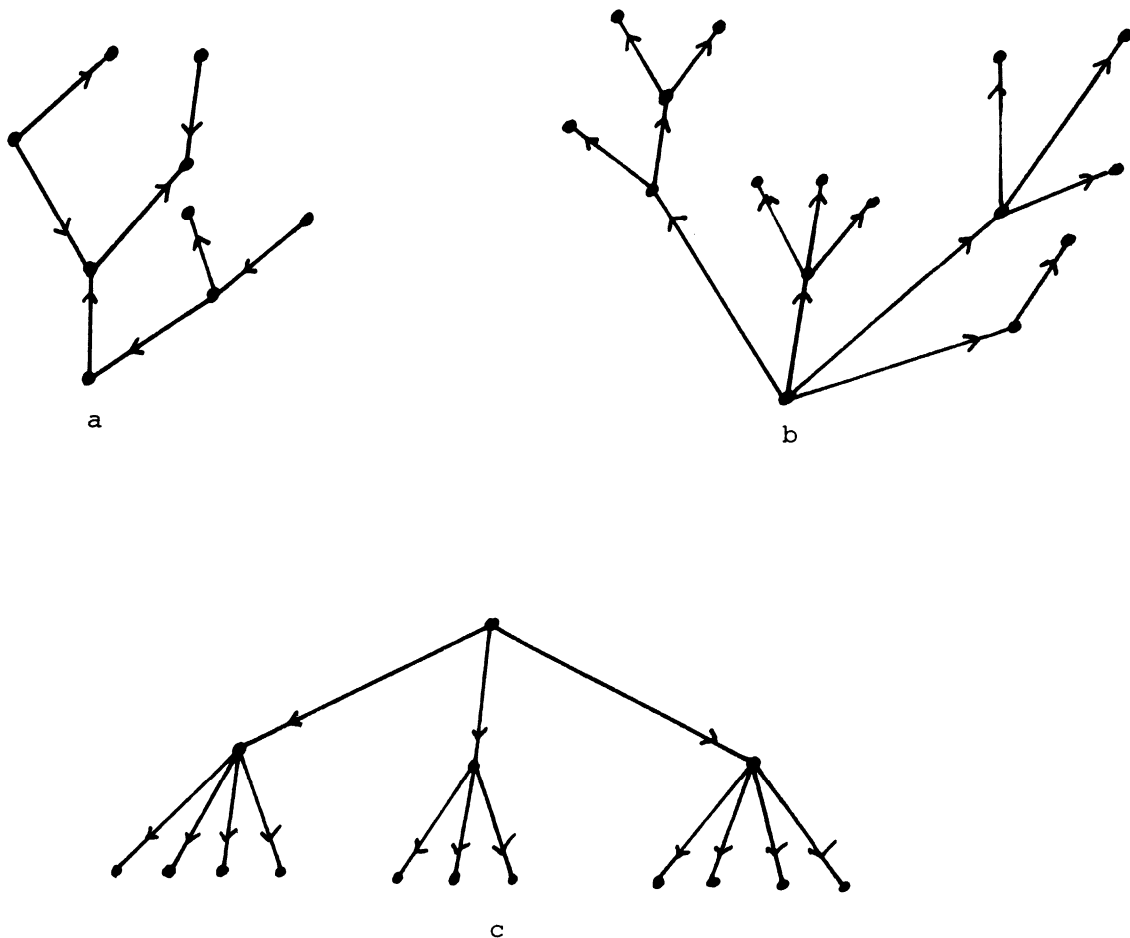


Graaf a is volgens stelling (4.6) een boom en de grafen b en c zijn dat volgens dezelfde stelling niet. De grafen d en e zijn geen bomen (aantal takken is niet minimaal), maar stelling (4.6) kan hierop niet worden toegepast, omdat de bijbehorende grafen niet enkelvoudig zijn. $\square$

Opmerking

Soms wordt het criterium uit stelling (4.6) als definitie-eigenschap voor bomen gebruikt.

Graaf a uit voorbeeld 9 is een ongerichte boom. We laten in onderstaande figuur nog enkele bijzondere bomen volgen



Figuur 4.3

Boom a is een gerichte boom. Deze bomen zijn van groot belang bij toepassingen in de informatica, netwerktheorie, speltheorie, telproblemen etc.

Bijzondere gerichte bomen zijn arborescenties. Dat zijn bomen met precies één knoop v , waarvoor geldt $\rho^-(v) = 0$; v heet de wortel van de boom. Boom b is hiervan een voorbeeld. Arborescenties waarvan de takken uitgaande van een knoop v geordend zijn, voor alle $v \in VG$, heten geordende bomen (zie boom c). Is bovendien $\rho^+(v) \leq 2$ voor alle $v \in VG$ dan spreekt men over een binaire boom.

Opmerking

Het begrip binaire boom kwam ter sprake in de hoofdstukken 2 en 3. Ga zelf na of de definitie, zoals gegeven in hoofdstuk 2, gelijkwaardig is met de beschrijving van hierboven.

Bewijs verder nog zelf de volgende stellingen.

Stelling (4.7)

Laat T een gerichte of ongerichte boom zijn. In de bijbehorende ongerichte boom $\overline{T}$ is elk tweetal knopen door precies één elementaire keten verbonden.

Stelling (4.8)

Toevoeging van één tak aan een boom geeft in de bijbehorende ongerichte graaf aanleiding tot precies één circuit.

4.1.5. Representatie van grafen in algoritmen

Een graaf kan op verschillende manieren gerepresenteerd worden in een algoritme. De implementatie zowel als de efficiency van een algoritme voor berekeningen in of met grafen kan sterk afhangen van die representatie.

We geven nu enkele mogelijkheden voor de datastructuur van een graaf (zie ook [6]).

Ook nu onderstellen we steeds, dat $n = |VG|$ en $m = |EG|$.

De knopen hebben als labels de natuurlijke getallen $1, 2, \dots, n$ en de takken de getallen $1, 2, \dots, m$.

- A. Een van de meest gebruikte manieren om een graaf zonder meervoudige takken te representeren is de nabuurmatrix of verbindingsmatrix.

Definitie

De nabuurmatrix A is een $(n \times n)$ -matrix, waarin $a_{ij} = 1$ als er een tak is van knoop i naar knoop j en $a_{ij} = 0$ als dit niet het geval is.

Een onmiddellijk gevolg van de definitie is, dat de nabuurmatrix van een ongerichte graaf symmetrisch is.

Voor graf en met meervoudige takken kan men op eenvoudige wijze de definitie generaliseren, zodanig dat a_{ij} het aantal takken aangeeft tussen i en j .

Bij opslag in een computergeheugen kan men elk matrixelement in een bit opslaan. Als elke rij van de matrix in een nieuw woord in het geheugen begint, is het aantal woorden nodig, om de hele graaf op te slaan gelijk aan $n \lceil \frac{n}{l} \rceil$, waarin l de woordlengte is. Is de graaf ongericht, dan kan men bijna de helft van deze ruimte uitsparen door gebruik te maken van de symmetrie.

In algoritmen moet men in het algemeen elke tak tenminste één keer onderzoeken. Dit houdt in, dat bij gebruik van een nabuurmatrix de rekentijd tenminste evenredig is met n^2 .

We geven nog een kombinatorische toepassing van de nabuurmatrix.

Stelling (4.9)

Zij G een gerichte of ongerichte graaf zonder parallelle takken. Het aantal paden van lengte k van knoop i naar knoop j is gelijk aan het (i, j) -element van de matrix A^k .

Bewijs

Het resultaat is juist voor $k = 0$, omdat $A^0 = I$.

Stel het resultaat is juist voor zekere $k \in \mathbb{N}$. Uit de relatie

$$(A^{k+1})_{ij} = \sum_{h=1}^n (A^k)_{ih} a_{hj}$$

volgt, dat het resultaat dan ook geldt voor $k + 1$. Immers een pad van lengte $k + 1$ van i naar j bestaat altijd uit een pad van lengte k van i naar een nabuur h van knoop j plus de tak (h, j) . $\square$

Ga voorts nog na dat voor een graaf met nabuurmatrix A geldt

$$\rho^+(v_i) = \sum_{j=1}^n a_{ij} \quad (\text{rijsommen})$$

en

$$\rho^-(v_j) = \sum_{i=1}^n a_{ij} \quad (\text{kolomsommen}).$$

- B. Een graaf waarin elke tak is gelabeld met een label uit $\mathbb{R}$ of een deelverzameling daarvan wordt vaak voorgesteld door de zogenaamde gewichtenmatrix.

Definitie

De gewichtenmatrix W van een takgelabelde graaf G is een $(n \times n)$ -matrix, waarin W_{ij} gelijk is aan de label van (i, j) als (i, j) een tak is. Als (i, j) geen tak is wordt W_{ij} gelijk gesteld aan ∞ of 0 , afhankelijk van het probleem.

- C. Een andere matrixvorm, waarmee men elke enkelvoudige graaf één-éénduidig (tot op isomorfieën na) kan representeren is de incidentiematrix.

Definitie

De incidentiematrix B van een ongerichte graaf G is een $(n \times m)$ -matrix, waarin $b_{ij} = 1$ als knoop i incident is met tak j , terwijl $b_{ij} = 0$ als dit niet het geval is.

Definitie

De incidentiematrix B van een gerichte graaf G is een $(n \times m)$ -matrix, waarin $b_{ij} = 1$ als knoop i de beginknoop is van tak j , $b_{ij} = -1$ als knoop i de eindknoop is van tak j , terwijl $b_{ij} = 0$ als knoop i niet incident is met tak j .

De rijsom $\sum_{j=1}^m b_{ij}$ is in het geval van een ongerichte graaf gelijk aan $\rho(v_i)$. Voor een gerichte graaf heeft deze som de waarde $\rho^+(v_i) - \rho^-(v_i)$.

Ga zelf na wat de waarde is van de kolomsommen.

- D. Er zijn ook representaties van grafen, waarbij niet van een matrix wordt gebruik gemaakt. Een mogelijkheid is, om een lijst van de takken van G te maken (takkenlijst) in de vorm van paren knopen. Deze representatie kan geïmplementeerd worden met twee rijen (arrays) knopen.

$$g = (g_1, g_2, \dots, g_m)$$

en $h = (h_1, h_2, \dots, h_m)$

Het element g_i is de beginknoop van tak i en h_i is de eindknoop. Duidelijk is dat zowel de grafen met als zonder parallelle takken op dezelfde wijze gerepresenteerd worden.

Om enig inzicht te krijgen in de hoeveelheid geheugenruimte die nodig is voor deze representatie, veronderstellen we dat elke knoop wordt opgeslagen in een l -bits woord. (Dit kan als $2^l \geq n$).

Het totale aantal woorden dat nodig is bedraagt dan $2m$.

Wanneer we deze representatie willen vergelijken met die van A is het van belang te weten hoe m afhangt van n . Als $m \sim cn^2$ ($c \neq 0$) dan gebruikt A , in het algemeen, asymptotisch minder geheugenruimte dan D . Als $m = o(n^2)$ is D te verkiezen wat hoeveelheid geheugenruimte betreft. In praktische gevallen zal echter ook de woordlengte l een beslissende rol kunnen spelen.

- E. Als laatste vorm van geheugenopslag noemen we de knopenlijst, ook wel nabuurstructuur genoemd. We voeren eerst in het begrip $\text{Adj}(v)$. Hiermee bedoelen we de lijst van knopen w met de eigenschap dat $(v, w) \in EG$. Als (v, w) een k -voudige tak is, komt w in totaal k keer in $\text{Adj}(v)$ voor. (De lijst $\text{Adj}(v)$ is dus een lijst van elementen van een multiverzameling).

Definitie

De knopenlijst of nabuurstructuur van een graaf G is de rij lijsten

$$v_1: \text{Adj}(v_1); v_2: \text{Adj}(v_2); \dots; v_n: \text{Adj}(v_n)$$

De hierboven gedefinieerde structuur kan geïmplementeerd worden als een array van n lineair geketende lijsten (zie [6]).

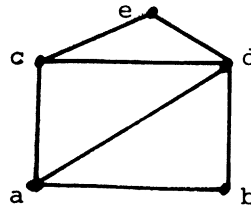
Wanneer per woord een label van een knoop wordt opgeslagen zijn er voor een gerichte graaf $n + m$ woorden nodig. Voor een ongerichte

graaf zijn er $n + 2m$ woorden nodig, omdat elke tak twee keer gerepresenteerd moet worden.

De knopenlijst is vooral handig in algoritmen, waar knopen worden toegevoegd of weggehaald.

Voorbeeld 10

Beschouw de ongerichte graaf, die bij graaf a uit voorbeeld 3 hoort.



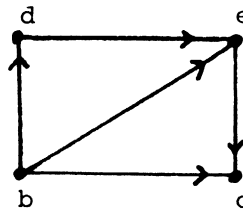
De nabuurstructuur van deze graaf luidt:

a: b,c,d; b: a,d; c: a,d,e; d: a,b,c,e; e: c,d

Voorbeeld 11

De digraaf

•
a



heeft als nabuurstructuur:

a:; b:c,d,e; c:; d:e; e:c

Let erop, dat de lijsten van a zowel als van c leeg zijn. $\square$

Geef zelf de representaties A,B,C en D voor de grafen uit voorbeeld 10 en uit voorbeeld 11.

4.2. Opspannende bomen

In deze en in de volgende paragrafen zullen we met het woord graaf een enkelvoudige graaf bedoelen, tenzij anders vermeld.

Overigens kan een graaf al of niet samenhangend en al of niet gericht zijn.

4.2.1. Definities en eigenschappen

Graaf G (gericht of ongericht) heeft n knopen en m takken.

Beschouw, in het geval dat G samenhangend is, een takdeelgraaf $T(G)$ met n knopen en $n-1$ takken. Een dergelijke deelgraaf is volgens de definitie een boom. De $n-1$ takken van zo'n deelgraaf vormen het minimale aantal, nodig om alle knopen van G met elkaar te verbinden. (Ga dit na!).

Definitie

Als G samenhangend is, heet een takdeelgraaf met n knopen en met $n-1$ takken een opspannende boom (spanning tree) van G .

Als G niet samenhangend is, heet een verzameling bomen die voor elke komponent een opspannende boom bevat, een opspannend bos (spanning forest).

Opmerkingen

1. De boom $T(G)$ is gericht of ongericht al naar gelang G dat is.
2. In plaats van het begrip opspannende boom wordt ook wel de term skelet gebruikt.

Definitie

In een (tak-) gelabelde, samenhangende graaf is een minimaalskelet een opspannende boom $T(G)$ waarvoor geldt dat het totale gewicht

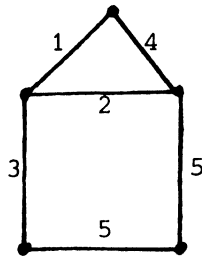
$$\sum_{e \in T(G)} l(e)$$

minimaal is.

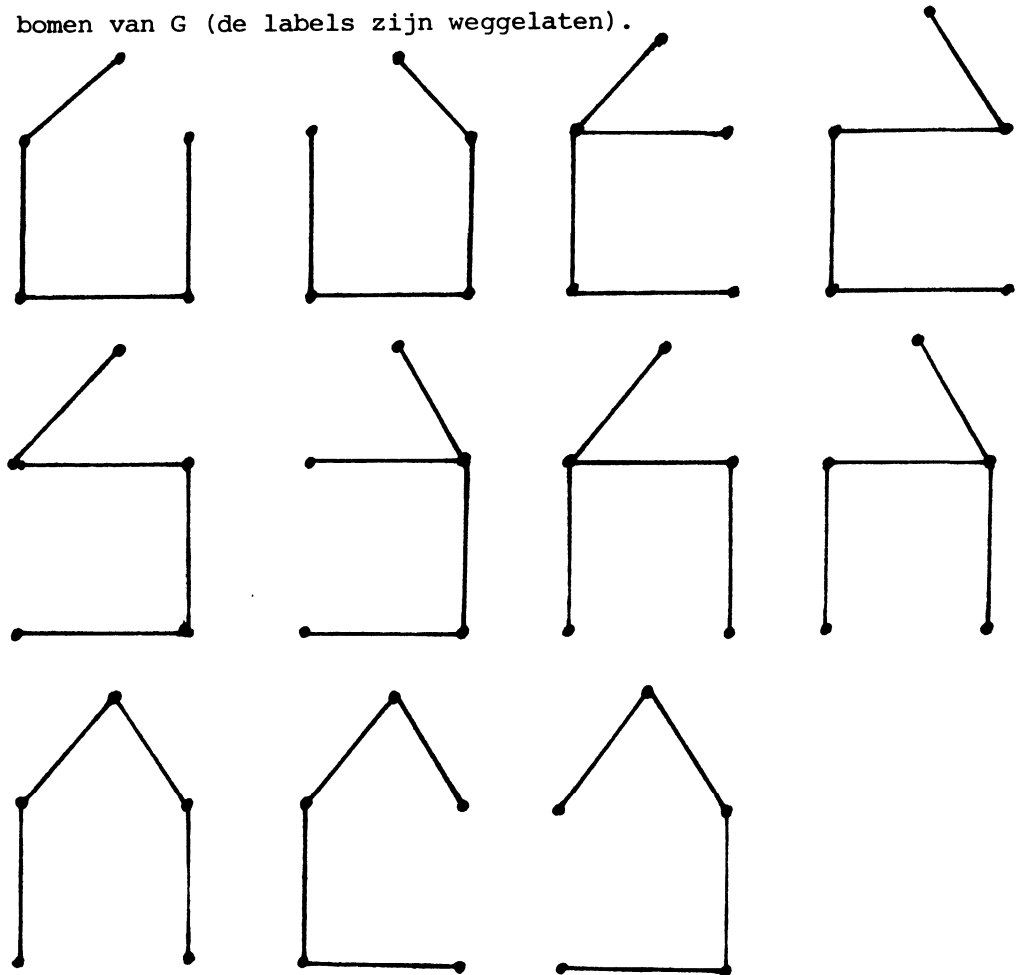
Een minimaalskelet wordt ook wel spanningsboom of minimaalkollektor genoemd. Dergelijke skeletten spelen een belangrijke rol in de theorie van elektrische netwerken en in sommige problemen die verband houden met kommunikatienetwerken.

Voorbeeld 12

Gegeven is de volgende takgelabelde graaf G .



De onderstaande serie van 11 grafen bestaat uit alle opspannende bomen van G (de labels zijn weggelaten).



De derde en de zevende graaf in deze serie zijn minimaalskeletten. □

Uit dit voorbeeld blijkt, dat een minimaalskelet niet eenduidig bepaald hoeft te zijn.

De volgende stelling verschaft enige informatie over de mogelijke samenstelling van een minimaalskelet.

Stelling (4.10)

Als G een samenhangende graaf is, waarvan de takken

gelabeld zijn met niet-negatieve getallen en als t een tak is incident met knoop v , waarvoor geldt $l(t) = \min \{ l(e) \mid e \text{ incident met } v \}$, dan is er een minimaalskelet dat de tak t bevat.

Bewijs

Stel T is een minimaalskelet en neem aan dat $t \notin ET$.

Door toevoeging van t aan ET ontstaat een circuit in de bijbehorende ongerichte graaf (stelling (4.8)). Dit circuit bevat naast t nog een tak $t' \in ET$ die incident is met v . Verwijder nu de tak t' . Er ontstaat dan weer een skelet $T' = (T - t') \cup t$. Omdat $l(t) \leq l(t')$ en omdat T minimaal was moet wel gelden $l(t) = l(t')$. Dus T' is ook een minimaalskelet en $t \in ET'$. $\square$

Opmerking

Met een lichte variant van dit bewijs kan men aantonen, dat als alle gewichten verschillend zijn er precies één minimaalskelet is. Zie hiervoor vraagstuk 25. Hetzelfde kan men aantonen m.b.v. de algoritmen in 4.2.

4.2.2. Totale aantal opspannende bomen

We zullen het totale aantal opspannende bomen van een samenhangende graaf G (ook wel de komplexiteit van G genoemd) bepalen. Ter voorbereiding leiden we de volgende, door Poincaré gevonden, stelling af.

Stelling (4.11)

Zij B de incidentiematrix van een gerichte graaf G . Elke vierkante deelmatrix van B heeft een determinantwaarde 0 of 1 of -1.

Bewijs

Laat S een vierkante deelmatrix van B zijn. Een kolom van S heeft hoogstens twee elementen ongelijk aan nul.

Als elke kolom twee elementen ongelijk aan nul heeft zijn dit steeds 1 en -1. Omdat alle kolomsommen nul zijn is S singulier en $\det S = 0$.

Als er geén kolom met elementen ongelijk nul is, geldt eveneens $\det S = 0$. Overblijft het geval, dat er een kolom is met slechts één element ongelijk aan nul. Dit element is dan 1 of -1. Ontwikkeling van $\det S$ naar de bewuste kolom levert dat $\det S = \pm \det S'$, waarin S' één rij en één kolom minder heeft dan S . Zetten we dit proces voort dan komen we terecht op een determinant met waarde 0 of op een determinant van een (1×1) -matrix met waarde 1 of -1. $\square$

Zij B weer de incidentiematrix van G . Verder is $\bar{G}$ de bij G behorende ongerichte graaf.

Met de notatie

$$B(i_1, i_2, \dots, i_p; j_1, j_2, \dots, j_p)$$

bedoelen we de determinant van de deelmatrix van B (onderdeterminant of minor), die ontstaat door het schrappen van alle rijen met index $i \notin \{i_1, i_2, \dots, i_p\}$ en van alle kolommen met index $j \notin \{j_1, j_2, \dots, j_p\}$, waarbij we nog veronderstellen $i_1 \leq i_2 \leq \dots \leq i_p$ en $j_1 \leq j_2 \leq \dots \leq j_p$. In onderstaande stelling is U een verzameling takken van G , met indices $j_1, j_2, \dots, j_{n-1}$.

Met G_U duiden we de door U bepaalde takdeelgraaf van G aan en met $\bar{G}_U$ de bijbehorende takdeelgraaf in $\bar{G}$. Verder is $i_1, i_2, \dots, i_{n-1}$ een willekeurig gekozen $(n-1)$ -tal rij-indices.

Stelling (4.12.)

Als G een samenhangende, gerichte graaf is met incidentiematrix B dan geldt dat G_U een opspannende boom van G is dan en slechts dan als

$$B(i_1, i_2, \dots, i_{n-1}; j_1, j_2, \dots, j_{n-1}) = \pm 1.$$

Bewijs

(i) We nemen aan dat bovenstaande determinant gelijk is aan 1 of aan -1 . Stel G_U is géén boom. Omdat $|EG_U| = n-1$ en $|VG_U| \leq n$ bevat $\bar{G}_U$ een circuit. De overeenkomstige takken in G vormen ook een circuit, mits we sommige takken (eventueel) van richting omdraaien. Dit omdraaien betekent in B het vermenigvuldigen van de bijbehorende kolommen met -1 . De determinant $B(i_1, i_2, \dots, i_{n-1}; j_1, j_2, \dots, j_{n-1})$ kan hierdoor van teken veranderen. Echter, de kolomvectoren die horen bij de takken van een circuit in een gerichte graaf zijn samen gelijk aan de nulvektor (Ga dit na). Hieruit volgt dat $B(i_1, i_2, \dots, i_{n-1}; j_1, j_2, \dots, j_{n-1}) = 0$, voor alle mogelijke keuzes van $i_1, i_2, \dots, i_{n-1}$. Dus hebben we een tegenspraak en G_U is een opspannende boom.

(ii) Ga er nu van uit, dat G_U een opspannende boom is. Het bewijs dat $|B| = 1$ verloopt met volledige inductie naar n .

Voor $n = 2$ is de bewering triviaal.

Neen verder aan, dat de bewering juist is voor zekere $n \geq 2$. We gaan dan bewijzen dat hij ook juist is voor een samenhangende graaf met $n+1$ knopen. De boom G_U bevat minstens twee knopen van graad 1 . De verzameling

$\{v_{i_1}, v_{i_2}, \dots, v_{i_k}\}$ bevat dus minstens één zo'n knoop, zeg v_{i_k} .

Ontwikkelen we $B(i_1, i_2, \dots, i_n; j_1, j_2, \dots, j_n)$ naar de elementen van de rij i_k (waarvan er maar één ongelijk aan nul is), dan blijkt de determinant gelijk te zijn aan ± 1 keer de waarde van een soortgelijke determinant met $n-1$ rijen en $n-1$ kolommen. De graaf die hierbij hoort is gelijk aan G , zonder v_{i_k} en zonder de ene tak incident met v_{i_k} . Deze graaf is dus weer samenhangend en daarom geldt volgens de inductie-aanname dat de bijbehorende determinant ongelijk aan nul is. Hiermee is het tweede gedeelte van de stelling bewezen. $\square$

Stelling (4.12) biedt ons een criterium waarmee we kunnen uitmaken of een takdeelgraaf een opspannende boom is.

Een laatste hulpmiddel is de stelling van Binet-Cauchy.

Stelling (4.13)

Is P een $(n \times m)$ -matrix en Q een $(m \times n)$ -matrix, met $n \leq m$, dan geldt

$$\det(PQ) = \sum_{1 \leq j_1 \leq j_2 \leq \dots \leq j_n \leq m} P(1, \dots, n; j_1, \dots, j_n) Q(j_1, \dots, j_n; 1, \dots, n).$$

Bewijs

Zie een willekeurig niet te elementair boek over lineaire algebra. $\square$

(Ga na wat er gebeurt als $n = m$).

Stelling (4.14)

Zij G een samenhangende gerichte graaf met incidentiematrix B . Zij B_0 de matrix die uit B ontstaat door het schrappen van een willekeurige rij. Het totale aantal opspannende bomen van G wordt gegeven door

$$\det(B_0 B_0^t).$$

Bewijs

Uit $|VG| = n$ en $|EG| = m$ volgt dat B_0 een $((n-1) \times m)$ -matrix is. Uit de stelling van Binet-Cauchy volgt

$$\begin{aligned} \det(B_0 B_0^t) &= \sum_{1 \leq j_1 \leq \dots \leq j_{n-1} \leq m} B_0(1, \dots, n-1; j_1, \dots, j_{n-1}) B_0^t(j_1, \dots, j_{n-1}; 1, \dots, n-1) = \\ &= \sum_{1 \leq j_1 \leq \dots \leq j_{n-1} \leq m} \{B_0(1, \dots, n-1; j_1, \dots, j_{n-1})\}^2. \end{aligned}$$

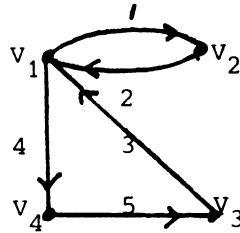
Nu geeft een term in het rechter lid een bijdrage 1 als de bijbehorende graaf een opspannende boom is en 0 als dit niet zo is.

Hieruit volgt de stelling. $\square$

Let erop dat stelling (4.12) hier toegepast mocht worden, omdat G gericht is.

Als G ongericht is kennen we eerst (op willekeurige wijze) een richting toe aan elke tak. De gerichte skeletten die men dan krijgt van de aldus verkregen gerichte graaf horen één-éénduidig bij de ongerichte skeletten van de ongerichte graaf. Hun aantallen zijn dus ook gelijk en worden gegeven door $\det (B_O B_O^t)$ waarbij B de incidentiematrix van de gerichte graaf is en niet van G zelf!

Voorbeeld 13



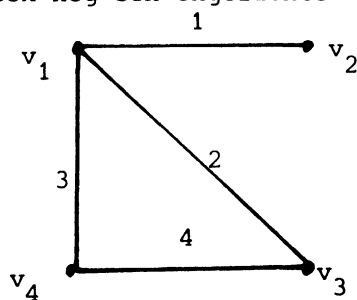
Door schrappen van de eerste rij in de incidentiematrix krijgen we

$$B_O = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix},$$

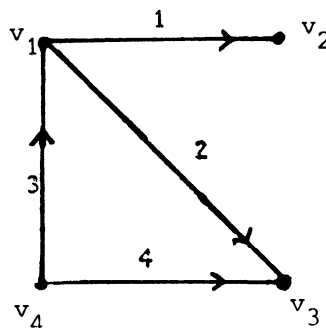
$$B_O B_O^t = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}.$$

Het aantal mogelijke opspannende bomen is dus gelijk aan $\det (B_O B_O^t) = 6$.

We beschouwen ook nog een ongerichte graaf.



Allereerst maken we hiervan de volgende gerichte graaf.



Schrappen van de eerste rij in de incidentiematrix levert

$$B_0 = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \quad B_0 B_0^t = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}.$$

Het aantal opspannende bomen is nu $\det(B_0 B_0^t) = 3$.

Ga voor beide grafen na hoe de opspannende bomen eruit zien.

Voor het tellen van het aantal skeletten van een ongerichte graaf G is het niet echt nodig om van G een gerichte graaf te maken. Er geldt namelijk (zie vraagstuk 12 van hoofdstuk 4):

$$(B_0 B_0^t)_{ij} = \begin{cases} -(\text{aantal takken tussen } v_i \text{ en } v_j), & i \neq j, \\ \rho(v_i) & , i = j, \end{cases}$$

waarin $\rho(v_i)$ de valentie van v_i in G is. (Voor gerichte grafen geldt dit overigens ook, mits we stellen $\rho(v_i) = \rho^+(v_i) + \rho^-(v_i)$).

Ga met behulp hiervan na dat door het schrappen van de 4^{de} rij van B , behorend bij de eerste graaf in voorbeeld 13, het aantal skeletten gegeven wordt door

$$\begin{vmatrix} 4 & -2 & -1 \\ -2 & 2 & 0 \\ -1 & 0 & 2 \end{vmatrix}.$$

Als bijzonder geval van het resultaat van vraagstuk 12 formuleren we nog:

Stelling (4.14a)

Zij G een samenhangende, enkelvoudige, ongerichte graaf. Laat A de nabuurnatrix zijn en Δ de diagonaalmatrix met $\Delta_{ii} = \rho(v_i)$ en $\Delta_{ij} = 0$ als $i \neq j$. Het aantal skeletten van G is dan gelijk aan D_i , waarbij D_i de minor is van $\Delta - A$, die ontstaat door schrappen van i ^{de} rij en kolom ($1 \leq i \leq n$).

Stelling (4.14) (en (4.14a)) is voor het eerst afgeleid door Kirchhoff en speelt een rol in de theorie van elektrische netwerken.

Voorbeeld 14

Een speciaal geval van de stelling van Kirchhoff verkrijgt men door toepassing op de volledige graaf K_n . Wij krijgen dan de $(n-1) \times (n-1)$ -matrix

$$B_0 B_0^t = \begin{pmatrix} n-1 & -1 & \cdots & \cdots & \cdots & -1 \\ -1 & n-1 & \cdots & \cdots & \cdots & -1 \\ \vdots & \vdots & & & & \vdots \\ \vdots & \vdots & & & & \vdots \\ \vdots & \vdots & & & & \vdots \\ -1 & -1 & \cdots & \cdots & \cdots & n-1 \end{pmatrix}.$$

Om $\det(B_0 B_0^t)$ te berekenen tellen we eerst alle rijen op bij de eerste rij en vervolgens tellen we deze "nieuwe" rij op bij alle overige rijen. Ga hiermee na dat het aantal skeletten van K_n gelijk is aan n^{n-2} .

Dit resultaat staat bekend als de stelling van Cayley.

Men kan dit antwoord ook interpreteren als het totale aantal bomen op n knopen!

4.2.3. Algoritmen voor minimaalskelet

In een gelabelde graaf G is het vaak belangrijk een minimaalskelet te konstrueren.

Denk bijvoorbeeld aan de situatie, dat de graaf een transportnetwerk voorstelt, waarbij de gewichten (labels) van de takken transportkosten zijn.

Een mogelijke konstruktie is de "greedy" algoritme (Kruskal):

Sorteer eerst de takken van G naar toenemende grootte van hun gewichten.

Neem de lichtste tak en voeg elke keer de lichtste, nog niet gebruikte tak toe, die niet een circuit vormt met de reeds gekozen takken. Deze tak hoeft niet aan de partiële boom vast te zitten.

Dat de zo gekonstrueerde boom T tenslotte een minimaalskelet vormt is als volgt te bewijzen.

Als de takken van T zijn $e_1, e_2, \dots, e_{n-1}$, dan nemen we hiervoor aan dat geldt $l(e_1) \leq l(e_2) \leq \dots \leq l(e_{n-1})$.

Stel $T_{\min}$ is een minimaalskelet dat $e_1, e_2, \dots, e_{i-1}$ bevat, maar niet e_i ($i < n$). Voegen we e_i toe aan $T_{\min}$, dan ontstaat er een circuit, dat zeker een tak $x \in E T_{\min}$ bevat, die ongelijk is aan $e_1, e_2, \dots$.

$\dots, e_i$. De takken $x, e_1, e_2, \dots, e_{i-1}$ zitten alle in $ET_{\min}$ en vormen dus geen circuit. Er moet dus wel gelden $l(x) \geq l(e_i)$ op grond van de konstruktie van T. Analoog aan het bewijs van stelling (4.10) toont men nu aan dat niet kan gelden $l(x) > l(e_i)$ en dat dus $T'_{\min} = (T_{\min} - x) \cup e_i$ ook een minimaalskelet is en dus $l(x) = l(e_i)$. Zo voortgaande verkrijgen we uiteindelijk een minimaalskelet, waarin $e_1, e_2, \dots, e_{n-1}$ zitten. Dit valt dan samen met T.

Opmerking

Elke keer als er twee of meer geschikte takken zijn met minimaal gewicht moet er een keuze gedaan worden. Dit maakt dat een minimaalskelet in het algemeen niet éénduidig bepaald is (zie ook stelling (4.10)).

Bewezen kan worden dat het sorteren van de takken naar gewicht bepalend is voor de complexiteit van de algoritme. De snelste sorteeralgoritme is namelijk asymptotisch van de orde $O(m \log m)$ terwijl de rest van de algoritme kan uitgevoerd worden in $o(m \log m)$ bewerkingen.

Een andere methode, om een minimaalskelet te bepalen is de naaste-buur-algoritme (nearest neighbour algorithm, Prim).

Begin met een knoop v_1 . Zoek de lichtste tak, die incident is met v_1 . Stel dat dit (v_1, v_2) is.

We hebben nu een partiële boom met de knopen v_1 en v_2 en de tak (v_1, v_2) . Zoek vervolgens elke keer de lichtste tak, die incident is met de knopen van de partiële boom en die een nieuwe knoop toevoegt. Het bewijs dat deze methode na $n-1$ stappen een minimaalskelet oplevert is weer analoog aan het bewijs van de vorige algoritme en aan dat van stelling (4.10). We laten dit bewijs hier achterwege.

Pas nu beide algoritmen toe op de gerichte graaf uit voorbeeld 12 om een minimaalskelet te konstrueren. Vat de labels 1,2,3,4,5 hierbij

als gewichten op.

Probeer ook een implementatie van de naaste-buur-algoritme te vinden, waarbij gebruik mag worden gemaakt van opdrachten waarmee een element wordt toegevoegd aan een verzameling of eruit wordt weggenomen. Tevens mag men beschikken over de arrays $naast(v)$ en $afst(v)$, die respectievelijk de naaste buur van v en de afstand daarvan tot v bevatten, voor alle $v \in VG$ ($naast$ en afstand hier met betrekking tot de gewichten van de takken die we als afstand interpreteren).

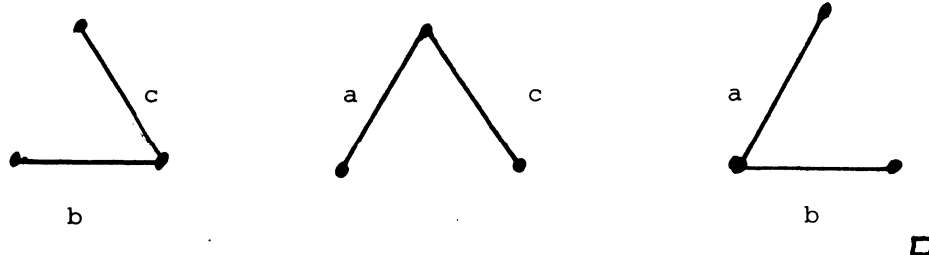
4.2.4. Algoritme voor alle opspannende bomen

In de theorie van elektrische netwerken komt het probleem naar voren, om alle opspannende bomen ofwel alle skeletten van een samenhangende graaf te bepalen. Omdat het totale aantal van zulke bomen reeds voor kleine grafen zeer groot kan zijn (vergelijk het resultaat voor volledige grafen uit 4.2.2) is het probleem van de efficiency van veel belang.

Een bekende algoritme die alle skeletten genereert, is gebaseerd op de volgende gedachte. Beschouw een tak (v,w) . De verzameling opspannende bomen kan gesplitst worden in twee klassen. De bomen uit de ene klasse bevatten (v,w) wel en die uit de andere doen dat niet. Het eerste type opspannende boom bestaat uit (v,w) en een opspannende boom van de graaf $G_{(v,w)}$, die uit G ontstaat door de knopen v en w te identificeren en de tak (v,w) weg te laten. Dit heet wel de kontraktie van G naar (v,w) . Let erop, dat $G_{(v,w)}$ meervoudige verbindingen kan bevatten. (Bij een andere definitie van kontraktie wordt een meervoudige verbinding vervangen door een enkelvoudige).

De bomen uit de tweede klasse zijn opspannende bomen van de graaf $G^{(v,w)} = G - (v,w)$.

We schrijven $G = G^{(v,w)} + G_{(v,w)}$.



De observatie, dat de skeletten van G ofwel in verband gebracht kunnen worden met die van G^e ofwel met die van G_e (e is een of andere tak) is essentieel voor de konstruktie en kan dienst doen om een rekursieve versie van de algoritme te ontwerpen. Immers, zowel G^e en G_e hebben beide minder takken dan G , waardoor het probleem verschoven is naar twee gelijksoortige problemen met kleinere parameterwaarden.

We geven nu de algoritme weer in de vorm van een rekursieve procedure, genaamd GROEI. Deze naam slaat erop, dat een opspannende boom als het ware groeit, door hem uit kleinere bomen te laten ontstaan (Mcilroy, Read, Tarjan). Omdat er rekursie is met betrekking tot EG, zal de variabele E in de procedure op elk niveau betekenen: de verzameling takken, die nog aan de reeds gekonstrueerde partiële boom T kunnen worden toegevoegd. De variabele B is een hulpvariabele en is lokaal voor de procedure evenals de variabele e die de tak voorstelt, waarnaar op elk niveau gekontraheerd wordt.

Opmerking

We veronderstellen weer dat set-operaties als de vereniging en de doorsnede beschikbaar zijn, alsmede de mogelijkheid om de voorwaarde $E \subseteq T$ te testen. Op implementaties ervan gaan we niet in. Vergelijk ook de opmerking aan het einde van 4.2.3.

GROEI		
T	$E \subseteq T$	F
$ T = n-1$	e := element van $E - T$	
T	T := $T \cup \{e\}$	
druk af (T)	B := $(E-T) \cap \{\text{takken die knopen in zelfde komponent van T verbinden}\}$	
	E := $E - B$	
	GROEI	
	E := $E \cup B$	
	T := $T - \{e\}$	
	E := $E - \{e\}$	
	GROEI	
	E := $E \cup \{e\}$	

Algoritme 4.1

De opdrachten voorafgaande aan de eerste aanroep van GROEI simuleren de kontraktie naar e. De opdrachten daarna tot de tweede aanroep van GROEI, simuleren het weglaten van e. De opdracht $E := E \cup \{e\}$ heeft als doel de verzameling E te herstellen, in verband met de betekenis van E.

Aanroep van de procedure met $E = EG$ en $T = \emptyset$ geeft aanleiding tot het afdrukken van alle opspannende bomen van G. Immers alle takken $e \in EG$ komen (rekursief) aan de beurt en steeds worden alle bomen gekonstrueerd die e wél en die e niet bevatten. (Ga de werking na op de graaf uit voorbeeld 16.)

De kontraktietechniek kan ook gebruikt worden om het totale aantal bomen van een graaf te bepalen zonder die bomen overigens te konstrueren. Hiervan geven we een eenvoudig voorbeeld.

Voorbeeld 17

We kontraheren de graaf uit voorbeeld 12 naar tak 2 en krijgen dan



De opspannende bomen die tak 2 niet bevatten zijn de opspannende bomen van de linker graaf. De opspannende bomen die tak 2 wél bevatten zijn de opspannende bomen van de rechter graaf.

Evident is dat de linker graaf 5 opspannende bomen heeft en de rechter graaf 6.

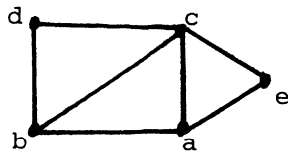
Het totale aantal is dus 11.

□

4.2.5. Depth-first algoritme voor een opspannende boom

Een van de meest gebruikte technieken, om in algoritmen alle knopen (en alle takken) van een graaf af te lopen en te onderzoeken op zekere eigenschappen berust op een zogenaamd depth-first onderzoek.

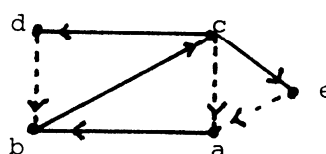
We illustreren deze techniek met de volgende ongerichte graaf G , die we door middel van zijn nabuurstructuur representeren.



a: b,c,e ;
 b: a,c,d ;
 c: a,b,d,e ;
 d: b,c ;
 e: a,c

Figuur 4.4

We beginnen de konstruktie in een willekeurige knoop, zeg a. Vanuit a doorlopen we de eerste tak, die we tegenkomen, namelijk (a,b). Omdat we in b nog niet eerder zijn geweest, blijven we daar en zoeken de eerste tak op, die nog niet eerder is doorlopen. Dat is tak (b,c). Vanuit c is (c,a) de eerste nog niet doorlopen tak. We doorlopen deze tak en komen dan in a, die al eerder bezocht is. Daarom keren we terug naar c en markeren (c,a), om deze tak te onderscheiden van takken die naar nieuwe knopen leiden. De volgende niet gebruikte tak is (c,d). In d blijven we weer, want d is een nieuwe knoop. De eerste niet gebruikte tak vanuit d is (d,b), die we weer markeren, daar b een reeds bezochte knoop is. Omdat er geen andere ongebruikte takken uit d vertrekken keren we terug naar c en nemen de volgende niet gebruikte tak, namelijk (c,e), etc. Elke keer, wanneer we geen ongebruikte takken meer kunnen vinden, gaan we terug naar de vorige knoop. Zo eindigen we tenslotte in a, en hebben dan alle takken gebruikt. Ga na dat op deze wijze de volgende gerichte graaf G' ontstaat.



Figuur 4.5

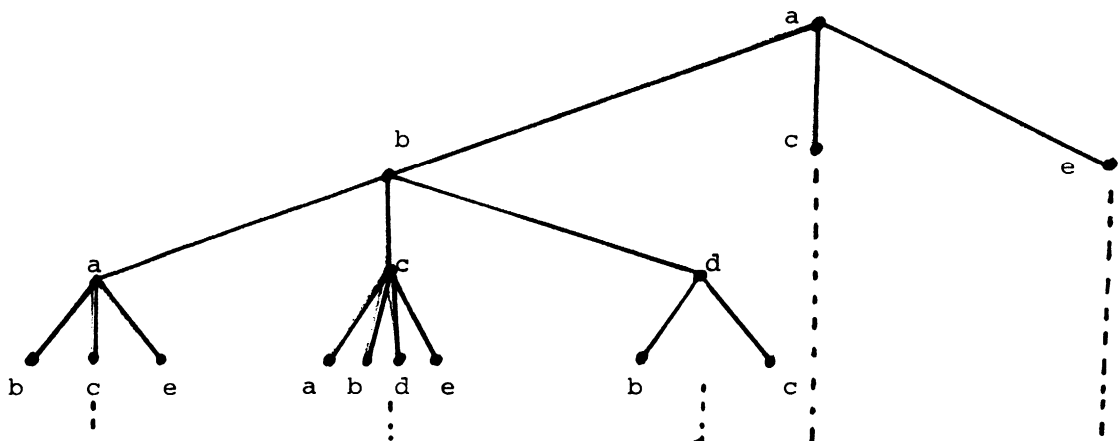
De niet-gestippelde takken vormen een gerichte opspannende boom van G' . Knoop a heet de wortel van deze boom. Vanzelfsprekend is de bijbehorende ongerichte boom een opspannende boom van de ongerichte graaf G .

Elke knoop in G' is vanuit de wortel via een gericht pad bereikbaar. In dit verband zeggen we dat x een voorouder (ancestor) is van y als er een gericht pad is van x naar y , terwijl y een afstammeling (descendant) is van x .

De gestippelde takken, waarlangs we terugkeren als we in een reeds eerder bezochte knoop terechtkomen, zullen we de terugkeertakken (back edges) noemen.

Opmerkingen

1. Het begrip depth-first wordt vrijwel altijd in verband gebracht met het zoeken in een boomstructuur. We kunnen de boven besproken zoekmethode ook als zodanig opvatten. Daartoe geven we de nabuurstructuur weer in de vorm van een boom.



Figuur 4.6

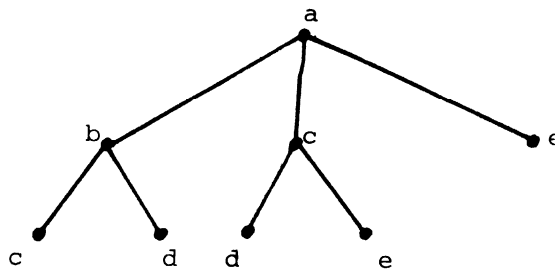
Ga na, dat de algoritme erop neerkomt, dat we ons uitgaande van a , eerst zo ver mogelijk vertikaal in de boom bewegen, totdat één of andere voorwaarde niet meer geldt (het bereiken van een nieuwe knoop). Daarna bewegen we ons horizontaal naar de volgende potentiële oplossing (de volgende ongebruikte tak).

2. We merken ook op, dat de algoritme een voorbeeld is van backtracking (zie ook [5]).

Op elk niveau wordt in de boom van figuur 4.6 beslist of de partiële oplossing (partiële opspannende boom) die tot dan toe gekonstrueerd is tot een volledige oplossing (opspannende boom) kan leiden. Is dat niet het geval dan vervalt een tak met alles wat er aanhangt.

3. De boom uit figuur 4.6 is in principe onbepaald in grootte.

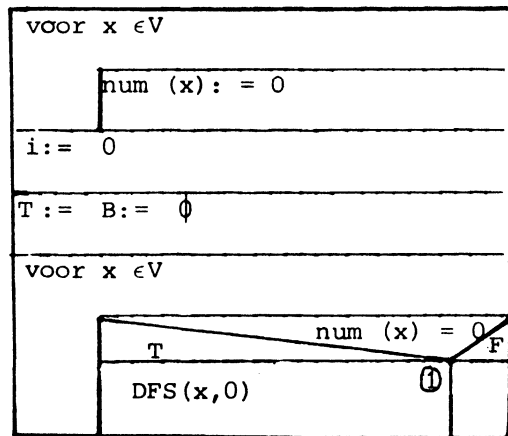
Aan elke knoop kunnen we opnieuw takken hangen die de verbindingen met zijn burens aangeven. Wanneer we de eis opleggen, dat elke tak precies één keer moet voorkomen, zouden we tot een boomstructuur komen als



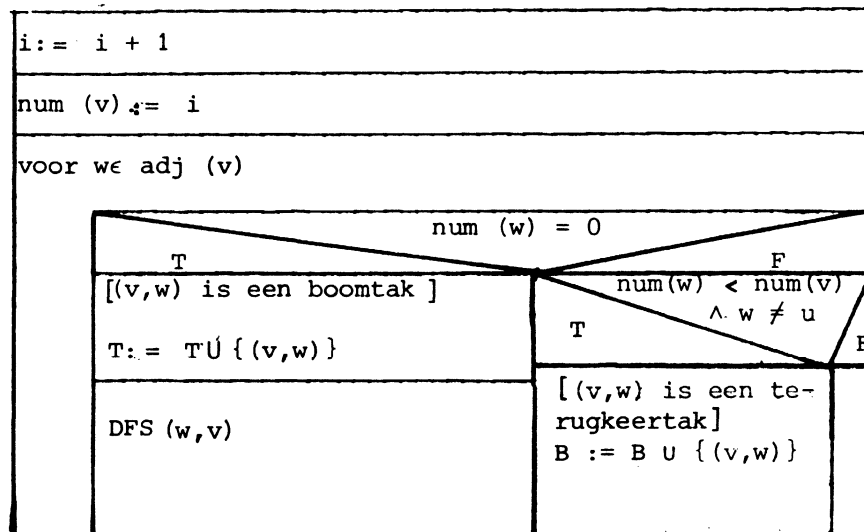
Figuur 4.7

Ga na wat er gebeurt als we nu de depth-first methode toepassen en geef een verklaring.

Om de depth-first methode in een algoritme te implementeren, moeten we onderscheid kunnen maken tussen knopen die wél en die niet bezocht zijn. We doen dit door ze te nummeren met de getallen $1, 2, \dots, n$ in de volgorde, waarin we ze bezoeken. Is een knoop nog niet bezocht dan heeft hij nummer 0. Algoritme 4.2 (Tarjan) is een depth-first algoritme. De rekursieve procedure $\text{DFS}(v, u)$ verricht een depth-first onderzoek op de graaf $G = (V, E)$ met $v \in V$ en konstrueert een gerichte opspannende boom T , zodanig dat u de onmiddellijke voorouder is van v in T (ofwel de vader).



① DFS (v, u)



Algoritme 4,2

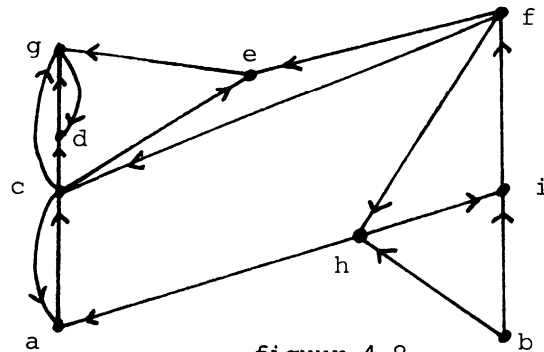
De herhalingsopdracht voor $x \in V$ doe opdracht betekent, dat de opdracht achter doe wordt uitgevoerd voor alle elementen van V in één of andere volgorde, afhankelijk van de representatie van V en de implementatie ervan. Verder is T steeds de tot dan toe gekonstrueerde boom en B is de verzameling van terugkeertakken. Als G niet samenhangend is zal algoritme 4,2 achtereenvolgens voor elke komponent een opspannende boom konstrueren en voor G dus een opspannend bos.

De procedure DFS wordt voor elke nieuwe knoop een keer aangeroepen, dus in totaal n keer. Het aantal stappen per aanroep is evenredig met het aantal takken, dat incident is met die knoop. De kompleksiteit

van algoritme 4.2 voor ongerichte grafen is dus $O(n+m)$. Voor samenhangende grafen is dit gelijk aan $O(m)$.

Opmerkingen

1. In 4.2.3 bespraken we een tweetal algoritmen (de greedy algoritme en de naaste-buur-algoritme) om een speciaal type opspannende boom te konstrueren, namelijk een minimaalskelet. In 4.2.4 gaven we een algoritme, waarmee men alle opspannende bomen kan konstrueren. In 4.2.5 tenslotte werd in het kader van de depth-first techniek een algoritme aangegeven voor de konstruktie van een opspannende boom, uitgaande van een gegeven nabuurstructuur. Als een soort tegenhanger hiervan zullen we in 4.5 nog een ander algoritme afleiden voor het bepalen van een opspannende boom en wel als illustratie van een breadth-first onderzoek.
- (*) 2. Past men algoritme 4.2 toe op gerichte grafen, dan kunnen er komplikaties optreden. Als voorbeeld nemen we de graaf uit figuur 4.8.

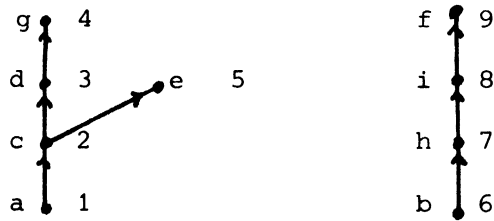


figuur 4.8

Uitgaande van de nabuurstructuur

a: c; b: h,i; c: a,d,e,g; d: g; e: g; f: c,e,h; g: d; h: a,i; i: f

en beginnende met knoop a in de algoritme, vinden we de volgende gerichte graaf. Gemakshalve duiden we deze graaf aan met "opspannend bos" van de graaf uit figuur 4.8, hoewel dit niet in overeenstemming is met de in 4.2 gegeven definitie van dit begrip.



Figuur 4.9

De nummers bij de knopen geven de waarde aan van $\text{num}(v)$ en duiden dus de volgorde aan, waarin de knopen zijn bezocht.

Het zal duidelijk zijn, dat het opspannende bos in hoge mate afhankelijk is van de volgorde der knopen zoals die in de nabuurstructuur is vastgelegd. Bovendien is de keuze van de beginknoop van invloed. Waren we begonnen met knoop b dan zouden we zelfs één opspannende boom gevonden hebben.

Vergeleken met de toepassing op ongerichtte grafen zijn nu niet alle weggelaten takken in figuur 4.9 terugkeertakken. Zo is bijvoorbeeld (c, g) een tak die van c naar g loopt, terwijl $\text{num}(g) > \text{num}(c)$ is. Een dergelijke tak wordt forward edge genoemd. Verder kan een tak (v, w) met $\text{num}(w) < \text{num}(v)$ of een echte back edge zijn, als w een voorouder van v is, of een zogenaamde cross edge als w niet een voorouder van v is. Een voorbeeld van een cross edge zou zijn tak (e, g) in figuur 4.8.

Met behulp van deze nieuwe klassen van takken is het mogelijk, om algoritme 4.2. uit te breiden tot een algoritme, waarmee men de sterk samenhangende componenten van een digraaf kan bepalen. We werken dat hier niet verder uit, maar verwijzen daarvoor naar [1, Section 8.2].

(†) 4.3. Transitieve afsluiting van een graaf

Stel G is een digraaf, die niet enkelvoudig hoeft te zijn, terwijl $A \subset VG$ en $A \neq \emptyset$. We voeren in de verzameling

$$\Gamma(A) := \{v \mid v \in VG, \exists u \in A \text{ met } (u,v) \in EG\} \quad (4.1)$$

en voorts definiëren we rekursief

$$\Gamma^{(k)}(A) = \Gamma(\Gamma^{(k-1)}(A)) \quad , \quad k \geq 2, \quad (4.2)$$

$$\Gamma^{(1)}(A) = \Gamma(A). \quad (4.3)$$

We kunnen $\Gamma^{(k)}(A)$ interpreteren als de verzameling knopen van G , die via een gericht pad ter lengte k bereikbaar zijn vanuit A .

Als $A = \{v\}$ schrijven we $\Gamma^{(k)}(v)$ in plaats van $\Gamma^{(k)}(\{v\})$.

Definitie

De transitieve afsluiting (transitive closure) van A is de verzameling

$$\Gamma^+(A) := \bigcup_{k=1}^{\infty} \Gamma^{(k)}(A).$$

De verzameling $\Gamma^+(A)$ bestaat dus uit die knopen, die vanuit A via een gericht pad bereikbaar zijn.

Analoog definiëren we

$$\Gamma^{(-1)}(A) := \{u \mid u \in VG, \exists v \in A \text{ met } (u,v) \in EG\}, \quad (4.4)$$

$$\Gamma^{(-k)}(A) = \Gamma^{(-1)}(\Gamma^{(-k+1)}(A)), \quad k > 1. \quad (4.5)$$

Definitie

De inverse transitieve afsluiting van A is de verzameling

$$\Gamma^-(A) := \bigcup_{k=1}^{\infty} \Gamma^{(-k)}(A).$$

De verzameling $\Gamma^-(A)$ bestaat uit die knopen van waaruit men A kan bereiken via een gericht pad.

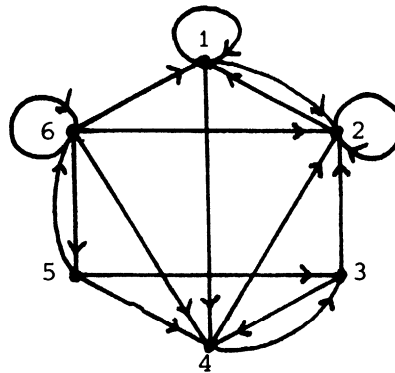
Opmerkingen

1. In de definities van $\Gamma^+(A)$ en $\Gamma^-(A)$ worden formeel een oneindig aantal verzamelingen verenigd. In de praktijk (van eindige grafen) is het echter voldoende om $\bigcup_{k=1}^n$ te beschouwen. Verklaar dit.

2. Soms wordt gedefinieerd $\Gamma^+(A) = \bigcup_{k=0}^{\infty} \Gamma^{(k)}(A)$, waarin $\Gamma^{(0)}(A) = A$.

Hierbij mogen de gerichte paden vanuit A dus ook lengte 0 hebben.

Voorbeeld 18



Men vindt eenvoudig

$$\begin{aligned}\Gamma^{(1)}(2) &= \{1, 2\} \\ \Gamma^{(2)}(2) &= \{1, 2, 4\} \\ \Gamma^{(3)}(2) &= \{1, 2, 3, 4\} \\ \Gamma^{(4)}(2) &= \{1, 2, 3, 4\} \\ \Gamma^{(-1)}(2) &= \{1, 2, 3, 4, 6\} \\ \Gamma^{(-2)}(2) &= \{1, 2, 3, 4, 5, 6\}.\end{aligned}$$

Uit de definities volgt dan

$$\Gamma^+(2) = \{1, 2, 3, 4\}$$

en

$$\Gamma^-(2) = \{1, 2, 3, 4, 5, 6\}.$$

Let erop dat niet altijd hoeft te gelden $\Gamma^{(k)}(A) \supset \Gamma^{(k-1)}(A)$.

Zo is $\Gamma^{(1)}(3) = \{2, 4\}$, terwijl $\Gamma^{(2)}(3) = \{1, 2, 3\}$. □

In het nu volgende veronderstellen we, dat G geen meervoudige takken bezit, maar eventueel wel lussen. Verder kan G gericht of onge-richt zijn. We zullen ons nu bezig houden met het bepalen van de transitieve afsluiting van de graaf G .

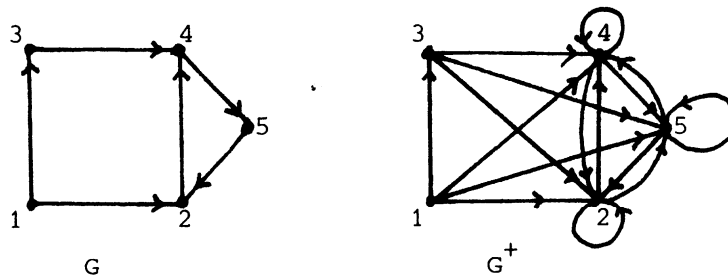
Definitie

De transitieve afsluiting G^+ van G wordt gedefinieerd door

$$\begin{aligned}VG^+ &:= VG, \\ EG^+ &:= \{(v, w) \mid v \in VG, w \in \Gamma^+(v)\}.\end{aligned}$$

De eindpunten van de takken van EG^+ vormen dus de verzameling $\Gamma^+(G)$.

Voorbeeld 19



□

We kunnen G^+ karakteriseren met een $(n \times n)$ -matrix A^+ (de verbindingsmatrix of connectivity matrix) met $A_{ij}^+ = 1$ als er een pad is van i naar j met lengte > 0 en $A_{ij}^+ = 0$ als er niet zo'n pad is. Om A^+ te berekenen zullen we verband gaan leggen tussen A^+ en de gewone nabuurmatrijs A van G .

In stelling (4.9) zagen we, dat het (i,j) -element van de matrix A^k gelijk is aan het aantal paden van lengte k tussen knoop i en knoop j . We voeren naast A^k in een matrix $B^{(k)}$ ($k \geq 0$), die wordt gedefinieerd als

$$B_{ij}^{(k)} = \begin{cases} 1, & \text{als } (A^k)_{ij} > 0, \\ 0, & \text{als } (A^k)_{ij} = 0. \end{cases} \quad (4.6)$$

Men kan $B^{(k)}$ opvatten als de nabuurmatrijs van de graaf $G^{(k)}$, die uit G ontstaat door te definiëren

$$VG^{(k)} := VG$$

en

$$EG^{(k)} := \{ (v,w) \mid v,w \in VG \wedge \exists \text{ een pad van lengte } k \text{ van } v \text{ naar } w \}.$$

Opmerking

Als $B_{ij}^{(k)} = 1$ dan is de afstand van i naar j kleiner dan of gelijk aan k .

Gemakkelijk valt nu in te zien, dat

$$A^+_{ij} = B_{ij}^{(1)} \vee B_{ij}^{(2)} \vee \dots \vee B_{ij}^{(n)}. \quad (4.7)$$

In (4.7) is gebruik gemaakt van de impliciete - of - operatie die op de gebruikelijke manier werkt op nullen en enen.

De matrices $B^{(k)}$ zijn rekursief te berekenen met behulp van de relatie

$$B_{ij}^{(k)} = (B_{i1}^{(k-1)} \wedge A_{1j}) \vee (B_{i2}^{(k-1)} \wedge A_{2j}) \vee \dots \vee (B_{in}^{(k-1)} \wedge A_{nj}) \quad (4.8)$$

met beginvoorwaarde $B_{ij}^{(0)} = \delta_{ij}$.

Nemen we de $\vee$ - en de $\wedge$ - operatie als basisbewerkingen, dan kost het berekenen van een matrixelement A_{ij}^+ met behulp van (4.7) en (4.8) $n(2n-1) + n-1 = 2n^2 - 1$ basisbewerkingen. Het totale aantal basisbewerkingen nodig om A^+ te berekenen is dus $n^2(2n^2-1) = O(n^4)$.

Er bestaat een snellere manier om A^+ te bepalen. Daartoe definiëren we nog een matrix $C^{(k)}$ ($k \geq 0$) volgens

$$C_{ij}^{(k)} = C_{ij}^{(k-1)} \vee (C_{ik}^{(k-1)} \wedge C_{kj}^{(k-1)}), \quad k > 0, \quad (4.9)$$

$$C_{ij}^{(0)} = A_{ij}. \quad (4.10)$$

De matrix $C^{(k)}$ heeft de volgende betekenis.

Het element $C_{ij}^{(k)} = 1$ dan en slechts dan als er een pad is van i naar j , met de eigenschap dat de knopen incident met dit pad, tussen i en j , alle liggen in de verzameling $\{1, 2, \dots, k\}$. Het bewijs is gemakkelijk te geven met volledige inductie.

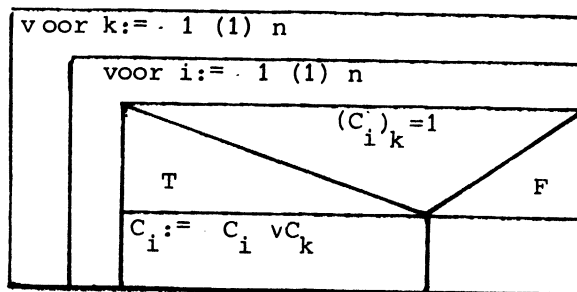
Ook zal duidelijk zijn, dat $A^+ = C^{(n)}$.

Geven we de i^{de} rij van $C^{(k)}$ aan met $C_i^{(k)}$ dan volgt uit (4.9)

$$C_i^{(k)} = \begin{cases} C_i^{(k-1)}, & \text{als } C_{ik}^{(k-1)} = 0, \\ C_i^{(k-1)} \vee C_k^{(k-1)}, & \text{als } C_{ik}^{(k-1)} = 1. \end{cases} \quad (4.11)$$

Met (4.11) is na te gaan, dat $C_k^{(k)} = C_k^{(k-1)}$, of met andere woorden de k^{de} rij verandert niet bij overgang van $C^{(k-1)}$ naar $C^{(k)}$.

Omdat verder de i^{de} rij van $C^{(k-1)}$, in het geval dat $i \neq k$ is, alleen maar gebruikt wordt om de i^{de} rij van $C^{(k)}$ te berekenen, kunnen we onderstaande algoritme opschrijven (Kleene, Warshall).



Algoritme 4.3

De transformatie van de i^{de} rij bij overgang van $c^{(k-1)}$ naar $c^{(k)}$ is hier als één (rij-)operatie geschreven, maar houdt natuurlijk de verandering van n elementen in. De complexiteit van algoritme 4.3 is dus van de orde $O(n^3)$. Door de algoritme op geschikte wijze te implementeren kan men zelfs de rekentijd nog verder terugdringen. De meeste machine- en assemblercodes bezitten namelijk instructies, waarmee de $\wedge$ - en de v -operatie per woord kan worden uitgevoerd. Als een rij van $c^{(k)}$, met n bits, in één of in enkele woorden kan worden opgeslagen betekent dit een aanzienlijke besparing in rekentijd. Vatten we deze rij-operaties op als basisoperaties, vergelijkbaar met de logische operaties uitgevoerd op twee variabelen of op twee bits, dan is de complexiteit van algoritme 4.3 van de orde $O(n^2)$.

4.4. Afstanden in grafen

In 4.1.2 definieerden we de afstand van knoop a naar knoop b als de lengte van het kortste pad van a naar b . Onder lengte van een pad verstonen we het aantal takken, waaruit dat pad bestaat. We kunnen deze begrippen wat generaliseren, door voor takgelabelde grafen (met labels uit $\mathbb{R}$ als lengte van een pad te definiëren de som van de gewichten van de takken waaruit het pad bestaat. (Een tak (i,j) zal de label of het gewicht w_{ij} hebben). Zijn de takken niet gelabeld, dan nemen we als gewichten steeds 1, waarmee we de in 4.1.2 ingevoerde lengte weer terug hebben. Als steeds definiëren we $w_{ij} = \infty$ als $(i,j) \notin EG$. We zullen ook nu weer enkelvoudige grafen beschouwen. Is een graaf G niet enkelvoudig, dan zullen de algoritmen toegepast kunnen worden op de graaf, die uit G ontstaat door alle lussen weg te laten en door elk stel meervoudige takken te vervangen door de tak met het kleinste gewicht. Verder nemen we steeds aan, dat de grafen gericht zijn. Willen we de algoritmen laten werken op ongerichte grafen, dan moet elke tak $\{i,j\}$ vervangen worden door het paar takken (i,j) en (j,i) met hetzelfde gewicht $w_{ij} = w_{ji}$.

4.4.1. Kortste pad tussen twee knopen

Het bepalen van het kortste pad tussen twee knopen kan gebeuren met een methode verwant aan de naaste-buur-algoritme uit 4.2. De beginknoop noemen we s en de eindknoop f . De methode bestaat eruit, dat we elke knoop van de graaf een tijdelijke label geven, die aangeeft hoe lang het kortste pad is van s naar die knoop, tot op dat ogenblik gevonden. Zodra er een korter pad blijkt te bestaan, wordt de label van de betrokken knoop verlaagd. Als in een bepaald

stadium van de algoritme blijkt, dat de label van één of andere knoop nooit verder verlaagd zal kunnen worden, markeren we die label als definitief. Zodra knoop f zijn definitieve label heeft gekregen kunnen we stoppen.

De zaken die essentieel zijn voor deze algoritme zijn natuurlijk:

- (i) de wijze waarop de labels systematisch worden aangepast;
- (ii) een criterium om uit te maken of een label definitief is.

Konkreet luidt de algoritme als volgt.

Begin met de label van s ($\text{lab}(s)$) de waarde 0 te geven en de labels van alle andere knopen de waarde ∞ . Markeer $\text{lab}(s)$ als definitief.

Bij elke iteratiestap worden de tijdelijke labels gewijzigd.

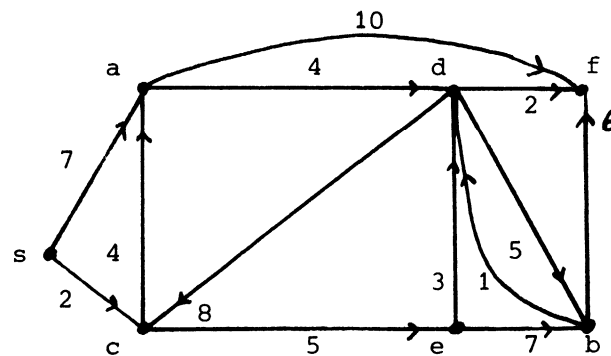
Elke $\text{lab}(j)$, die nog niet zijn eindwaarde heeft bereikt, krijgt een nieuwe waarde gelijk aan het minimum van de oude waarde van $\text{lab}(j)$ en het getal $w_{ij} + \text{lab}(i)$, als i de knoop is, die bij de vorige iteratiestap een eindlabel heeft gekregen. Daarna wordt die tijdelijke label, die de kleinste waarde heeft, als definitief gemarkeerd. Wanneer er meer dan één label is met een minimale waarde, doet het er niet toe welke genomen wordt.

De iteratie gaat door totdat $\text{lab}(f)$ zijn eindwaarde heeft gekregen.

Deze waarde geeft de lengte aan van het kortste pad van s naar f .

We illustreren de methode aan de hand van een voorbeeld.

Voorbeeld 20



Men wil het kortste pad berekenen van s naar f . Door toepassing van de besproken methode vinden we het volgende overzicht van de labels. Een onderstreepte waarde duidt de eindwaarde aan van de betreffende label.

s	a	b	c	d	e	f
<u>0</u>	∞	∞	∞	∞	∞	∞
	7	∞	2	∞	∞	∞
	7	∞	<u>2</u>	∞	∞	∞
	6	∞		∞	7	∞
	<u>6</u>	∞		∞	7	∞
		∞		10	7	16
		∞		10	<u>7</u>	16
		14		10		16
		14		<u>10</u>		16
		14				12
		14				<u>12</u>

□

We zullen nu bewijzen, dat de eindwaarde van een label die hoort bij een knoop v altijd de lengte van het kortste pad aangeeft van s naar v (ofwel de afstand van s naar v).

Evenals in 4.1.2 noteren we de lengte van het kortste pad van a naar b met $d(a,b)$.

Als er geen pad is van a naar b definiëren we

$d(a,b) = \infty$ ($b \neq a$), terwijl $d(a,a) = 0$.

Verder onderstellen we dat de knopen gerangschikt zijn naar hun afstand tot s:

$$(s=) v_{i_0}, v_{i_1}, \dots, v_{i_{n-1}}$$

Om typografische redenen zullen we, wanneer dat geen verwarring kan wekken, in plaats van v_{i_j} eenvoudig de index i_j schrijven.

Er geldt dus

$$d(s, i_k) \leq d(s, i_{k+1}), \quad \forall k \in \{0, 1, \dots, n-1\}. \quad (4.12)$$

We merken op, dat als $l := d(s, i_{k+1})$ er geldt

$$l = \min_{0 \leq t \leq k} (w_{i_t, i_k} + d(s, i_t)), \quad (4.13)$$

$$l \leq \min_{0 \leq t \leq k} (w_{i_t, i_u} + d(s, i_t)), \quad u > k+1. \quad (4.14)$$

Vlak voordat i_k zijn eindlabel krijgt, en de overige labels worden aangepast, geldt:

$$lab(i_u) = \begin{cases} d(s, i_u) & , 0 \leq u \leq k \\ \min_{0 \leq t \leq k-1} (w_{i_t, i_u} + d(s, i_t)) & , u > k \end{cases} \quad (4.15)$$

en $i_0, i_1, \dots, i_{k-1}$ hebben hun eindlabel, terwijl $i_k, i_{k+1}, \dots, i_{n-1}$ nog een tijdelijk label hebben.

We bewijzen (4.15) en de daarop volgende bewering over eindlabels met inductie naar k.

Voor $k=0$ is (4.15) triviaal, omdat wegens de initialisatie van de labels geldt $\text{lab}(i_0) = 0$ en $\text{lab}(i_u) = \infty$ voor $u > 0$. Definieer hier - zoals wel vaker gebeurt - $\min \emptyset = \infty$ (beter $\inf \emptyset = \infty$).

Neem nu aan, dat (4.15) geldt vlak voordat i_k zijn eindlabel krijgt. Direkt nadat i_k zijn eindlabel ($= d(s, i_k)$) gekregen heeft, worden de tijdelijke labels aangepast:

$$\begin{aligned} \text{lab}(i_u) &:= \min(\text{lab}(i_u), w_{i_k, i_u} + \text{lab}(i_k)) \\ &= \min \left[\min_{0 \leq t \leq k-1} (w_{i_t, i_u} + d(s, i_t)), \right. \\ &\quad \left. w_{i_k, i_u} + d(s, i_k) \right] \\ &= \min_{0 \leq t \leq k} (w_{i_t, i_u} + d(s, i_t)), \end{aligned}$$

voor $u \geq k+1$. In het bijzonder is dus $\text{lab}(i_{k+1}) = d(s, i_{k+1})$ volgens (4.13).

Volgens het principe van volledige inductie is hiermee het bewijs van (4.15) geleverd.

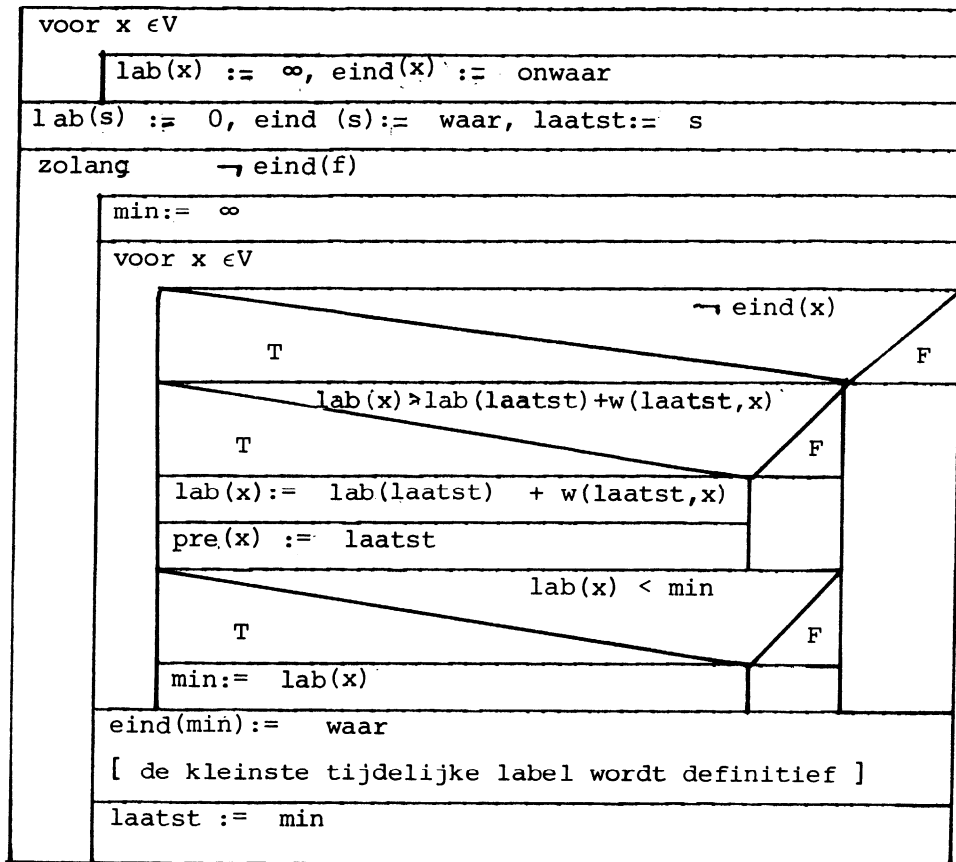
Door bij elke iteratiestap bij te houden welke buurknoop voor de eindwaarde van de betreffende label heeft gezorgd, zijn we in staat om naast zijn lengte ook het kortste pad zelf te bepalen, in de vorm van een rij knopen incident met dat pad. Een en ander is geformuleerd als algoritme 4.4 waaraan de naam van Dijkstra is verbonden en die een verbetering is van een oudere versie van Moore. De enkelvoudige digraaf $G = (V, E)$ is gerepresenteerd als een $(n \times n)$ -gewichtenmatrix W met elementen $W_{ij} = w_{ij}$. In de algoritme wordt hiervoor een tweedimensionale array $w(1:n, 1:n)$ gebruikt. Als het paar (i, j) geen tak is wordt W_{ij} gelijk gesteld aan ∞ . Een boolean array $\text{eind}(1:n)$ dient om aan te geven welke labels tijdelijk (onwaar) zijn en welke definitief (waar). De variabele laatst geeft de laatste knoop aan, waarvan de label zijn eindwaarde heeft bereikt. Na afloop van de algoritme geeft $\text{lab}(f)$ de afstand $d(s, f)$. Wil men ook het pad zelf weten, dan moet men laten afdrukken de rij

$$f, \text{pre}(f), \text{pre}(\text{pre}(f)), \dots, s,$$

die de knopen van het pad in omgekeerde volgorde geeft.

Als $\text{lab}(f) = \infty$, dan is er geen pad van s naar f en minstens één

van de waarden $\text{pre}(x)$ zal niet gedefinieerd zijn.



Algoritme 4.4

De zolang-opdracht in algoritme 4.4 wordt hoogstens $n-1$ keer uitgevoerd. Het aantal opdrachten, dat per keer wordt uitgevoerd is $O(n)$, wegens de herhaling voor $x \in V$. Algoritme 4.4 is dus van de orde $O(n^2)$.

Opmerking

Algoritme 4.4 werkt als alle gewichten groter dan of gelijk aan 0 zijn. Zijn er negatieve gewichten, dan kan het gebeuren, dat het kortste pad niet wordt gevonden. De reden is, dat als een label eenmaal definitief is geworden, hij niet meer veranderd kan worden. Takken met negatief gewicht "verder weg in de graaf" die een korter pad kunnen veroorzaken, hebben dan dus geen invloed meer op zo'n label.

4.4.2. Kortste paden tussen alle paren knopen

Wil men tussen alle geordende paren knopen de kortste paden vinden, dan kan men als volgt te werk gaan. Eerst verandert men de zolang-opdracht in algoritme 4.4 als volgt

zolang $\forall_{x \in V} \rightarrow \text{eind } (x) \text{ doe}$
(Met $\forall_{x \in V}$ bedoelen we de logische of-operatie voor n voorwaarden.)

De algoritme levert dan afstanden van s naar alle andere knopen van G (Wil men de paden zelf ook hebben, dan dient men elke keer als een label definitief wordt het bijbehorende pad te laten afdrukken). De zolang-opdracht wordt nu precies n-1 keer uitgevoerd en de algoritme is nog steeds van de orde $O(n^2)$. De gewijzigde algoritme 4.4 laat men nu uitvoeren voor alle knopen van G als beginknoop. Dit levert dus uiteindelijk een algoritme op van $O(n^3)$.

In plaats van deze algoritme bespreken we nog een andere algoritme, die eveneens van orde $O(n^3)$ is. De algoritme bewerkstelligt de transformatie van de gewichtenmatrix W naar de matrix W^* , waarvan het element w_{ij}^* de lengte aangeeft van het kortste pad van knoop i naar knoop j. (De knopen zijn in willekeurige volgorde genummerd van 1 tot en met n.)

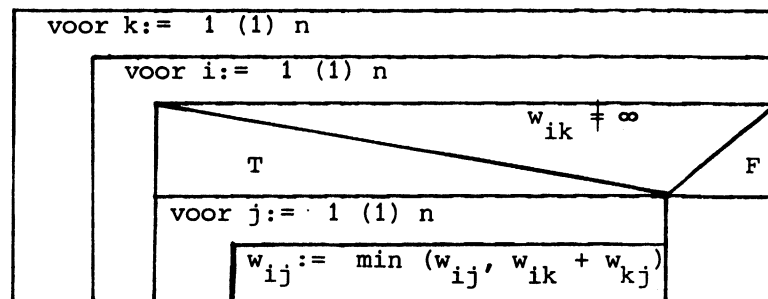
We definiëren de volgende rij matrices $W^{(k)}$ met elementen $w_{ij}^{(k)}$ ($k \geq 0$)

$$w_{ij}^{(k)} = \min (w_{ij}^{(k-1)}, w_{ik}^{(k-1)} + w_{kj}^{(k-1)}), \quad k > 0, \quad (4.16)$$

$$w_{ij}^{(0)} = w_{ij}.$$

(Vergelijk de definitie van de matrices $C^{(k)}$ in (4.9) en (4.10).)

Met inductie naar k bewijst men weer gemakkelijk dat $w_{ij}^{(k)}$ de lengte is van het kortste pad van i naar j, waarbij alleen gebruik gemaakt wordt van de knopen $\{1, 2, \dots, k\}$. Op deze eigenschap is de volgende algoritme gebaseerd (Floyd).



Algoritme 4.5

Opmerking

Algoritme 4.5 werkt ook in het geval, dat er negatieve gewichten zijn. Echter moet men zich realiseren, dat een kortste padlengte niet bestaat als in het pad een circuit kan worden opgenomen, waarvan de totale lengte negatief is.

In feite kan men dan de padlengte zo klein maken als men wil, door maar vaak genoeg dit circuit af te leggen.

In algoritme 4.5 worden alleen de afstanden tussen de knopen uitgerekend en niet de paden zelf. Om die tegelijkertijd te bepalen voert men in een padmatrix P , waarvan het element P_{ij} gelijk is aan de label van de knoop, die na i komt in het kortste pad van i naar j .

Het kortste pad van i naar j wordt dan gegeven door de rij takken

$$(v_1, v_2), (v_2, v_3), \dots, (v_{r-1}, v_r),$$

$$\text{met } v_1 = i, v_r = j, v_t = P_{v_{t-1}, j}, \text{ voor } t > 1.$$

We voeren naast de matrices $W^{(k)}$ nu een rij matrices $P^{(k)}$ in

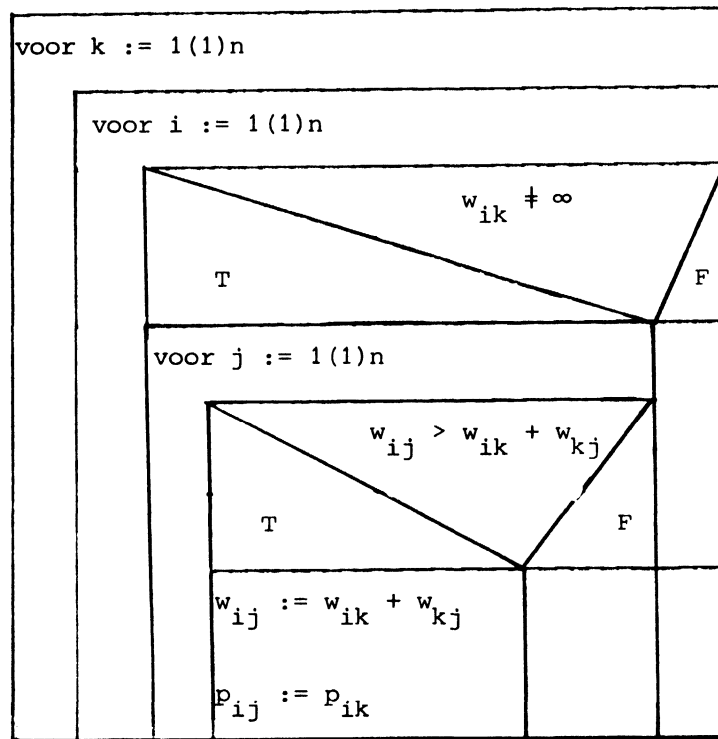
$$P_{ij}^{(k)} = \begin{cases} P_{ij}^{(k-1)}, & \text{als } W_{ik}^{(k-1)} + W_{kj}^{(k-1)} \geq W_{ij}^{(k-1)}, \\ P_{ik}^{(k-1)}, & \text{als } W_{ik}^{(k-1)} + W_{kj}^{(k-1)} < W_{ij}^{(k-1)}, \end{cases} \quad (4.18)$$

voor $k > 0$, terwijl we voor $k = 0$ definiëren

$$P_{ij}^{(0)} = \begin{cases} j, & \text{als } w_{ij} \neq \infty, \\ 0, & \text{als } w_{ij} = \infty. \end{cases} \quad (4.19)$$

Het matrixelement $P_{ij}^{(k)}$ geeft de tweede knoop in het kortste pad van i naar j , waarbij alleen knopen uit de verzameling $\{1, 2, \dots, k\}$ als tussenliggende knopen zijn toegestaan. Bewijs dit zelf met volledige inductie naar k .

Met gebruikmaking van $P^{(k)}$ kunnen we algoritme 4.5 uitbreiden tot algoritme 4.6.



Algoritme 4.6

De initialisatie van p_{ij} volgens (4.19) is in algoritme 4.6 weggelaten, evenals die van w_{ij} in 4.5 en in 4.6. Beide bewerkingen zijn van $O(n^2)$ en zijn voor grote n te verwaarlozen ten opzichte van de rest van de algoritmen.

Opmerking

Door in algoritme 4.5 een variabele $\max$ op te nemen die bijhoudt welke afstand de grootste is (of waaraan men na afloop van 4.5 de waarde toekent van het grootste array-element $w(i,j)$) kan men tevens de diameter van een graaf bepalen.

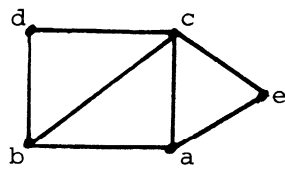
4.5. Breadth-first techniek

De in 4.4.1 besproken algoritme om het kortste pad te zoeken tussen twee knopen is verwant aan de in 4.2.3 genoemde naaste-buur algoritme. In beide wordt op een zogenaamde breadth-first manier gezocht naar een volgende partiële oplossing. In termen van een boomstructuur betekent dit, dat er eerst horizontaal gezocht wordt (ofwel op hetzelfde niveau) totdat één of ander criterium vervuld is en dat er dan pas in verticale richting (naar het volgende niveau) wordt verder gegaan. Dit zoekproces kan dus gezien worden als de tegenhanger van het depth-first zoeken uit 4.2.5.

We lichten de breadth-first zoekmethode (BFS) nog eens wat uitge-

breider toe bij het konstrueren van een opspannende boom van een ongerichte graaf G (zie ook 4.2.5).

In figuur 4.10 is opnieuw de graaf getekend uit figuur 4.3.



a: b, c, e ;

b: a, c, d ;

c: a, b, d, e ;

d: b, c ;

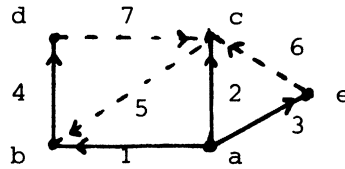
e: a, c

Figuur 4.10

We beginnen met het toekennen van het rangnummer 0 aan alle knopen, om aan te geven dat er nog geen knoop bezocht is. Vervolgens onderzoeken we de takken incident met knoop a en gaan na of ze leiden naar knopen die nog niet eerder bezocht zijn. In dit geval levert dat op de takken (a,b) , (a,c) en (a,e) (in die volgorde). De daarbij optredende knopen b, c en e krijgen in de volgorde waarin ze bezocht zijn een nieuw rangnummer en de bijbehorende takken nemen we op in de partiële boom. Daarna gaan we verder met de ongebruikte takken van b. De tak (b,c) leidt naar een reeds bezochte knoop met hoger rangnummer dan b. Deze laten we weg, omdat we ons als opgave stellen een opspannende boom te konstrueren, alsmede een verzameling back edges en cross edges. Deze laatste twee categorieën takken hebben altijd een beginknoop die een hoger rangnummer heeft dan hun eindknoop. (Zie ook de opmerking in 4.2.5.)

De tak $\{b,c\}$ uit de ongerichte graaf G zal daarom als een tak (c,b) worden opgenomen in onze konstruktie en zal pas verschijnen bij onderzoek van knoop c. De laatste tak incident met b is (b,d) . Deze leidt naar een nog niet eerder bezochte knoop d, die dus nu een rangnummer krijgt, terwijl (b,d) wordt opgenomen in de boom. Hierna wordt verder gegaan met de knoop die na b aan de beurt is. Dat is knoop c, want die heeft na b een rangnummer ($\neq 0$) gekregen.

Uiteindelijk komen we tot de volgende gerichte opspannende boom,



Figuur 4.11

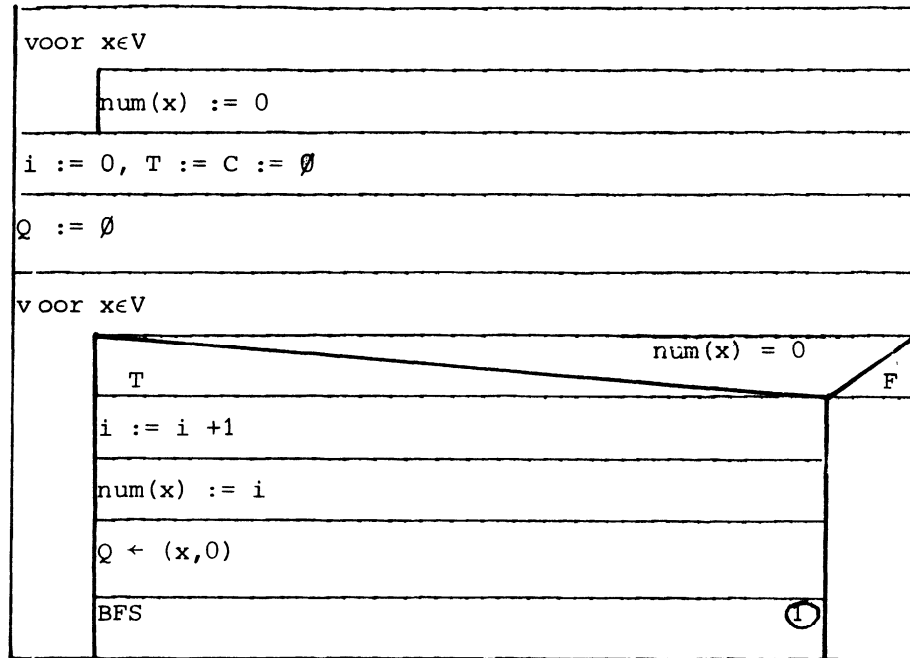
De takken zijn genummerd in de volgorde waarin ze zijn ontstaan. Gestippelde takken zijn cross edges en behoren niet tot de opspannende BFS-boom.

Karakteristiek voor de BFS-methode is dus, dat een knoop een rangnummer krijgt, zodra zijn voorouder bezocht wordt en dat de knopen ook in die volgorde worden afgehandeld. (De volgorde der takken, waar één bepaalde knoop incident mee is, is niet van belang en hangt af van de representatie van de graaf). Het zal duidelijk zijn, dat men bij de formulering van een dergelijk zoekproces uitstekend gebruik kan maken van een queue (zie [6]). Immers het element, dat het langst in de queue verblijft, is het eerst aan de beurt.

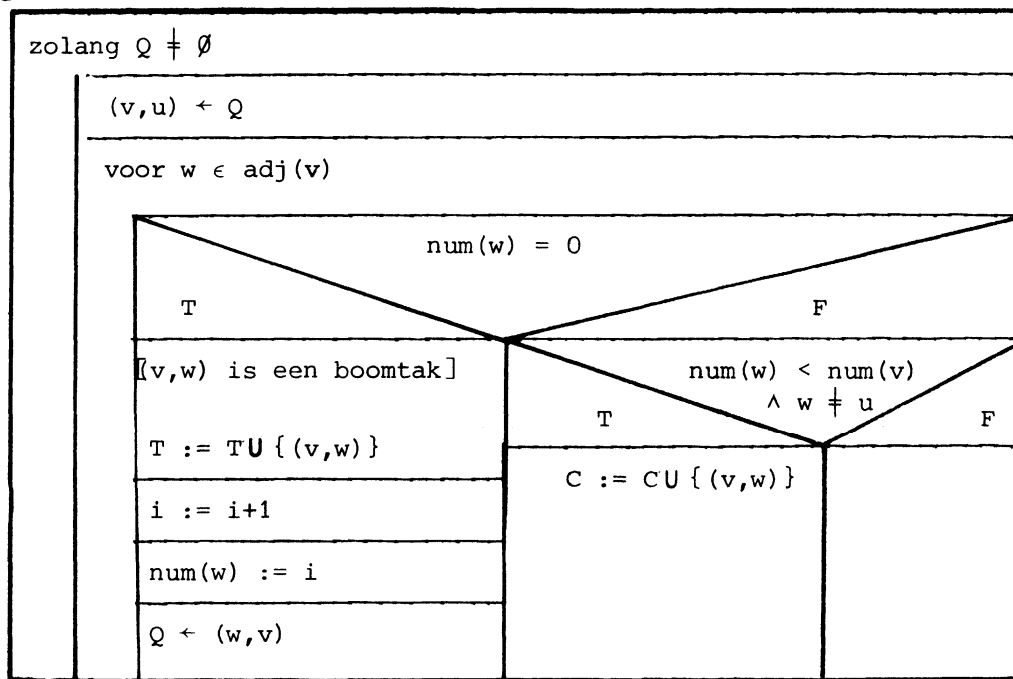
In algoritme 4.7 wordt een queue Q gebruikt. In Q worden de takken (v,w) gezet, die incident zijn met v , waarbij v de knoop is die behandeld wordt, terwijl $w \in \text{Adj}(v)$ (let erop, dat de tak (v,w) als het paar (w,v) aan Q wordt toegevoegd, zodat de voorouder dus achteraan staat).

De verzameling T bevat na afloop de takken van de opspannende boom.

De verzameling C bevat alle gevonden cross edges.



① BFS



Algoritme 4.7

De opdrachten $(v, u) \leftarrow Q$ en $Q \leftarrow (w, v)$ hebben de betekenis dat de kop van Q wordt weggehaald uit Q en wordt toegekend aan (v, u) , respectievelijk dat (w, v) aan de staart van Q wordt gezet.

Opmerkingen

1. Ga na dat het aantal bewerkingen in algoritme 4.7 van de orde $O(m+n)$ is en dus vergelijkbaar is met de complexiteit van een depth-first onderzoek.
2. Probeer voor de algoritme uit 4.4.1 de BFS-methodiek te interpreteren in termen van een boomstructuur, zoals we dat voor de DFS-algoritme voor een opspannende boom in 4.2.5 deden.

4.6. Circuits in een graaf

In deze paragraaf zullen we met circuit steeds een elementair circuit bedoelen, waarin dus elke knoop van G ($\neq$ beginknoop) hoogstens één keer voorkomt. Verder zullen we steeds ongerichte, samenhangende grafen beschouwen, die bovendien enkelvoudig zijn.

De af te leiden stellingen en algoritmen kunnen zonder veel moeite gegeneraliseerd worden voor niet-samenhangende, ongerichte enkelvoudige grafen. Neemt men een gerichte graaf of een niet-enkelvoudige graaf dan kan de theorie niet zonder meer toegepast worden.

Voorts wordt er met nadruk opgewezen dat anders dan in 4.1 in de nu volgende paragrafen de volgorde van de takken in een circuit er niet toe zal doen, zoals in het vervolg nog duidelijk gemaakt zal worden. Twee circuits zullen daarom identiek zijn in de kontekst van 4.6 als hun takkenverzamelingen dezelfde zijn. In dit verband bezit een graaf dus ofwel géén of één Eulercircuit.

4.6.1. Circuitruimte

We beschouwen de verzameling van elementaire circuits, die gekonstrueerd kunnen worden uit de takken van een graaf G . Aan zo'n circuit wordt géén richting toegekend. Alleen de in het circuit aanwezige takken zijn relevant, zodat we het circuit ook kunnen opvatten als een deelverzameling van EG .

Om tot een opteloperatie voor circuits te komen definiëren we eerst het symmetrische verschil van twee verzamelingen.

Definitie

Het symmetrische verschil van de verzamelingen A en B is de verzameling

$$A \oplus B = \{x \mid x \in A \cup B, x \notin A \cap B\}.$$

In dit verband zegt men ook wel dat het element x in A òf in B moet zitten, waarbij òf het expliciete of voorstelt.

Vat men elementaire circuits op als takkenverzamelingen, dan is met bovenstaande definitie dus de som van twee (of meer) circuits vastgelegd. Zijn C_1 en C_2 twee circuits van een graaf G en telt men ze op volgens de aangegeven manier dan spreekt men over de ringsom van C_1 en C_2 en men schrijft

$$C_1 \oplus C_2.$$

Definitie

De circuitruimte $\mathcal{C}$ van een graaf G is de verzameling van elementaire circuits van G en alle mogelijke ringsommen van deze circuits.

Een direct gevolg van deze definitie is, dat men twee elementen van $\mathcal{C}$ kan optellen en dat het resultaat opnieuw in $\mathcal{C}$ ligt.

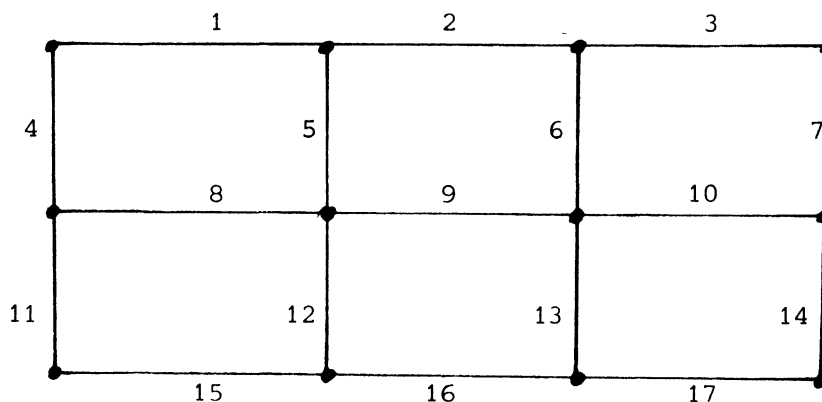
Men formuleert dit als

Stelling (4.15)

De circuitruimte $\mathcal{C}$ is gesloten onder de operatie (ring) som.

Voorbeeld 21

We gaan uit van de volgende graaf :



$$C_1 = \{1, 5, 9, 13, 16, 15, 11, 4\} \quad ,$$

$$C_2 = \{8, 5, 2, 6, 13, 16, 15, 11\} \quad ,$$

$$C_1 \oplus C_2 = \{1, 2, 6, 9, 8, 4\} \quad ,$$

$$C_3 = \{1, 2, 3, 7, 10, 9, 8, 4\} \quad ,$$

$$C_4 = \{2, 6, 9, 5\} \quad ,$$

$$C_3 \oplus C_4 = \{1, 5, 8, 4, 3, 7, 10, 6\} \quad ,$$

$$C_5 = \{8, 12, 15, 11\} \quad ,$$

$$C_4 \oplus C_5 = \{2, 6, 9, 5, 8, 12, 15, 11\} \quad . \quad \square$$

Uit voorbeeld 21 blijkt dat de ringsom van twee elementaire circuits opnieuw een elementair circuit kan zijn, maar ook een enkelvoudig circuit of de vereniging van disjunkte elementaire circuits. Algemeen blijkt de volgende stelling te gelden.

Stelling (4.16)

Elk element van $\mathcal{C}$, ongelijk aan de lege verzameling, is te schrijven als de vereniging van een aantal elementaire circuits, die géén takken gemeen hebben (takdisjunkte elementaire circuits).

Bewijs

Neem een willekeurig element $C \in \mathcal{C}$, $C \neq \emptyset$. Uit de konstruktie van $\mathcal{C}$ volgt, dat elke knoop van C een even valentie heeft. Als alle knopen valentie 2 hebben is C de vereniging van een aantal volledig disjunkte elementaire circuits.

Stel dat niet alle valenties gelijk zijn aan 2. Neem een willekeurige knoop uit een komponent waarin minstens één valentie ≥ 4 is. Neem met deze knoop als beginknoop "aansluitende" takken, totdat men een knoop twee maal is tegengekomen (dit hoeft niet de beginknoop te zijn). Verwijder het aldus gekonstrueerde elementaire circuit uit de graaf. De

overblijvende graaf heeft opnieuw slechts even valenties. Ga door met het afsplitsen van elementaire circuits, totdat men een graaf overhoudt, waarvan alle knopen een valentie 2 hebben. $\square$

Opmerkingen

1. In het algemeen zal men een element $C \in \mathcal{C}$ op meer dan één manier kunnen schrijven als de vereniging van een aantal takdisjunkte elementaire circuits.
2. Met dezelfde redenering als in het laatste bewijs is aan te tonen, dat elk enkelvoudig circuit in $\mathcal{C}$ zit. Uit de definitie van $\mathcal{C}$ volgt dan dat $\mathcal{C}$ ook is op te vatten als de verzameling van alle enkelvoudige circuits en alle (volledig) disjunkte verenigingen van enkelvoudige circuits.

In het nu volgende gaan we het rekenen met elementen van $\mathcal{C}$ formaliseren.

Bij de takkenverzameling EG van de graaf G zullen we een vektorruimte definiëren. Daarvoor hebben we nodig het lichaam (zie [5,8]) $\mathbb{Z}_2 = \{0,1\}$ waarin de optel- en vermenigvuldigingsoperatie gedefinieerd zijn als

$$\begin{aligned} 0 + 0 &= 1 + 1 = 0 & , & \quad 0 + 1 = 1 + 0 = 1, \\ \text{respektievelijk} \\ 0 \cdot 0 &= 0 \cdot 1 = 1 \cdot 0 = 0 & , & \quad 1 \cdot 1 = 1. \end{aligned}$$

Opmerking

Het lichaam $\mathbb{Z}_2$ staat ook bekend als het lichaam der gehele getallen modulo 2 en wordt vaak met $\text{GF}(2)$ aangeduid.

We labelen de takken van G met de labels $e_1, e_2, \dots, e_m$. Zij $A \subseteq E$ dan kunnen we A karakteriseren met een vektor $\underline{a}$ met componenten a_i ($1 \leq i \leq m$) waarvoor geldt

$$a_i = \begin{cases} 1 & , \text{ als } e_i \in A, \\ 0 & , \text{ als } e_i \notin A. \end{cases}$$

De vektor $\underline{a}$ heet de karakteristieke vektor van A (zie ook hoofdstuk 3). Twee karakteristieke vectoren worden opgeteld door hun componenten modulo 2 op te tellen.

In hoofdstuk 2 van [8] werden vektorruimten over het lichaam $\mathbb{R}$ geïntroduceerd. Hier voeren we een vektorruimte over $\mathbb{Z}_2$ in, dat wil zeggen dat de vectoren kunnen worden vermenigvuldigd met koëffi-

cienten uit $\mathbb{Z}_2$. In het bijzonder definiëren we $0 \cdot \underline{a} = \underline{0}$ en $1 \cdot \underline{a} = \underline{a}$.

Definitie

De tak-vektorruimte $\mathcal{V}(EG)$ van een graaf G is de vektorruimte bestaande uit de karakteristieke vectoren van deelverzamelingen van EG , over het lichaam $\mathbb{Z}_2$.

Ga zelf na, dat aan alle definitie-eigenschappen van een vektorruimte is voldaan.

Bij elke $C \in \mathcal{C}$ hoort nu ook een vektor en wel de karakteristieke vektor $\underline{c}$ van de takkenverzameling van C . Uit stelling (4.15) volgt dat de circuits een vektordeelruimte van $\mathcal{V}(EG)$ vormen. Deze deelruimte noemen we de circuitdeelruimte en geven hem ook met $\mathcal{C}$ aan. De nulvektor correspondeert met de lege verzameling, die we in dit verband opvatten als een circuit zonder takken.

We tonen nu aan, dat er een nauw verband bestaat tussen de circuitdeelruimte en de opspannende bomen van een graaf G .

Zij T een opspannende boom van een graaf G . Elke tak uit $EG-ET$ kreëert volgens stelling (4.8) een elementair circuit.

Een dergelijke verzameling circuits noemen we een stelsel fundamentealcircuits ofwel een fundamenteaalstelsel (van circuits). Elk fundamenteaalstelsel bevat $m-n+1$ circuits.

Stelling (4.17)

Het stelsel fundamenteaal-circuits horend bij een opspannende boom T van G is een basis van de circuitdeelruimte van G .

Bewijs

Eerst tonen we aan dat elk element uit $\mathcal{C}$ als lineaire combinatie van fundamentealcircuits kan worden geschreven.

Neem aan dat $E_1 = \{e_1, e_2, \dots, e_j, e_{j+1}, \dots, e_k\}$

de takken zijn van een element C_1 , zodanig gerangschikt, dat

$e_1, e_2, \dots, e_j \notin ET$ (koorden) en $e_{j+1}, e_{j+2}, \dots, e_k \in ET$.

In het fundamenteaalstelsel is een circuit dat e_1 bevat, een

circuit dat e_2 bevat, ..., een circuit dat e_j bevat. Zij C_2 de som van

deze circuits met E_2 als takkenverzameling. Als $E_1 \neq E_2$ is de som $E_1 \oplus E_2$ niet leeg.

Omdat E_2 de koorden $e_1, e_2, \dots, e_j$ van T bevat, maar géén andere

koorden zal $E_1 \oplus E_2$ slechts takken van T bevatten.

Omdat $C_1 + C_2$ een element van $\mathcal{C}$ is en dus volgens stelling (4.16) op te vatten is als vereniging van elementaire circuits geeft dit een tegenspraak. Dus $E_1 = E_2$ en C_1 is een lineaire combinatie van fundamentealcircuits.

Aangezien de fundamentealcircuits bij de opbouw van E_2 éénduidig bepaald zijn, is de lineaire combinatie dus ook éénduidig bepaald. $\square$

Een gevolg van deze eigenschap is, dat geldt

$$\dim \mathcal{C} = m - n + 1. \quad (4.20)$$

Stelling (4.18)

Vatten we de elementen van de incidentiematrix B als elementen van $\mathbb{Z}_2$ op, dan geldt

$$\text{rang } (B) = n-1.$$

Bewijs

Noem de i^{de} rij van de incidentiematrix $\underline{b}_i$ ($1 \leq i \leq n$). Aangezien elke kolom van B precies twee enen bevat geldt

$$\sum_{i=1}^n \underline{b}_i = \underline{0},$$

en dus is $\text{rang } (B) \leq n-1$.

Stel $\text{rang } (B) < n-1$. Dan zou er voor één of andere waarde van k gelden

$$\sum_{i \neq k} \beta_i \underline{b}_i = \underline{0},$$

met niet alle β_i ($i \neq k$) gelijk aan nul. Omdat G samenhangend is, is er zeker een v_j te vinden die naburig is met v_k . Dus is er een $s \in \{1, 2, \dots, m\}$ met $B_{js} = B_{ks} = 1$, terwijl $B_{is} = 0$ als $i \neq j$ en $i \neq k$.

Uit $\sum_{i \neq k} \beta_i \underline{b}_i = \underline{0}$ volgt onder andere $\sum_{i \neq k} \beta_i B_{is} = 0$.

Omdat B_{ks} niet optreedt in de sommatie moet gelden $\beta_j = 0$. Uit het samenhangend zijn van G volgt dan door bovenstaande redenering herhaaldelijk toe te passen, dat $\beta_i = 0$, $\forall i \neq k$. Tegenspraak.

Dus $\text{rang } (B) = n-1$. $\square$

Op analoge wijze als de tak-vektorruimte $\mathcal{V}(EG)$ van G voeren we in de knoop-vektorruimte $\mathcal{V}(VG)$.

Definitie

De knoop-vektorruimte $\mathcal{V}(VG)$ van een graaf G is de vektorruimte bestaande uit de karakteristieke vectoren van deelverzamelingen van VG over het lichaam $\mathbb{Z}_2$.

Omdat de matrix B n rijen en m kolommen heeft en omdat alle matrix-elementen waarden uit $\mathbb{Z}_2$ hebben, kunnen we B opvatten als een lineaire operator, die $\mathcal{V}(EG)$ afbeeldt in $\mathcal{V}(VG)$:

$$B : \mathcal{V}(EG) \rightarrow \mathcal{V}(VG).$$

Stelling (4.19)

De circuitdeelruimte $\mathcal{C}$ van een graaf G is gelijk aan de kern van B ofwel

$$\mathcal{C} = \text{Ker}(B).$$

Bewijs

Neem $\underline{a} \in \mathcal{C}$. Elke knoop v_i is incident met een even aantal takken van het circuit dat bij $\underline{a}$ hoort. Zij $\underline{b}_i$ de i^{de} rij van B .

Er geldt dan $\underline{b}_i \cdot \underline{a} = 0$ ($1 \leq i \leq n$).

Dat wil zeggen $B\underline{a} = \underline{0}$ ofwel $\mathcal{C} \subset \text{Ker}(B)$.

Omdat $\dim \text{Ker}(B) = \dim \mathcal{V}(EG) - \dim \text{Im}(B) = m - \text{rang}(B) = m - n + 1$

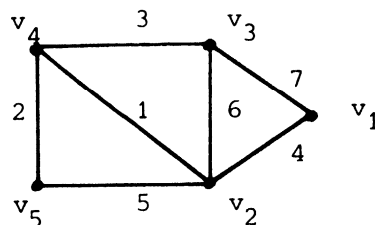
en omdat ook $\dim \mathcal{C} = m - n + 1$ geldt zelfs $\mathcal{C} = \text{Ker}(B)$. □

Opmerking

Zijner k componenten, dan geldt dat $\dim \mathcal{C} = m - n + k$ en $\text{rang}(B) = n - k$.
Toon dit aan.

Voorbeeld 22

Gegeven is de graaf



Een opspannende boom wordt voortgebracht door de takken 1, 2, 3 en 4.

Ga na dat de bijbehorende basis van de circuitdeelruimte gevormd

wordt door de vectoren $(1, 1, 0, 0, 1, 0, 0)$, $(1, 0, 1, 0, 0, 1, 0)$, $(1, 0, 1, 1, 0, 0, 1)$,

en dat alle eigenvectoren van B met eigenwaarde 0 lineaire combinaties zijn van deze drie vectoren. □

4.6.2. Snedenruimte

In deze paragraaf zullen we een begrip behandelen, dat enigszins dual is aan het begrip circuit uit 4.6.1.

Allereerst definiëren we een scheidende verzameling van een graaf.

Definitie

Een scheidende verzameling van een graaf G is een niet-lege verzameling $A \subset EG$, zodanig dat de takdeelgraaf, horend bij $EG - A$ niet-samenhangend is.

(N.B. In het begin van 4.6 is G samenhangend verondersteld.)

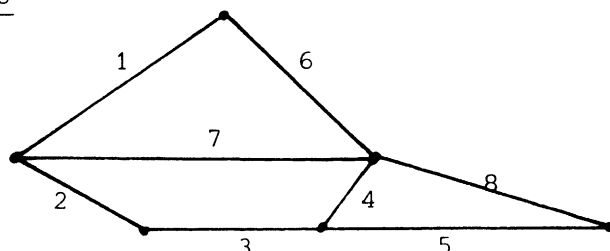
We komen nu tot het begrip snede.

Definitie

Een snede is een scheidende verzameling, waarvan elke deelverzameling géén scheidende verzameling is.

Een gevolg van deze definitie is dat de takdeelgraaf van $EG - S$ (S is een snede) uit precies twee componenten bestaat.

Voorbeeld 23



De verzameling $\{2,3,6,7\}$ is een scheidende verzameling en $\{3,6,7\}$ is een snede. De verzameling $\{5,8\}$ is ook een snede. $\square$

Zij S_1 en S_2 twee sneden. Op geheel analoge wijze als voor elementaire circuits is nu de ringsom $S_1 \oplus S_2$ te definiëren.

Definitie

De snedenruimte $\mathcal{S}$ van een graaf G is de verzameling sneden van G en alle mogelijke ringsommen van deze sneden.

Vanzelfsprekend geldt weer:

Stelling (4.20)

De snedenruimte $\mathcal{S}$ is gesloten onder de operatie ringsom.

De volgende stelling geeft een grafentheoretische karakterisering van de elementen van $\mathcal{S}$ en moet gezien worden als een analogon van stelling (4.16).

Stelling (4.21)

Bij elk element $S \in \mathcal{J}$ ($S \neq \emptyset$) hoort precies één partitie van VG in twee deelverzamelingen V en W, zodanig dat de takken van S juist die takken van EG zijn, die incident zijn met een knoop van V en met een knoop van W. Omgekeerd behoort bij een dergelijke partitie precies één element van $\mathcal{J}$.

(*) Bewijs

(i) We bewijzen eerst dat bij elk element van $\mathcal{J}$ een partitie van VG behoort als in de stelling omschreven.

Als $S \in \mathcal{J}$ een echte snede is, dan definieert S wegens zijn minimaliteit twee deelgrafen van G, waarvan de knopen de deelverzamelingen V respectievelijk W vormen. Dus dan is de stelling waar.

Het is nu verder voldoende om aan te tonen dat de som van twee echte sneden S_1 en S_2 ook de bedoelde eigenschap heeft. Neem aan $S_1 \neq S_2$.

Met S_1 correspondeert een verdeling van VG in twee deelverzamelingen die we V_1 en W_1 zullen noemen. Met S_2 correspondeert een verdeling van VG in V_2 en W_2 .

Beschouw nu de knopenverzamelingen $V_1 \oplus V_2$ en $W_1 \oplus V_2$.

Als $x \notin V_1 \oplus V_2$ dan geldt of $x \notin V_1 \wedge x \notin V_2$ of $x \in V_1 \wedge x \in V_2$. In beide gevallen is dan $x \in W_1 \oplus V_2$. We kunnen dus schrijven

$$VG = (V_1 \oplus V_2) \cup (W_1 \oplus V_2).$$

Verder zijn $V_1 \oplus V_2$ en $W_1 \oplus V_2$ disjunkt. Stel namelijk dat $x \in V_1 \oplus V_2$. Hieruit volgt of $x \in V_1 \wedge x \notin V_2$ of $x \notin V_1 \wedge x \in V_2$. In beide gevallen geldt $x \notin W_1 \oplus V_2$.

De knopen van VG worden zo dus weer verdeeld in 2 disjunkte deelverzamelingen. Men kan nu aantonen, dat $S_1 \oplus S_2$ precies de verzameling takken is, die een knoop van $V_1 \oplus V_2$ verbinden met een knoop van $W_1 \oplus V_2$.

Daartoe moet men eerst laten zien dat een willekeurige tak, die een knoop van $V_1 \oplus V_2$ verbindt met een knoop van $W_1 \oplus V_2$ in $S_1 \oplus S_2$ zit (4 gevallen). Vervolgens moet men aantonen dat elke tak van $S_1 \oplus S_2$ ($S_1 \oplus S_2 \neq \emptyset$) incident is met een knoop van $V_1 \oplus V_2$ en van $W_1 \oplus V_2$. De details laten we hier achterwege (zie bijvoorbeeld [2]).

Als S de som is van een willekeurig aantal sneden kan men een analoog bewijs via inductie geven.

(ii) Laat S de volledige takkenverzameling zijn die twee disjunkte knopenverzamelingen V_1 en W_1 ($V_1 \cup W_1 = VG$) verbindt. Dus S is een scheidende verzameling. Als S minimaal is in dit opzicht is S een snede en is het bewijs geleverd. Is S niet minimaal dan bevat S een echte deelverzameling S_1 , die wel een snede vormt. Als de overblijvende verzameling takken $S - S_1$ opnieuw een scheidende verzameling is kunnen we nogmaals een snede S_2 af-

splitsen. We gaan hiermee door totdat we een verzameling $\bar{S} \subset S$ overhouden, die niet scheidend meer is. Stel nu dat $\bar{S} \neq \emptyset$. Uit het onder (i) bewezene volgt, dat de ringsom van $S_1, S_2, \dots$ twee disjunkte knopenverzamelingen V_2 en W_2 ($V_2 \cup W_2 = VG$) definieert, met de eigenschap dat $S_1 \oplus S_2 \oplus \dots = S - \bar{S}$ precies alle takken bevat tussen V_2 en W_2 . Op geheel dezelfde wijze als in (i) volgt tevens dat nu $V_1 \oplus V_2$ en $W_1 \oplus V_2$ de verzameling VG partitioneren en dat de verzameling takken tussen $V_1 \oplus V_2$ en $W_1 \oplus V_2$ gelijk is aan $(S - \bar{S}) \oplus S = \bar{S}$. Dus $\bar{S}$ is een scheidende verzameling en we hebben een tegenspraak, zodat $\bar{S} = \emptyset$.

(iii) Tenslotte tonen we aan dat met een $S \in \mathcal{F}$ niet meer dan één partitionering van VG kan horen.

Stel (V, W) en $(\bar{V}, \bar{W})$ zijn allebei partitioneringen bij S . Analoog aan de redenering in (i) volgt nu dat $(V \oplus \bar{V}, W \oplus \bar{V})$ een partitionering is bij $S \oplus S = \emptyset$. Als zowel $V \oplus \bar{V}$ als $W \oplus \bar{W}$ niet leeg zouden zijn, zou volgen dat G een niet-samenhangende graaf is, hetgeen in strijd is met de aanname omtrent G . Dus moet wel $\bar{V} = V$ zijn of $\bar{V} = W$ ofwel (V, W) en $(\bar{V}, \bar{W})$ zijn dezelfde partitioneringen.

Uit (i), (ii) en (iii) samen volgt nu de stelling. □

Opmerking

Uit de laatste stelling volgt, dat elk element van S een scheidende verzameling is. Omgekeerd echter hoeft een scheidende verzameling nog géén element van S te zijn. Probeer zelf hiervan een voorbeeld te vinden.

Geheel analoog als voor $\mathcal{V}$ volgt uit stelling (4.20), dat $\mathcal{F}$ is op te vatten als een deelruimte van $\mathcal{V}_{(EG)}$. We spreken in dit verband over $\mathcal{F}$ als over de snedendeelruimte.

Beschouw nu weer een opspannende boom T van een graaf G . Laat men een willekeurige tak van T weg, dan valt T uiteen in 2 subbomen T_1 en T_2 . Dit heeft tot gevolg, dat we kunnen schrijven $VG = VT_1 \cup VT_2$. We nemen nu alle takken van G , die een knoop van VT_1 verbinden met een knoop van VT_2 (deze verzameling is niet leeg!). Eenvoudig is in te zien, dat deze verzameling een snede vormt.

We konkluderen, dat met elke tak uit ET een snede correspondeert. Deze sneden noemen we fundamentealsneden. (Bijzonder geval van stelling (4.21).)

Analoog aan stelling (4.17) hebben we nu

Stelling (4.22)

Het stelsel fundamentealsneden horend bij een opspannende boom T van G is een basis van de snedendeelruimte van G .

Het bewijs van deze stelling is analoog aan het bewijs van stelling (4.19) en wordt als een oefening aan de lezer overgelaten.

Omdat een opspannende boom $n-1$ takken bevat is dus

$$\dim \mathcal{F} = n-1. \quad (4.21)$$

Stelling (4.23)

De snedendeelruimte $\mathcal{F}$ van een graaf G is gelijk aan het orthogonale komplement van $\text{Ker}(B)$, ofwel

$$\mathcal{F} = \text{Ker}(B)^\perp.$$

Bewijs

Neem een snede S . Hiermee correspondeert een vektor $\underline{s} \in \mathcal{F}$.

Deze snede definieert knopenverzamelingen V_1G en V_2G met $V_1G \cup V_2G = VG$ en $V_1G \cap V_2G = \emptyset$, terwijl elke tak van S incident is met een knoop uit V_1G en uit V_2G . (stelling (4.2) (4.21)).

Zij $\underline{b}_i$ de i^{de} rij van B , $1 \leq i \leq |VG|$.

We kunnen schrijven

$$\underline{s} = \sum_{i \in V_1G} \underline{b}_i \quad (= \sum_{i \in V_2G} \underline{b}_i).$$

Immers in de sommatie doen op de j^{de} plaats 2 enen mee als tak j incident is met 2 knopen uit V_1G (V_2G) en 1 één als hij incident is met 1 knoop uit V_1G (V_2G).

Stel $\underline{c} \in \mathcal{C} = \text{Ker}(B)$, dan geldt $B\underline{c} = \underline{0}$ ofwel $\underline{b}_i \cdot \underline{c} = 0, \forall i$. Dus $\underline{s} \cdot \underline{c} = 0$ en bijgevolg $\mathcal{F} \subset \mathcal{C}^\perp$.

Echter is $\dim \mathcal{F} = n-1$ en ook $\dim \mathcal{C}^\perp = m - (m-n+1) = n-1$ en er volgt $\mathcal{F} = \mathcal{C}^\perp = \text{Ker}(B)^\perp$. $\square$

Voorbeeld 24

Beschouw opnieuw de graaf uit voorbeeld 22.

De boom T met takken 1,2,3 en 4 definieert een basis van $\mathcal{J}$, bestaande uit de vectoren

$$(1, 0, 0, 0, 1, 1, 1),$$

$$(0, 1, 0, 0, 1, 0, 0),$$

$$(0, 0, 1, 0, 0, 1, 1),$$

$$(0, 0, 0, 1, 0, 0, 1).$$

Ga na, dat deze vier vectoren alle orthogonaal zijn met de drie vectoren uit voorbeeld 22. $\square$

Voorbeeld 25

Bereken het totale aantal enkelvoudige circuits, met lengte ≥ 3 , in K_5 , als de volgorde van de takken in een circuit er niet toe doet.

Oplossing

Gevraagd wordt naar het aantal vectoren $\underline{c} \in \mathcal{C}(K_5)$, $\underline{c} \neq \underline{0}$, zodanig dat $\underline{c}$ met een enkelvoudig circuit correspondeert. Nu kunnen er in K_5 géén verenigingen van elementaire circuits bestaan, die knoop-disjunkt zijn, omdat hiervoor minstens 6 knopen nodig zijn. Dus corresponderen alle vectoren $\neq \underline{0}$ uit $\mathcal{C}(K_5)$ met enkelvoudige circuits.

Er geldt $\dim(\mathcal{C}) = \frac{1}{2} \times 5 \times 4 - 5 + 1 = 6$.

Dus is het aantal vectoren $\neq \underline{0}$ gelijk aan $2^6 - 1 = 63$. $\square$

Opmerking

Ga zelf na dat in voorbeeld 25 er

1 circuit is van lengte 10;

10 circuits van lengte 7;

15 circuits van lengte 6;

12 circuits van lengte 5;

15 circuits van lengte 4;

10 circuits van lengte 3.

Let ook op de verschillen met voorbeeld 6.

(†) 4.6.3. Genereren van fundamentealcircuits

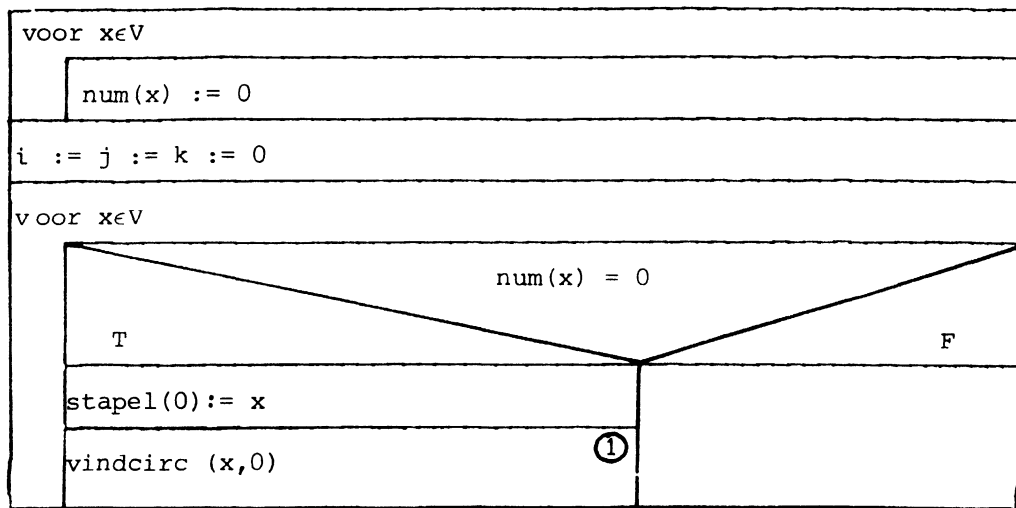
De DFS-algoritme uit 4.2.5 waarmee men een opspannende boom T kan konstrueren is ook toe te passen om een fundamenteaalstelsel van circuits te genereren. Daartoe kan men de terugkeertakken gebruiken, die tegelijk met T worden voortgebracht. Elke terugkeertak vormt een circuit met takken van T . Als de algoritme een terugkeertak (v,w) produceert, bestaat het circuit uit (v,w) en de takken van het kortste pad in de boom van w naar v .

Om de circuits zelf te vinden en af te drukken gebruikt de algoritme een stapel, die alle knopen bevat vanaf de wortel tot en met de knoop die onderzocht wordt. De twee bovenste knopen van de stapel bepalen dus de laatste aan T toegevoegde tak. Stel dat dit de knopen u en v zijn (v is de top van de stapel). Een rekursieve procedure $\text{vindcirc}(v,u)$ tracht dan een circuit te vinden, waarin zitten de knopen u,v en w , waarbij w een buur van v is. Als (v,w) een terugkeertak blijkt te zijn, bestaat het circuit uit w en de p bovenste knopen van de stapel, als w de p -de knoop van boven is.

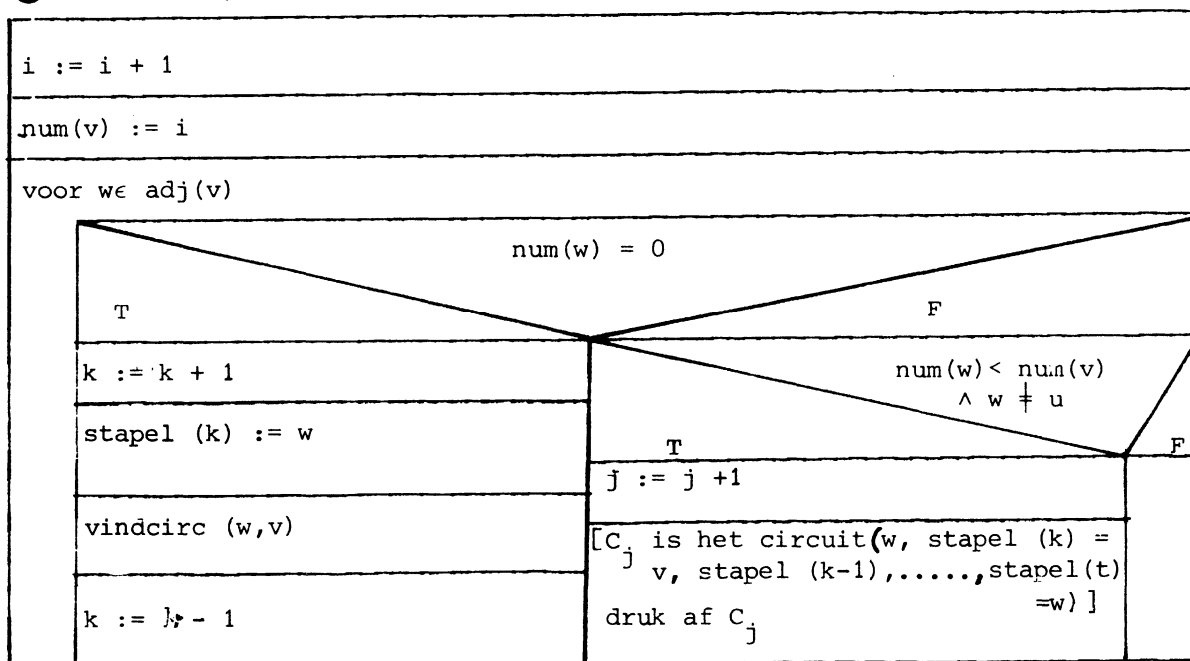
Blijkt (v,w) géén terugkeertak te zijn, maar een boomtak, dan roept de procedure zichzelf aan met $\text{vindcirc}(w,v)$.

De stapel is in de algoritme geïmplementeerd als de array stapel en een pointer k wijst de top aan.

De algoritme staat op naam van Paton.

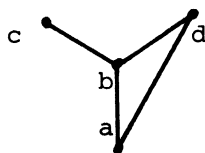


① vindcirc (v,u)



Algoritme 4.8

Ga zelf de werking van deze algoritme na op de graaf uit figuur 4.13 uitgaande van de gegeven nabuurstructuur



Figuur 4.13

a: b,d ;

b: a,c,d;

c: b ;

d: b,a

De kompleksiteit van algoritme 4.8 wordt bepaald door het aantal keren dat een tak van G wordt "behandeld". In de voor-opdracht van de procedure wordt een tak $\{a,b\}$ twee keer onderzocht: één keer als (a,b) en één keer als (b,a) . Echter wordt hij bij het laten afdrukken van de circuits opnieuw behandeld en wel net zo vaak als hij in het stelsel fundamentealcircuits optreedt. Het totaal aantal keren, dat de takken van G op die manier worden behandeld is dus gelijk aan de totale lengte van alle fundamentealcircuits samen. Noem deze lengte t . De algoritme is dus van de orde $O(m + t)$.

Omdat de lengte van één circuit $\leq n$ is, is $t \leq n(m-n+1) \leq \frac{1}{2}n(n-1)(n-2)$. (zie stelling (4.5)).

Ga na dat voor de volledige graaf ("worst case") de algoritme van de orde $O(n^3)$ is.

4.6.4. Genereren van alle circuits

In de vorige paragraaf werden de circuits van een fundamenteaalstelsel van een ongerichte graaf gegenereerd. Deze circuits bepalen, door lineaire combinaties, alle circuits van de graaf, dat wil zeggen (zie 4.6.1) alle enkelvoudige circuits en alle takdisjunkte verenigingen van enkelvoudige circuits. Het totale aantal hiervan bedraagt 2^{m-n+1} en kan dus, afhankelijk van het gedrag van m als functie van n , exponentieel toenemen als n groter wordt.

Men heeft soms behoefte aan een lijst van alle "echte" elementaire circuits (bijvoorbeeld om de datastroom in computerprogramma's te bestuderen). Het is dan niet efficiënt om uit te gaan van een fundamenteaalstelsel, omdat de verzameling van takdisjunkte verenigingen in het algemeen een groot aantal "circuits" bevat, bestaande

uit meer dan één komponent terwijl een enkelvoudig circuit ook nog niet altijd elementair hoeft te zijn.

We bespreken hier een algoritme, die alleen deze circuits genereert en die bovendien geschikt is voor digrafen.

Voor het volgende gaan we uit van een digraaf G . (Een ongerichte graaf moet dan op de bekende manier worden omgezet in een digraaf.) De algoritme berust op een depth-first onderzoek. De knopen worden gelabeld met de getallen 1 tot en met n . We spreken af, dat een circuit begint in die knoop van het circuit, die de kleinste labelwaarde heeft. Van elke knoop van G gaan we nu na welke circuits daar beginnen. Vanwege onze afspraak bestaat er niet het gevaar dat we een circuit vaker dan één keer behandelen.

Veronderstel dat we onderzoeken welke circuits er beginnen in knoop s . Vanuit s konstrueren we een pad

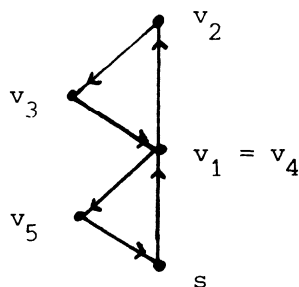
$$(s, v_1, v_2, \dots, v_k)$$

zo dat $v_i > s$, $1 \leq i \leq k$. Als de volgende knoop v_{k+1} van het pad gelijk is aan s is er een circuit gevonden. Nadat het circuit

$$(s, v_1, v_2, \dots, v_k, s)$$

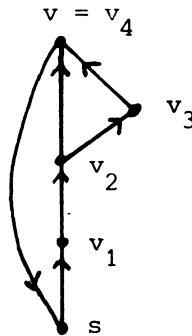
is afgedrukt, onderzoeken we de volgende beschikbare tak, die van v_k uitgaat, dat wil zeggen het pad van s naar v_k wordt verder uitgebreid met een knoop v_{k+1} , waarmee het proces opnieuw begint. Zijn alle beschikbare takken uitgaande van v_k onderzocht, dan gaan we terug naar v_{k-1} , waarvan we de volgende beschikbare tak van onderzoeken (back-tracking), etc. Uiteindelijk keren we terug in s en proberen daarvan de volgende uitgaande tak. Op deze wijze worden alle circuits gevonden, die in s beginnen.

Om te voorkomen, dat het pad van s naar v_k wordt uitgebreid met een knoop v_{k+1} , die gelijk is aan één der knopen $v_1, v_2, \dots, v_k$ wordt een boolean array bereik bijgehouden, zodanig dat $\text{bereik}(v) \equiv \text{false}$ het opnemen van v in het bestaande pad verbiedt. Dit voorkomt het registreren van niet-elementaire circuits, zoals is afgebeeld in figuur 4.14



Figuur 4.14

Zodra dus een knoop v wordt opgenomen in het pad wordt $\text{bereik}(v) := \text{false}$. Deze situatie blijft zo totdat we, via backtracking, weer teruggekeerd zijn in de knoop voorafgaande aan v . Als in deze situatie het pad van s naar v een circuit heeft opgeleverd wordt v weer bereikbaar gemaakt. Heeft het pad van s naar v geen circuit opgeleverd dan blijft $\text{bereik}(v)$ op false staan. Het heeft dan namelijk geen zin om via een ander pad nogeens naar v terug te keren, daar dit zeker geen circuit meer zal opleveren. Als voorbeeld verwijzen we naar figuur 4.15.



Figuur 4.15

Het circuit $(s, v_1, v_2, v_3, v_4, s)$ treedt zeker niet op als niet ook het circuit (s, v_1, v_2, v_4, s) wordt gevonden!

In algoritme 4.9 zijn de knopen van het pad $(s, v_1, v_2, \dots, v_k)$ opgeslagen op de stapel pad. Knoop $v (= v_{k+1})$ wordt op de top van de stapel gezet, waarna het zoeken vervolgd wordt door middel van een rekursieve oproep van de procedure circuit. De knoop wordt van de stapel verwijderd als via backtracking wordt teruggekeerd naar de vorige knoop. Dit gebeurt algoritmisch als de procedure weer verlaten wordt. Bij het op de stapel zetten wordt v onbereikbaar gemaakt, terwijl v weer bereikbaar wordt gemaakt als hij van de stapel wordt afgehaald en als v bovendien aanleiding heeft gegeven tot een circuit. Is dit laatste niet het geval, dan blijft v onbereikbaar. De array bereik houdt alleen het onbereikbaar-zijn bij van de knopen die liggen op het huidige pad van s naar v . Er zijn echter ook knopen die in het verleden onbereikbaar zijn geworden en die niet meer op het huidige pad van s naar v liggen. Er wordt daarom ook van elke knoop $w \in V$ een verzameling $B(w)$ bijgehouden, gedefinieerd door

$$B(w) = \{v \in V \mid (v, w) \in EG \text{ en } v \text{ is onbereikbaar en } v \text{ ligt niet op huidige pad}\}.$$

We vatten nog even samen wat er gebeurt als het pad $(s, v_1, v_2, \dots, v_k)$ wordt uitgebreid met de tak (v_k, v_{k+1}) . Alle handelingen worden verricht door een procedure circuit (v, sukses) . Bij aanroep van de procedure met $v = v_k$ treedt één der volgende gevallen op:

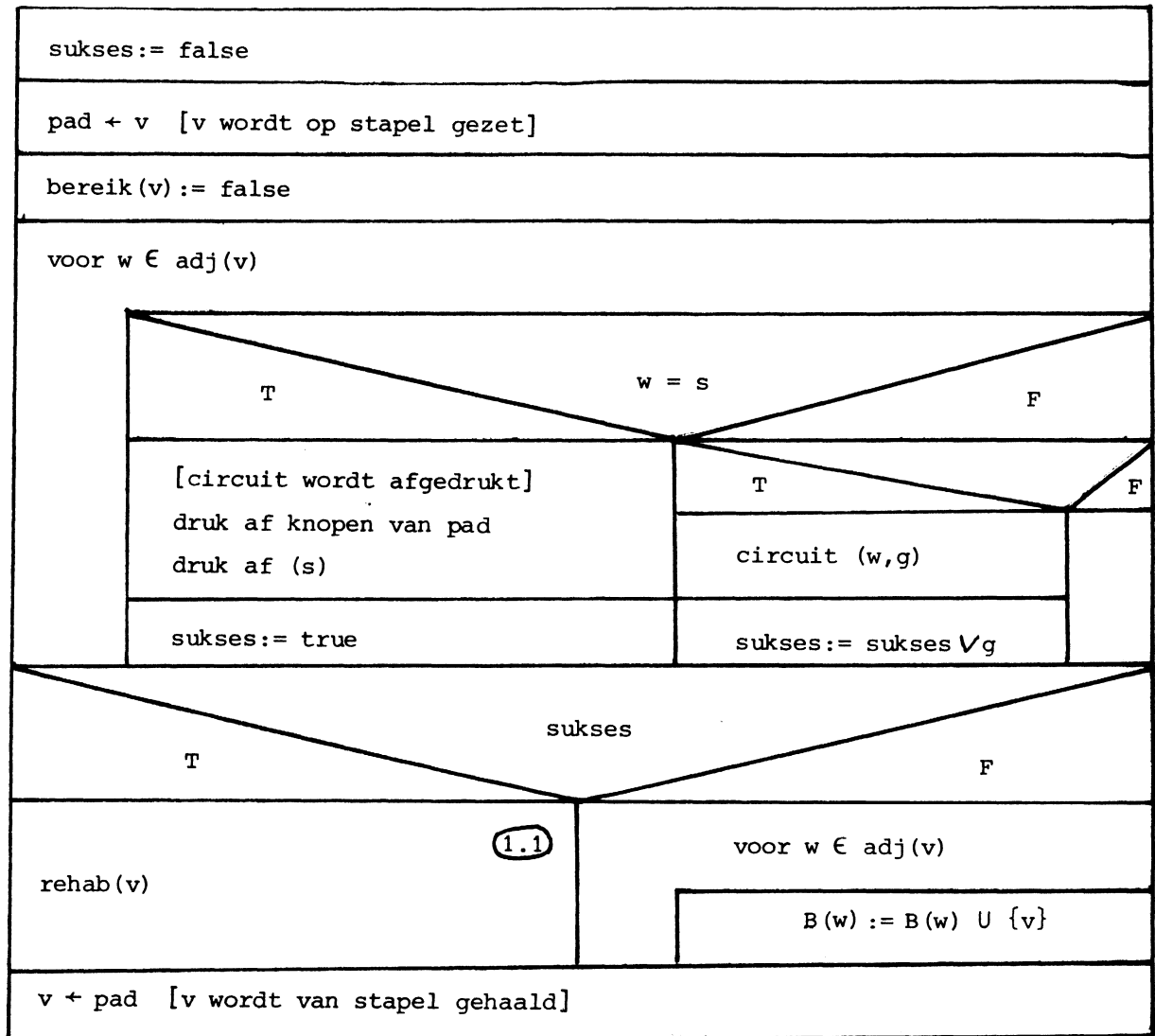
- (i) Als $w (=v_{k+1}) = s$ is er een nieuw circuit gevonden. Dit circuit wordt afgedrukt en een boolean variabele `sukses` wordt op true gezet.
- (ii) Als $w \neq s$ en w is bereikbaar, dan wordt w aan het pad toegevoegd en de procedure circuit wordt opnieuw opgeroepen voor $v = w$.
- (iii) Als w niet bereikbaar is, wordt w overgeslagen en wordt er verder gegaan met de volgende buur van v . Is er niet zo'n buur dan keert de procedure terug naar v_{k-1} . Dat wil zeggen, dat v van de stapel wordt gehaald en eventueel (als `sukses` $\equiv$ true) weer bereikbaar wordt gemaakt. Is `sukses` $\equiv$ false dan blijft v onbereikbaar en voor $w \in VG$ met $(v, w) \in G$ wordt de verzameling $B(w)$ uitgebreid met v .
- (iv) Het opnieuw bereikbaar maken van v (zie (iii)) gebeurt met de procedure `rehab(v)`. Behalve dat `bereik(v)` weer op true gezet moet worden moet ook de verzameling $B(v)$ worden herzien. Immers, als $u \in B(v)$, dan is u onbereikbaar, u is een voorouder van v en u ligt niet op het huidige pad. Daar v bereikbaar wordt gemaakt moet men de mogelijkheid open laten, dat er in een later stadium nog een circuit gevonden wordt dat de knopen u en v - in deze volgorde - bevat. (Bedenk zelf een voorbeeld, waar dit het geval is.) Alle elementen van $B(v)$ moeten daarom weer bereikbaar worden gemaakt. Echter geldt dan hetzelfde argument voor de elementen van $B(u)$, zodat ook die elementen bereikbaar gemaakt moeten worden, etc. De procedure `rehab(v)` roept daarom aan `rehab(u)` en is dus ook rekursief.

De volledige algoritme

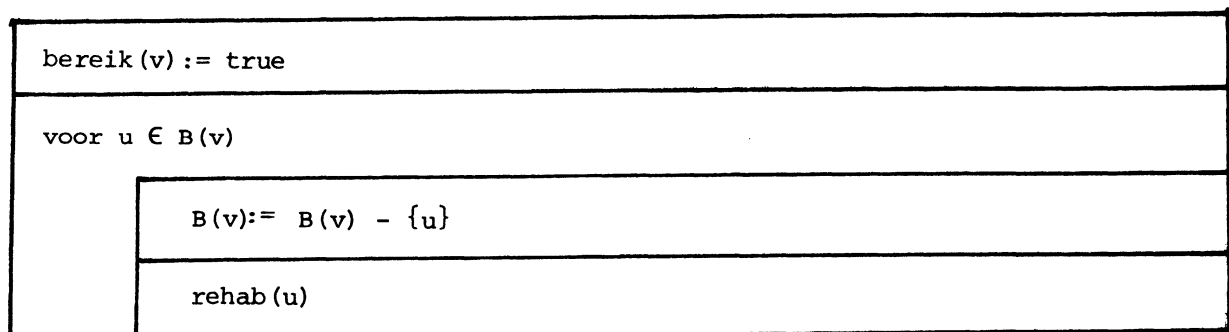
(Johnson) staat twee bladzijden verder. De procedure circuit werkt steeds op een sterk samenhangende komponent. Uit de definitie van sterk samenhangend volgt, dat elk circuit in zo'n komponent zit.) Neem aan dat G_1 de sterk samenhangende komponent is van G waarin knoop 1 zit. De procedure circuit (s, sukses) met $s=1$ genereert eerst alle circuits met knoop 1 als beginknoop. Omdat er geen andere circuits zijn waarin 1 ligt laten we daarna knoop 1 weg uit G en beschouwen de knoopdeelgraaf $\langle 2, 3, \dots, n \rangle$, gegenereerd door de knopen $2, 3, \dots, n$. In deze graaf is 2 de knoop met de kleinste label. Laat nu G_2 de sterk samenhangende

komponent van $\langle 2, 3, \dots, n \rangle$ zijn die knoop 2 bevat. Toepassen van circuit (2, sukses) op G_2 levert alle circuits met beginknoop 2, etc. We gaan hiermee door totdat alleen knoop n is overgebleven.

① circuit (v,sukses)



(1.1) rehab(v)



5. APPENDIX

5.1. Asymptotiek

In de asymptotiek wordt de mate van toenemen onderzocht van reële functies, gedefinieerd op $\mathbb{R}$ (of op een deelverzameling van $\mathbb{R}$), als de onafhankelijke variabele onbeperkt groot of klein wordt of naar een vaste waarde nadert.

In dit diktaat is gebruik gemaakt van een drietal hulpmiddelen, waarmee men het asymptotisch gedrag van functies meer of minder precies kan uitdrukken.

Uit [4] generaliseren we twee begrippen.

Definitie

$f(x) = O(g(x))$ voor $x \rightarrow x_0$ dan en slechts dan als er een konstante c bestaat met

$$\limsup_{x \rightarrow x_0} \left| \frac{f(x)}{g(x)} \right| = c.$$

Definitie

$f(x) = o(g(x))$ voor $x \rightarrow x_0$ dan en slechts dan als

$$\lim_{x \rightarrow x_0} \left| \frac{f(x)}{g(x)} \right| = 0.$$

Als $f(x) = O(g(x))$ zeggen we, dat $f(x)$ niet sneller toeneemt dan $g(x)$, als $x \rightarrow x_0$.

Als $f(x) = o(g(x))$ zeggen we, dat $f(x)$ langzamer toeneemt dan $g(x)$, als $x \rightarrow x_0$.

Verder voeren we in het symbool $\sim$.

Definitie

$f(x) \sim g(x)$ als $x \rightarrow x_0$ dan en slechts dan als

$$\lim_{x \rightarrow x_0} \frac{f(x)}{g(x)} = 1.$$

We zeggen, dat $f(x)$ asymptotisch gelijk is aan $g(x)$.

Voorbeeld 1

Voor $x \rightarrow \infty$ geldt:

$$\sum_{i=0}^n a_i x^i = O(x^n) ,$$

$$\sum_{i=0}^n a_i x^i = o(x^{n+1}) ,$$

$$\sum_{i=0}^n a_i x^i \sim a_n x^n, \text{ als } a_n \neq 0. \quad \square$$

Voorbeeld 2

Met volledige inductie of met behulp van de methoden uit hoofdstuk 2 is te bewijzen dat voor $n \rightarrow \infty$ en voor elke $k \in \mathbb{N}$ geldt:

$$\sum_{i=1}^n i^k \sim \frac{1}{k+1} n^{k+1} , \quad (5.1)$$

$$\sum_{i=1}^n i^k = \frac{1}{k+1} n^{k+1} + o(n^{k+1}) , \quad (5.2)$$

$$\sum_{i=1}^n i^k = \frac{1}{k+1} n^{k+1} + O(n^k) . \quad (5.3)$$

De betrekkingen (5.1) en (5.2) bevatten dezelfde informatie. Betrekking (5.3) verschaft meer informatie dan (5.1) en (5.2). Een nog sterkere uitspraak is

$$\sum_{i=1}^n i^k = \frac{1}{k+1} n^{k+1} + \frac{1}{2} n^k + O(n^{k-1}) . \quad (5.4)$$

Bewijs deze uitspraak zelf. $\square$

Opmerking

Het in het laatste voorbeeld gevolgde principe kunnen we als volgt formaliseren.

Gegeven is een functie $f(x)$, waarvoor geldt

$$f(x) = O(g_1(x)) , \quad \text{als } x \rightarrow x_0 .$$

Bij nadere bestudering blijkt er een konstante c_1 te bestaan, zodanig dat

$$f(x) \sim c_1 g_1(x) .$$

Bestudering van het verschil $f(x) - c_1 g_1(x)$ laat vervolgens zien

$$f(x) - c_1 g_1(x) \sim c_2 g_2(x) ,$$

met $g_2(x) = o(g_1(x))$.

Zetten we dit proces voort dan kan er een zogenaamde asymptotische ontwikkeling (of een asymptotische reeks) ontstaan voor $x \rightarrow x_0$:

$$f(x) \sim \sum_{i=1}^{\infty} c_i g_i(x) .$$

Hierbij moet voor $x \rightarrow x_0$ voldaan zijn aan

$$g_{i+1}(x) = o(g_i(x)), \quad i \geq 1$$

en

$$f(x) = \sum_{i=1}^n c_i g_i(x) + o(g_n(x)), \quad n \geq 0.$$

Dergelijke asymptotische ontwikkelingen kunnen soms dienst doen om in de buurt van x_0 funktiewaarden van f goed te benaderen, hoewel de ontwikkeling niet hoeft te konvergeren voor vaste x .

5.2. Formule van Stirling

Het komt dikwijls voor, dat men iets wil weten over het asymptotisch gedrag van uitdrukkingen van het type

$$\sum_{k=1}^n f(k) ,$$

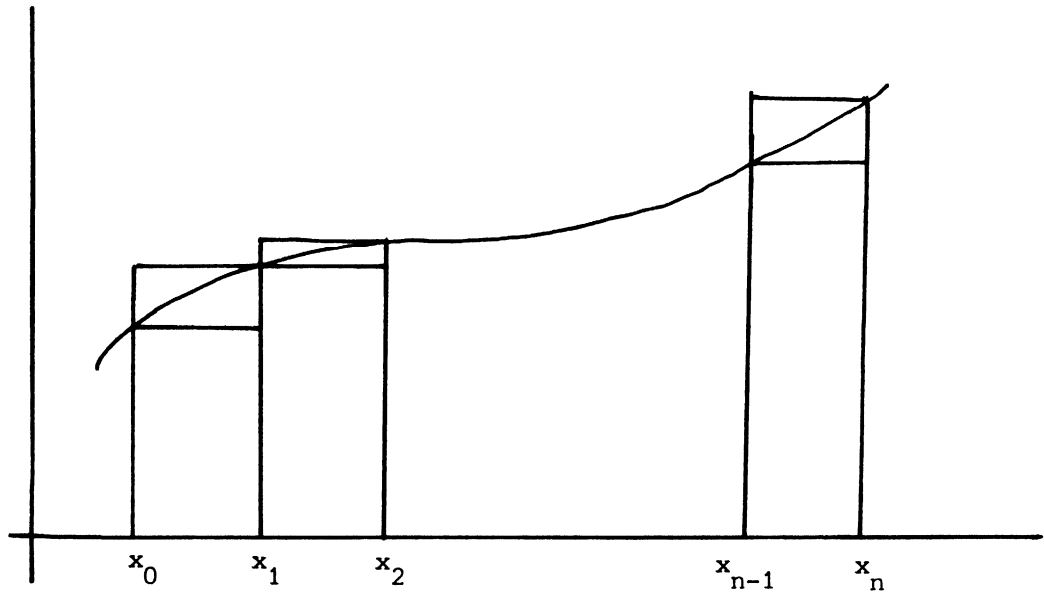
voor grote waarden van n .

Een meestal goed toepasbare techniek berust op de afschatting van $\int_{x_0}^{x_n} f(x) dx$ door middel van boven- en ondersommen.

Wanneer $f(x)$ een monotoon toenemende, integreerbare funktie op $[x_0, x_n]$ is geldt

$$\sum_{k=1}^n f(x_{k-1}) (x_k - x_{k-1}) \leq \int_{x_0}^{x_n} f(x) dx \leq \sum_{k=1}^n f(x_k) (x_k - x_{k-1}) . \quad (5.5)$$

Dit is onmiddellijk duidelijk aan de hand van figuur 5.1.
(Zie ook [4].)



Figuur 5.1

Voor $x_k = k+1$ ($0 \leq k \leq n$) wordt dit

$$\sum_{k=1}^n f(k) \leq \int_1^{n+1} f(x) dx \leq \sum_{k=1}^n f(k+1),$$

en dus

$$\int_1^{n+1} f(x) dx - f(n+1) + f(1) \leq \sum_{k=1}^n f(k) \leq \int_1^{n+1} f(x) dx. \quad (5.6)$$

Pas (5.5) toe op $f(k) = \ln k$. Dat levert ons

$$(n+1) \ln(n+1) - n - \ln(n+1) \leq \sum_{k=1}^n \ln k \leq (n+1) \ln(n+1) - n.$$

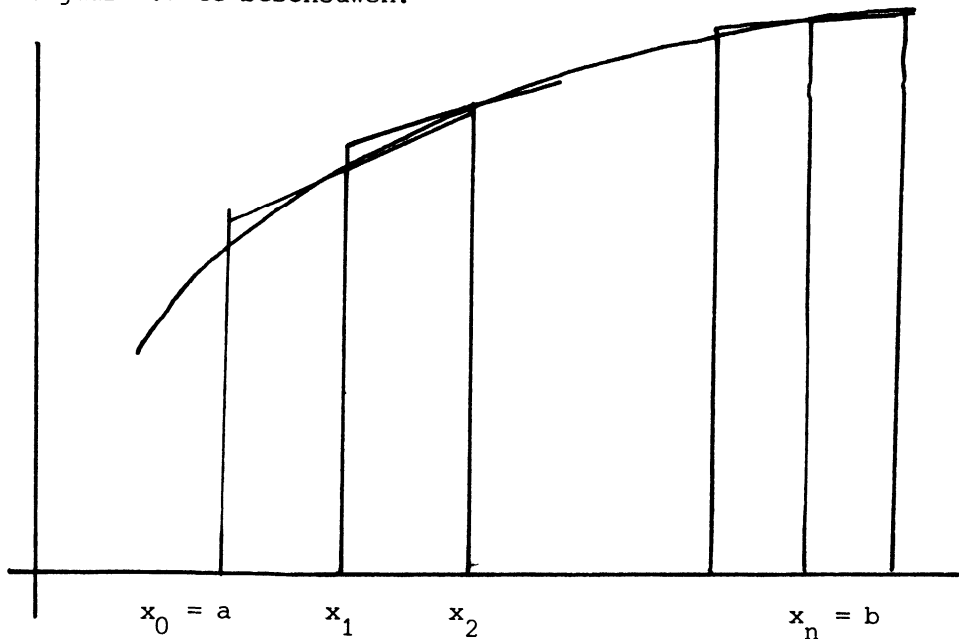
Hieruit volgt

$$\ln n! = (n+1) \ln(n+1) - n + O(\ln n). \quad (5.7)$$

We kunnen een betere asymptotische benadering voor $\ln n!$ afleiden. Daartoe gaan we uit van

$$\begin{aligned} \sum_{k=1}^n \frac{1}{2} (f(x_{k-1}) + f(x_k)) (x_k - x_{k-1}) &\leq \int_{x_0}^{x_n} f(x) dx \\ &\leq \sum_{k=1}^n \frac{1}{2} \left[\left(f(x_k) - \frac{f(x_{k+1}) - f(x_k)}{x_{k+1} - x_k} (x_k - x_{k-1}) \right) + f(x_k) \right] (x_k - x_{k-1}). \end{aligned} \quad (5.8)$$

Deze ongelijkheden gelden in het geval, dat f een monotoon toenemende integreerbare functie is, die op $[x_0, x_n]$ bol is. Men kan (5.8) afleiden door figuur 5.2 te beschouwen.



Figuur 5.2

De bovenste zijde van het trapezium met hoekpunten x_{k-1} en x_k is het verlengde van de verbindingslijn van de punten $(x_k, f(x_k))$ en $(x_{k+1}, f(x_{k+1}))$. De kortste opstaande zijde van dit trapezium is daarom gelijk aan

$$f(x_k) - \frac{f(x_{k+1}) - f(x_k)}{x_{k+1} - x_k} (x_k - x_{k-1}) .$$

Door van alle trapezia de oppervlakte te nemen ontstaat op eenvoudige wijze (5.8).

Substitutie van $x_k = k+1$ ($0 \leq k \leq n+1$) levert dan

$$\begin{aligned} \sum_{k=1}^n f(k) + \frac{1}{2}(f(n+1) - f(1)) &\leq \int_1^{n+1} f(x) dx \\ &\leq \sum_{k=1}^n f(k) + f(n+1) - f(1) - \frac{1}{2}(f(n+2) - f(2)) , \end{aligned}$$

waaruit tenslotte volgt

$$\begin{aligned} \int_1^{n+1} f(x) dx - f(n+1) + f(1) + \frac{1}{2}(f(n+2) - f(2)) &\leq \sum_{k=1}^n f(k) \\ &\leq \int_1^{n+1} f(x) dx - \frac{1}{2}(f(n+1) - f(1)) . \end{aligned} \quad (5.9)$$

We mogen (5.9) toepassen op $f(k) = \ln k$, omdat $f''(k) = -\frac{1}{k^2} < 0$ op $(0, \infty)$.
 Substitutie van $f(k) = \ln k$ en vervanging van n door $n-1$ geeft

$$\begin{aligned} n \ln n - n + 1 - \ln n + \frac{1}{2} \ln(n+1) - \frac{1}{2} \ln 2 &\leq \sum_{k=1}^{n-1} \ln k \\ &\leq n \ln n - n + 1 - \frac{1}{2} \ln n, \end{aligned}$$

en dus

$$\begin{aligned} n \ln n - n + \frac{1}{2} \ln(n+1) + 1 - \frac{1}{2} \ln 2 &\leq \sum_{k=1}^n \ln k \\ &\leq \left(n + \frac{1}{2}\right) \ln n - n + 1. \end{aligned}$$

Daar $\ln(n+1) - \ln n = \ln\left(1 + \frac{1}{n}\right) = O(1)$ vinden we tenslotte

$$\ln n! = \sum_{k=1}^n \ln k = \left(n + \frac{1}{2}\right) \ln n - n + O(1), \quad (5.10)$$

voor $n \rightarrow \infty$.

Door van (5.10) de e-macht te nemen krijgt men

$$n! \sim \sigma \frac{n^{n+\frac{1}{2}}}{e^n}. \quad (5.11)$$

De konstante σ laat zich nader bepalen en blijkt dan gelijk te zijn aan $\sqrt{2\pi}$, zodat we tenslotte krijgen

$$n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n. \quad (5.12)$$

Dit resultaat staat bekend als de formule van Stirling.

Een nauwkeuriger benadering krijgt men door de eerstvolgende term uit de asymptotische ontwikkeling mee te nemen:

$$n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \left(1 + \frac{1}{12n}\right). \quad (5.13)$$

De korrektieterm is berekend door Robbins.

Lijst van referenties

1. Reingold, E.M., Nievergelt, J. and Narsingh Deo, Combinatorial Algorithms, Prentice-Hall, Englewood Cliffs (N.J.), 1977.
2. Liu , E.L., Introduction to Combinatorial Mathematics, McGraw-Hill, New York, 1968.
3. Wilson, R.J., Introduction to Graph Theory, Oliver and Boyd, Edinburg, 1977.
4. Almering, J.H.J. e.a., Analyse, D.U.M., Delft, 1986.
5. Townsend, M., Discrete Mathematics: Applied Combinatorics and Graph Theory, The Benjamin/Cummings Publishing Company, Menlo Park (Cal.)
6. Amstel, J.J. v., e.a., Inleiding tot het Programmeren, dl. 1 en 2, Academic Service, Den Haag, 1982.
7. Baase, S., Computer Algorithms, Addison-Wesley, Reading (Mass.), 1978
8. Decnop, G.W., Iperen, H. v. en Martini, R., Lineaire Algebra, D.U.M., Delft, 1983.
9. Aho, A.V., Hopcroft, J.E. and Ullman, J.D., Data Structures and Algorithms, Addison-Wesley, Reading (Mass.), 1975.
10. Diktaat "Inleiding Informatica", T.U. Delft, 1987
11. Grimaldi, R.P., Discrete and Combinatorial Mathematics, Addison-Wesley, Reading (Mass.), 1984.

VRAAGSTUKKEN

Hoofdstuk 1

1. Iemand bezit 4 hoeden, 6 jasjes, 2 broeken, 3 overhemden, en 5 dassen.
Op hoeveel manieren kan hij zich kleden als hij van elk type kledingstuk precies één exemplaar moet dragen ?
Hoe groot wordt dit aantal als in het kledingvoorschrift het woord "precies" wordt vervangen door "hoogstens" ?
2. Hoeveel getallen van 9 cijfers zijn er te formeren door elk van de cijfers 1,2,...,9 precies één keer te gebruiken ?
Hoeveel van deze getallen zijn groter dan 500000000 ?
3. Hoeveel permutaties zijn er van de 26 letters van het alfabet, waarin de 5 klinkers op opéénvolgende plaatsen voorkomen ?
4. Bepaal de som van alle 4-cijferige getallen, die kunnen worden gevormd met 4 verschillende cijfers uit het alfabet { 1,3,5,7 }.
5. Hoeveel 8-cijferige getallen zijn er, waarin de cijferblokken 16,26,36 en 46 alle voorkomen ?
Wat is de som van deze getallen ?
6. Hoe groot is de som van alle 5-cijferige getallen bestaande uit de cijfers 1,1,1,3 en 4 ?
7. Hoeveel even 6-cijferige getallen kunnen met de cijfers 1,1,1,2,2 en 4 gemaakt worden ?
8. Men heeft de beschikking over n kralen die een verschillende kleur hebben. Iemand maakt van deze kralen een halsketting, zodanig dat de n kralen op gelijke afstand van elkaar zijn aangebracht. Hoeveel verschillende halskettingen kan hij maken als
 - a. de halskettingen niet omgedraaid mogen worden
 - b. de halskettingen wel omgedraaid mogen worden?
9. Op hoeveel onderscheidbare manieren kan men 2 rode, 2 witte, 2 blauwe en 2 groene ballen in een rij leggen?

10. Hoeveel rechthoeken (waarvan de zijden de velden niet snijden) kunnen worden gevormd op een schaakbord ?
Dezelfde vraag voor vierkanten.
11. Bepaal het aantal manieren, waarop 5 torens op een schaakbord kunnen worden geplaatst zonder dat ze elkaar bedreigen.
12. Zij A een verzameling met n elementen.
Bepaal het aantal afbeeldingen $A \rightarrow A$, die precies k elementen van A invariant laten.
13. Hoeveel Morse-tekens van lengte ≤ 4 kunnen worden gevormd over het alfabet $\{., -\}$?
14. In een tennispark verschijnen op zekere dag m dames en n heren.
Hoeveel spelen zijn er mogelijk waarbij 2 dames tegen 2 heren optreden ?
Hoeveel gemengd-dubbel-partijen zijn er mogelijk ?
15. In een parlement zijn 3 partijen vertegenwoordigd : 50 liberalen, 40 konservatieven en 30 socialisten.
Hoeveel kommissies kunnen worden gevormd met de verdeelsleutel $(5,3,2)$?
16. Op hoeveel manieren kunnen 3 verschillende getallen worden gekozen uit de getallen $1,2,\dots,300$ zodanig dat hun som deelbaar is door 3 ?

- 17a. Er doen $2n$ voetbalklubs mee aan een voetbaltoernooi. De eerste ronde bestaat uit n wedstrijden, die door loting worden vastgelegd. De wedstrijden worden eventueel verlengd totdat één der teams gewonnen heeft; gelijke spelen komen derhalve niet voor.
- Op hoeveel manieren kan de eerste ronde worden georganiseerd, als men onderscheid maakt tussen "uit" en "thuis" ?
- b. De winnaars van de eerste ronde gaan over naar de tweede ronde, die op dezelfde wijze wordt georganiseerd. De winnaars van de tweede ronde gaan weer verder, enz. Als men start met 2^m klubs (dus $n = 2^{m-1}$), op hoeveel manieren kan de organisatie van het hele toernooi dan plaatsvinden ?
- Geef een numerieke benadering van het antwoord in het geval $m = 4$ met behulp van de formule van Stirling.
18. Er worden 7 identieke dobbelstenen gegooid. Hoeveel uitkomsten zijn mogelijk ? Hoeveel uitkomsten zijn mogelijk met een even aantal zessen ?
19. Voor een gezelschap van 7 personen zijn 2 tweepersoons- en 1 driepersoonskamer gereserveerd. Op hoeveel manieren kan dit gezelschap over de kamers verdeeld worden als er twee personen in het gezelschap zijn die in geen geval in dezelfde kamer willen worden ondergebracht ?
20. Hoeveel verschillende termen bevat de ontwikkeling van $(a + b + c + d)^{10}$?
21. Een monieke veelterm in x van de graad $2n$ heeft $2n$ verschillende nulpunten. Op hoeveel manieren kan deze veelterm in monieke veeltermen van de graad 2 ontbonden worden, als de volgorde der factoren niet van belang is ?
22. Een chocoladereep bevat 8 "breekplaatsen". Op hoeveel manieren kan de reep gebroken worden ?
23. Wat is het grootste aantal snijpunten van 30 rechten in het platte vlak ? Hoe groot is dit aantal als a) 10 rechten onderling parallel zijn ;
b) 8 rechten onderling parallel zijn en 6 andere rechten door één punt gaan ?
24. Er zijn 4 verschillende postzegels van 50 cent in omloop. Op hoeveel manieren kan men een pakket met dergelijke zegels frankeren als de verschuldigde porto 250 cent is ?

25. Op hoeveel manieren kan men een verzameling met rn elementen verdelen in klassen met elk r elementen ?

26. Bepaal het totale aantal delers van $10!$ en ook van 10^r .

27. Bewijs de volgende betrekkingen :

a) $\sum_{r=0}^s \binom{n+r}{n} = \binom{n+s+1}{s}$;

b) $\sum_{k=1}^n k \binom{n}{k} = n \cdot 2^{n-1}$;

c) $\sum_{k=1}^n (-1)^k k \binom{n}{k} = 0$;

d) $\sum_{k=0}^n k \cdot k! = (n+1)! - 1$;

e) $\sum_{k=0}^n \frac{1}{k+1} \binom{n}{k} = \frac{2^{n+1}-1}{n+1}$.

28. a) Hoeveel 2-cijferige gehele getallen zijn er ?

(00,01,... zijn ook 2-cijferige getallen)

b) Hoeveel combinaties van 2 verschillende cijfers zijn er?

c) Hoeveel combinaties van 2 cijfers al of niet verschillend zijn er ?

d) Hoeveel getallen van 2 cijfers zijn er, waarbij de cijfers stijgend zijn gerangschikt ?

e) Hoeveel getallen van 2 cijfers zijn er, waarbij de cijfers niet dalend zijn gerangschikt ?

29. Iemand verdeelt 61 rode knikkers over 3 dozen A,B en C.

Op hoeveel manieren kan dat ?

Hoeveel manieren zijn mogelijk als elke doos minstens 20 knikkers moet bevatten ?

Zelfde vraag met 15 knikkers.

30. De studentenadministratie moet 61 informatica-studenten verdelen over 3 groepen In_1 , In_2 en In_3 .

Op hoeveel manieren kan dat?

Hoeveel manieren zijn mogelijk als elke groep minstens 20 studenten moet bevatten?

Zelfde vraag met 15 studenten.

Hoofdstuk 2

1. Zij a_n het aantal binaire rijen met lengte n waarin geen twee opeenvolgende nullen mogen voorkomen. Leid een recurrente betrekking van de tweede orde af voor a_n .
Los deze vergelijking op met behulp van de randvoorwaarden voor a_1 en a_2 .
2. Gegeven is een rechthoekig bord van 2 bij n , alsmede een onbeperkt aantal (dubbelblanke) dominostenen van 2 bij 1. Het aantal manieren, waarop men het bord met dominostenen kan beleggen noemt men t_n .
Leid een expliciete uitdrukking af voor t_n .
3. Een ternaire rij is een rij van getallen, die gekozen worden uit de verzameling $\{0, 1, 2\}$. Men beschouwt ternaire rijen ter lengte n , waarbij geen naburige enen optreden. Hun aantal noemt men a_n ($n \geq 1$).
Leid een recurrente betrekking af voor a_n .
Bepaal a_0 zodanig, dat de betrekking geldig is voor $n \geq 0$.
Los de betrekking op.
4. Los op de recurrente betrekking
$$a_n = 2a_{n-1} + 3a_{n-2} + 3^n, \quad a_0 = 1, \quad a_1 = 2.$$
5. Zij a_n het aantal cijferrijen van lengte n bestaande uit cijfers van de verzameling $\{0, 1, 2, 3\}$, waarin een oneven aantal nullen voorkomt.
Bewijs dat geldt
$$\begin{cases} a_{n+1} = 2a_n + 4^n, & n \geq 1, \\ a_1 = 1. \end{cases}$$

Los bovenstaande recurrente betrekking op en vergelijk het antwoord met het resultaat van voorbeeld 14.
6. In het platte vlak zijn n rechten getekend, die elkaar twee aan twee snijden, terwijl er geen drie door één punt gaan. Deze rechten verdelen het platte vlak in a_n gebieden.
Bepaal een recurrente betrekking voor a_n en los deze op.
7. In de ruimte zijn n vlakken gegeven, zo dat elk drietal vlakken juist één punt gemeen heeft, terwijl er geen viertal vlakken is met een gemeenschappelijk punt. Deze vlakken verdelen de ruimte in d_n gebieden.
Toon aan dat
$$d_{n+1} = d_n + a_n,$$
waarbij a_n het in vraagstuk 6 gedefinieerde aantal is. Bepaal d_n .

8. Los de recurrente betrekkingen van de vraagstukken 3,4 en 6 op met behulp van genererende funkties.

9. Gegeven is $GF(x) = (1 + x^3 + x^5)^{10} = \sum_{k=0}^{\infty} a_k x^k$.
Bepaal a_{11} , a_{12} , a_{14} en a_{30} .

10. Bepaal b_n in
 $(1 + x + x^2 + \dots + x^{q-1})^2 = \sum_{n=0}^{2q-2} b_n x^n$, ($q \in \mathbb{N}$).

11. a. Vervang in de betrekking

$$(1+x)^r (1+x)^s = (1+x)^{r+s}$$

alle machten van $1+x$ door hun binomiaalontwikkelingen. Welke relatie volgt hieruit voor binomiaalkoefficienten?

b. Dezelfde vraag voor de betrekking

$$\frac{1}{(1-x)^r} \frac{1}{(1-x)^s} = \frac{1}{(1-x)^{r+s}}.$$

Beschouw in het bijzonder het geval $s=1$.

12. a. Bepaal a_0 t.m. a_3 in

$$\frac{x-3}{x^2-3x+2} = \sum_{n=0}^{\infty} a_n x^n.$$

b. Bepaal ook a_{50} .

13. In een Senaat zijn 50 staten vertegenwoordigd, elk door 3 senatoren.

Men stelt een kommissie samen door van elke staat nul, één of twee senatoren te kiezen. Op hoeveel manieren kan men deze kommissie samenstellen als het totale aantal leden 50 bedraagt en men

a) geen onderscheid maakt tussen twee senatoren uit dezelfde staat;

b) wel onderscheid maakt tussen twee senatoren uit dezelfde staat ?

Bereken in de gevallen a en b ook het aantal mogelijke kommissies als het aantal leden willekeurig groot mag zijn.

14. Op hoeveel manieren kan men 200 identieke stoelen verdelen over 4 konferentiezalen, als elke zaal 20, 40, 60, 80 of 100 stoelen mag bevatten ?

15. Hoeveel verschillende monotone woorden van lengte $3n$ kan men maken als men de beschikking heeft over $2n$ A's $2n$ B's en $2n$ C's ?

16. Zij a_n het aantal verschillende monotone woorden van lengte n over het alfabet $\{0,1,2\}$ met een even aantal nullen. Bepaal een genererende funktie voor a_n . Bepaal a_n door "breuksplitsing" toe te passen op deze funktie.

- 17.a. Zij a_r het aantal manieren waarop het ogentotaal r kan zijn bij het werpen van 2 verschillend gekleurde dobbelstenen.

Bepaal de gewone genererende funktie van de rij $a_0, a_1, a_2, \dots$.

- b. Zij b_r het aantal manieren, waarop het ogentotaal r kan worden verkregen bij twee keer achterelkaar gooien van dezelfde dobbelsteen. (Verschillende volgordes worden als verschillende manieren gerekend.)

Bepaal de gewone genererende funktie van de rij $b_0, b_1, b_2, \dots$.

- c. Zij c_r het aantal manieren, waarop het ogentotaal r kan worden verkregen bij herhaald werpen van dezelfde dobbelsteen; ongeacht het aantal keren dat men gooit. Bepaal de gewone genererende funktie van de rij $c_0, c_1, c_2, \dots$.

18. Een kompositie van een natuurlijk getal r in k delen is een verdeling van r in delen $f_1, f_2, \dots, f_k$, zodanig dat

$$\begin{cases} f_1 + f_2 + \dots + f_k = r, \\ f_i \geq 0, \quad 1 \leq i \leq k. \end{cases}$$

Zo luiden bijvoorbeeld alle komposities van 3 in 3 delen :

003, 030, 300, 012, 102, 120, 201, 210, 021, 111.

- a. Toon met behulp van stelling (2.1) aan, dat het aantal komposities van r in k delen gelijk is aan

$$\binom{k+r-1}{r}.$$

- b. Bepaal op dezelfde wijze het aantal komposities van r in k positieve delen en ook in een willekeurig aantal positieve delen.

- c. Leid de genererende funkties af voor het aantal komposities van r in een willekeurig aantal positieve delen, zodanig dat alle delen kleiner dan of gelijk zijn aan een vast getal s .

Bepaal hiermee het totale aantal komposities van 25 in 4 positieve delen, die alle niet groter zijn dan 10.

- d. Pas het hierboven afgeleide toe op vraagstuk 29 van hoofdstuk 1.

19. Beschouw binaire rijen van lengte n . Onder een patroon zullen we verstaan één of meer opéénvolgende nullen en enen. We zeggen dat een patroon optreedt op de k -de plaats als het laatste bit van het patroon op de k -de plaats staat. Nadat een patroon in zijn geheel is opgetreden kan men verder zoeken naar een volgend optreden van dat patroon in de rij.

Zo treedt het patroon 010 twee keer op in de rij 110101010101, en wel op de 5-de en de 9-de plaats, (en niet op de 7-de en 11-de plaats).

Zij a_n het aantal binaire rijen van lengte n , waarin het patroon 010 optreedt op de n -de plaats. Leid af, dat geldt

$$a_n + a_{n-2} = 2^{n-3},$$

en los deze vergelijking, met geschikte randvoorwaarden, op.

Leid ook een genererende funktie af voor a_n .

20. Zij b_n ($n \geq 3$) het aantal binaire rijen van lengte n , waarin het patroon 010 voor het eerst optreedt op de n -de plaats en niet daarvoor. Leid voor b_n de volgende recurrente betrekking af

$$b_n + b_{n-2} + b_{n-3} 2^0 + b_{n-4} 2^1 + \dots + b_3 2^{n-6} = 2^{n-3}, \quad n \geq 6.$$

Bepaal een genererende functie $B(x)$ voor de getallen b_n door bovenstaande betrekking als een bijzonder geval van konvolutie te beschouwen.

21. Zij a_n het aantal binaire rijen, zoals in vraagstuk 19 is vastgelegd en b_n zoals in vraagstuk 20 is gedefinieerd.

Toon aan, dat

$$a_n = a_n b_0 + a_{n-1} b_1 + \dots + a_0 b_n, \quad n \geq 3,$$

voor geschikte keuze van $a_0, a_1, a_2, b_0, b_1, b_2$.

Bepaal met deze betrekking opnieuw een genererende functie voor de b_n .

22. Leid een uitdrukking af voor de som van de 3^{de} machten van de eerste n positieve natuurlijke getallen

$$1^3 + 2^3 + \dots + n^3.$$

23. Een partitie van een natuurlijk getal r in k delen is een verdeling van r in g_i delen $i, i \in \{1, 2, \dots, r\}$, met

$$\begin{cases} g_1 + g_2 + \dots + g_r = k. \\ g_i \in \mathbb{IN}, \quad 1 \leq i \leq r. \end{cases}$$

Hieruit volgt dus dat

$$1 \cdot g_1 + 2 \cdot g_2 + \dots + r \cdot g_r = r.$$

Zo luiden bijvoorbeeld alle partities van 4 in 2 delen 13, 22 (de volgorde der delen is niet van belang), terwijl 112 de enige partitie van 4 is in 3 delen.

- a. Toon met behulp van stelling (2.1) aan, dat het totale aantal partities van r wordt gegenereerd door de functie

$$GF(x) = \prod_{i=1}^{\infty} (1 - x^i)^{-1}.$$

Bereken hiermee het totale aantal partities van 4.

- b. Toon aan, dat het totale aantal partities van r in verschillende delen wordt gegenereerd door

$$GF(x) = \prod_{i=1}^{\infty} (1 + x^i).$$

- c. Bepaal een genererende functie voor het aantal partities van r in oneven delen en toon met behulp van b aan, dat dit aantal gelijk is aan het totale aantal partities van r in verschillende delen (stelling van Euler).

- d. Bepaal een genererende functie voor het aantal partities van r , zodanig dat de delen niet groter zijn dan k .

24. We geven een partitie weer door een diagram, dat bestaat uit rijen punten, zodanig dat elke rij correspondeert met een deel van de partitie. Het aantal punten in een rij is gelijk aan de waarde van het corresponderende deel. De rijen worden zo gerangschikt, dat elke rij steeds minstens evenveel punten bevat als de rij eronder. Als voorbeeld geven we het diagram

```

. . . . .
. . . .
. . . .
. .

```

dat de partitie 5 4 4 2 van 15 voorstelt.

Zij nu gegeven een partitie van een natuurlijk getal r in hoogstens k delen.

- a. Maak aannemelijk dat door het verwisselen van rijen met kolommen in zo'n (Ferrari) diagram er opnieuw een diagram ontstaat van een partitie, met de eigenschap, dat de delen niet groter dan k zijn.
- b. Bepaal nu met behulp van 23d een genererende functie voor het aantal partities van r in precies k delen.

25. Bepaal het antwoord van vraagstuk 5 (van dit hoofdstuk) met behulp van stelling (2.2).

26. a. Bepaal met behulp van stelling (2.2) het aantal manieren om r verschillende objecten over n verschillende, niet geordende, cellen te verdelen, zodanig dat elke cel minstens een object bevat.

Vergelijk het resultaat met voorbeeld 15.

- b. Beschouw nu verdelingen van r verschillende objecten over n niet-verschillende cellen (lege cellen zijn toegestaan). Laat met behulp van a zien dat het totale aantal van zulke verdelingen gelijk is aan

$$S(r,1) + S(r,2) + \dots + S(r,n), \text{ als } r \geq n,$$

en

$$S(r,1) + S(r,2) + \dots + S(r,r), \text{ als } r \leq n.$$

($S(r,n)$ is het getal van Stirling van de tweede soort).

- c. Bewijs dat in het geval $r \leq n$ het aantal verdelingen van b gegenereerd wordt door de functie

$$e^{e^x - 1} - 1.$$

Hoofdstuk 3

1. Gegeven is de relatie

$$a_1 \cdot 1! + a_2 \cdot 2! + \dots + a_m \cdot m! = 0,$$

met $|a_i| \leq i$, $1 \leq i \leq m$. Laat r de kleinste index zijn met $a_r \neq 0$. Leid een tegenspraak af door de gegeven relatie te delen door een geschikt getal.

Geef hiermee een alternatief bewijs van stelling (3.1).

2.a. Bepaal de representatie van 10^6 in het fakulteitsstelsel.

b. Welk getal wordt in dat stelsel gerepresenteerd door $(0,1,2,3,4,5,6,7,8,9)$?

3. Toon aan, dat de kompleksiteit van algoritme 3.1 asymptotisch van de orde $O(\ln n)$ is,

(i) als $n = N!$, voor zekeren $N \in \mathbb{N}$

(ii) als n een willekeurig natuurlijk getal is.

4. Bepaal de index van de 9-permutatie

$$5 \quad 2 \quad 7 \quad 1 \quad 9 \quad 8 \quad 3 \quad 4 \quad 6,$$

zowel in de lexicografische ordening als in de minimal-change ordening.

5. Wat zijn de opvolgers van de permutatie uit vraagstuk 4 in beide ordeningen?

6. Bepaal ook de voorgangers van de permutatie uit vraagstuk 4 in de beide ordeningen.

Volgens welke regels zou men die in het algemeen kunnen vinden?

7. Bepaal de honderdduizendste 12-permutatie in de beide ordeningen.

8.a. Bepaal het aantal 7-permutaties die in de lexicografische ordening nog volgen op de permutatie

$$\pi := 5 \quad 1 \quad 3 \quad 7 \quad 6 \quad 2 \quad 4$$

Geef ook de twee permutaties, die onmiddellijk op π volgen.

b. Hoeveel 7-permutaties zijn er, waarvan het eerste cijfer groter is dan 3 en het laatste cijfer kleiner dan 5?

9. Men spreekt van een transpositie als twee permutaties van n getallen in elkaar overgaan door het verwisselen van slechts twee (niet noodzakelijk naburige) getallen. Als een permutatie te verkrijgen is uit de permutatie $1 \ 2 \ \dots \ n$ door een even aantal transposities toe te passen spreekt men van een even permutatie en anders van een oneven permutatie. Men kan bewijzen dat het even of oneven zijn onafhankelijk is van welke transposities er precies zijn toegepast.

Zij nu $\text{ind}_n(\pi)$ het rangnummer van π in de lexicografische ordening. We schrijven dit getal in het fakulteitsstelsel als $(d_0, d_1, \dots, d_{n-1})$. $\sum_{j=0}^{n-1} d_j$ even is. Toon aan, dat π een even permutatie is dan en slechts dan als $\sum_{j=0}^{n-1} d_j$ even is.

- 10.a Toon aan dat de binair gespiegelde Gray kode ook wordt vastgelegd door de definitie

$$G(1) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$G(n+1) = \begin{pmatrix} G_0 & 0 \\ G_0 & 1 \\ G_1 & 1 \\ G_1 & 0 \\ \cdot \\ \cdot \\ \cdot \end{pmatrix}$$

waarin $G_0, G_1, \dots$ de woorden van $G(n)$ zijn.

- b. Zij $T_n = t_0, t_1, \dots, t_{2n-2}$ de overgangsrij van de binair gespiegelde Gray kode $G(n)$ voor zekere $n \in \mathbb{Z}^+$.

Bewijs dat T_{n+1} de gedaante heeft

$$T_{n+1} = 1, t_0 + 1, 1, t_1 + 1, 1, t_2 + 1, \dots, 1, t_{2n-2} + 1, 1.$$

11a. Laat $B_0, B_1, \dots, B_{2^n-1}$ de binaire voorstellingen zijn van de getallen $0, 1, \dots, 2^n-1$.

Toon aan dat

$$\begin{pmatrix} B_0 & 0 \\ B_0 & 1 \\ B_1 & 0 \\ B_1 & 1 \\ \vdots & \vdots \\ B_{2^n-1} & 0 \\ B_{2^n-1} & 1 \end{pmatrix}$$

de rij binaire representaties is van $0, 1, \dots, 2^{n+1}-1$.

b. In de binaire optelling

$$\begin{array}{r} 1 \quad 0 \quad 1 \quad 1 \\ + \quad \quad \quad 1 \\ \hline 1 \quad 1 \quad 0 \quad 0 \end{array}$$

vindt er een overdracht (carry) plaats van "de 1" van rechts naar links. Deze overdracht stopt bij het 3^{de} bit van rechts.

Laat nu $K_n = k_0, k_1, \dots, k_{2^n-2}$ de rij zijn, zodanig dat k_i het bit is waar de overdracht stopt, als we 1 optellen bij B_i (zie a). Hierbij worden de bits genummerd van rechts naar links van 1 tot en met n .

Bewijs dat $K_n = T_n$ (vraagstuk 10), door aan te tonen dat

$$K_{n+1} = 1, k_0+1, 1, k_1+1, 1, \dots, k_{2^n-2}+1, 1$$

12. Uit het vorige vraagstuk kunnen we het volgende konkluderen.

Zij

$$i = b_{n+1} b_n \dots b_1$$

de binaire representatie van het getal i , met $0 \leq i < 2^n$ (dus $b_{n+1} = 0$).

Als we bij i het getal 1 optellen en de overdracht stopt bij b_k , $1 \leq k \leq n$, dan moet in het i ^{de} codewoord van $G(n)$ ($0 \leq i \leq 2^n-1$):

$$G_i = g_n g_{n-1} \dots g_1$$

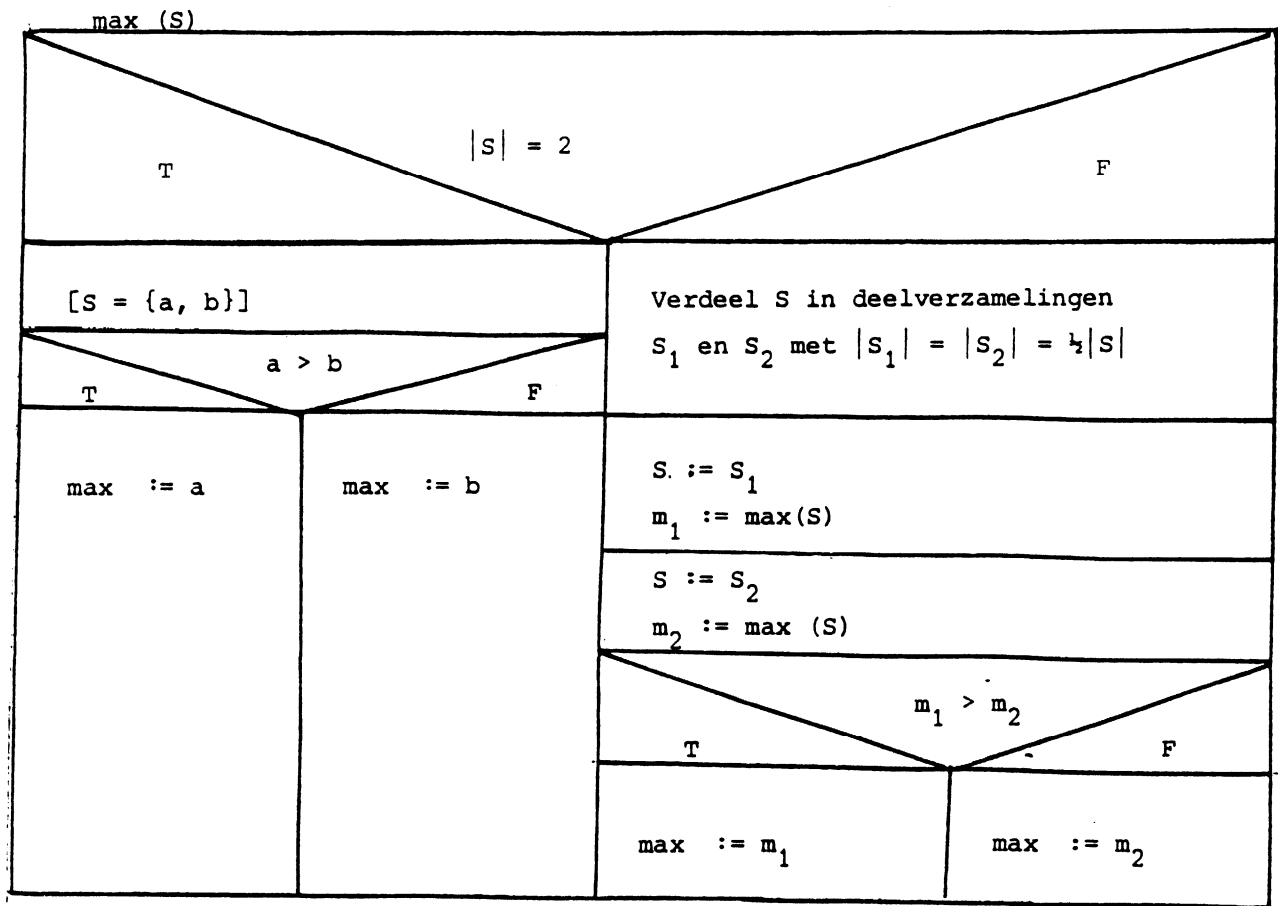
de waarde van g_k veranderd worden, teneinde G_{i+1} te krijgen (ofwel $t_i = k$).

Bewijs nu door inductie naar i toe te passen

$$g_j = b_{j+1} + b_j \pmod{2}, \quad 1 \leq j \leq n.$$

Met dit resultaat kan men bij een gegeven waarde van i op snelle wijze het

13. We definiëren een n -digit ternaire Gray kode als een rij van 3^n woorden van lengte n over het alfabet $\{0,1,2\}$, zodanig dat opeenvolgende kodewoorden slechts op één plaats verschillen, terwijl het verschil in digitwaarde 1 is. Geef een voorbeeld van een dergelijke kode met behulp van een rekursieve definitie.
14. Hoeveel 3-kombinaties uit de getallen $1,2,\dots,7$ liggen er tussen 1 4 6 en 3 4 7 als men deze combinaties lexicografisch ordent?
15. Gegeven is de volgende rekursieve functieprocedure



De invoerparameter S stelt een verzameling van 2^n reële, verschillende getallen voor. De waarde van $\max$ wordt gelijk aan het maximum van S . Als basisbewerkingen beschouwen we de vergelijkingen $a > b$ en $m_1 > m_2$. Zij a_n het aantal basisbewerkingen, dat wordt uitgevoerd voor S met $|S| = 2^n$.

Leid een recurrente betrekking af voor a_n en los die op.

16. Bepaal het totale aantal bitsveranderingen dat optreedt als men met behulp van algoritme 3.4 een lijst van alle bitrijen ter lengte n produceert . (Maak gebruik van het resultaat van opgave 11b).
Vergelijk het antwoord met dat voor een Gray kode.
17. Beschouw alle woorden van lengte 3 over het latijnse alfabet $\{a,b,\dots,z\}$, die strikt $\grave{e}$ n monotoon zijn.
Hoeveel van dergelijke woorden gaan in de lexikografische ordening vooraf aan het woord 'dik'?
18. Leid met behulp van het resultaat van vraagstuk 12 een regel af om het rangnummer i te bepalen van een gegeven kodewoord $G_1(n)$ van de n -bits Gray kode $G(n)$.
(Dit is stelling (3.8).)

Hoofdstuk 4

1. Bewijs dat een graaf bipartite is dan en slechts dan als alle circuits even lengte hebben.
2. De matrix

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

is de nabuurmatrijs van de gerichte graaf G.

Toon aan, dat er in G tussen elk tweetal verschillende knopen minstens één gerichte weg van de lengte 3 bestaat.

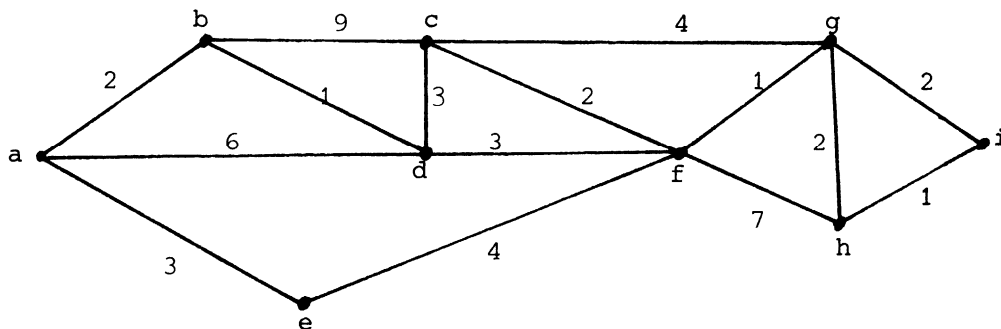
3. Ga na of de graaf uit vraagstuk 2 een Eulerpad bezit .
4. Zij A de nabuurmatrijs van een enkelvoudige ongerichte graaf G.
We definiëren een driehoek in G in als een drietal paarsgewijze verbonden knopen. Als t het totale aantal driehoeken in G voorstelt en als S het spoor is van A^3 ($S = \text{Tr}(A^3)$), geef dan het verband tussen t en S.
5. Toon aan dat een enkelvoudige graaf met n knopen samenhangend is als hij meer dan $\frac{1}{2}(n-1)(n-2)$ takken bezit.
6. Als een enkelvoudige, samenhangende, ongerichte graaf precies k knopen van oneven graad bezit ($k > 0$), dan zijn er $\frac{1}{2}k$ paden zodanig, dat elke tak van de graaf precies één keer voorkomt in één van deze paden.
- 7.a. De takken van de volledige graaf K_6 worden rood of blauw gekleurd. Toon aan dat hoe men de kleuring ook uitvoert er altijd wel een driehoek is aan te wijzen waarvan de zijden alle dezelfde kleur hebben.
- b. Bewijs dat in een groep van 6 mensen altijd wel één van de volgende situaties zich voordoet:
 - (i) er zijn 3 personen die elkaar wederzijds kennen;
 - (ii) er zijn 3 personen die elkaar wederzijds niet kennen.

8. Gegeven is een (enkelvoudige) samenhangende graaf G . In 4.1.2 werd ingevoerd het begrip afstand tussen twee knopen v en w , zijnde de lengte van het kortste pad tussen v en w .

We geven deze lengte aan met de notatie $d(v,w)$. Bewijs dat deze op $VG \times VG$ gedefinieerde functie inderdaad te interpreteren is als een afstand, door na te gaan dat

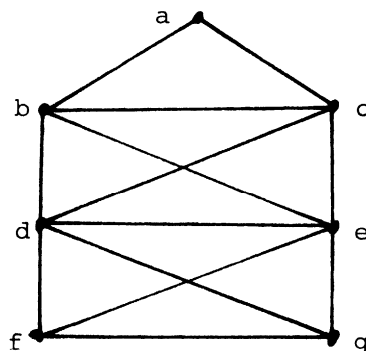
- (i) $d(a,b) = d(b,a)$, $\forall a, b \in VG$;
- (ii) $d(a,b) \geq 0$, $d(a,b) = 0 \leftrightarrow a = b$;
- (iii) $d(a,b) + d(b,c) \geq d(a,c)$.

9. Gegeven is de takgelabelde graaf G



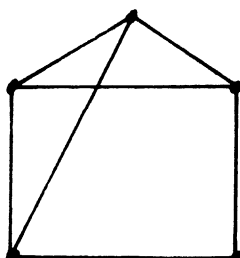
Konstrueer een minimaalskelet met de "greedy" algoritme en ook met de naaste-buur-algoritme.

10. Schrijf van de graaf G



de nabuurstructuur op, waarbij de knopen steeds alfabetisch zijn geordend. Bepaal met behulp van deze nabuurstructuur een opspannende boom van G met de DFS-techniek. Doe hetzelfde met de BFS-techniek.

11. Bepaal het aantal opspannende bomen van de graaf



12a. Ga na, dat de uitdrukking $\det(B_0 B_0^t)$ voor het totale aantal opspannende bomen ook nog goed is als G parallelle takken bezit.

b. Bewijs dat in het geval van een ongerichte graaf, die parallelle takken mag bevatten (maar geen lussen), geldt:

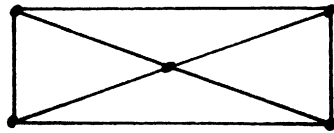
$$(i) (B_0 B_0^t)_{ij} = -(\text{aantal takken tussen } v_i \text{ en } v_j), \text{ als } i \neq j \text{ is};$$

$$(ii) (B_0 B_0^t)_{ii} = \rho(v_i).$$

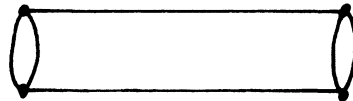
c. Men kan de nabuurmatrijs A uit 4.1.4 generaliseren door te definiëren, dat a_{ij} gelijk is aan het aantal takken van i naar j .

Druk nu het aantal opspannende bomen van de graaf van b uit in A en in de matrix $\text{diag}(\rho(V))$, die op de diagonaal de getallen $\rho(v_1), \rho(v_2), \dots, \rho(v_n)$ heeft staan en verder overal nullen.

13. Gebruik het resultaat van vraagstuk 12, om het totale aantal opspannende bomen te bepalen van de graaf

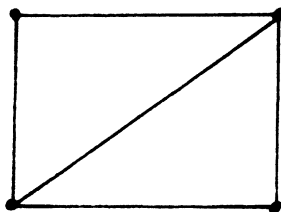


Doe hetzelfde voor de niet-enkelvoudige graaf



Kontroleer in het laatste geval het antwoord door alle opspannende bomen te tekenen.

14. Konstrueer met behulp van de kontraktietechniek alle opspannende bomen van de graaf



15a. Zij G een enkelvoudige graaf.

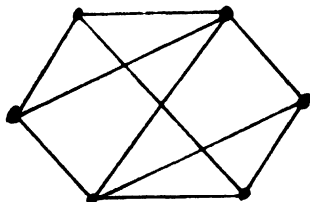
Ga met (4.6) na, dat

$$B^{(1)} = A.$$

b. Bewijs (4.6) met volledige inductie naar k .

16. Konstrueer met de methode uit 4.4.1 het kortste pad van de meest linkse naar de meest rechtse knoop van de graaf uit vraagstuk 9.

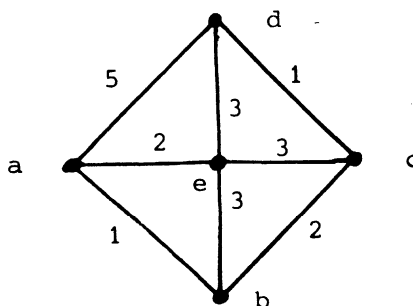
17. Gegeven is de graaf



Hoeveel elementen bevat de verzameling bestaande uit enkelvoudige circuits en takdisjunkte verenigingen van zulke circuits?
Bepaal een fundamenteelstelsel van sneden.

18. Zij G een ongerichte, samenhangende, enkelvoudige graaf.
Toon aan dat een willekeurig enkelvoudig circuit en een willekeurige snede van G altijd een even aantal takken gemeen hebben.

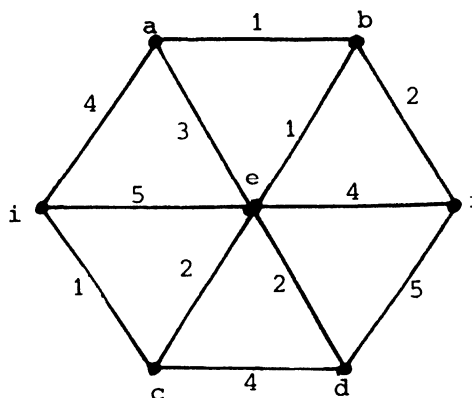
19. Gegeven is de takgelabelde graaf G :



Hoe groot is de diameter van G ?

Konstrueer een afstandstabel met formule (4.16).

20. Gegeven is de volgende gelabelde graaf :



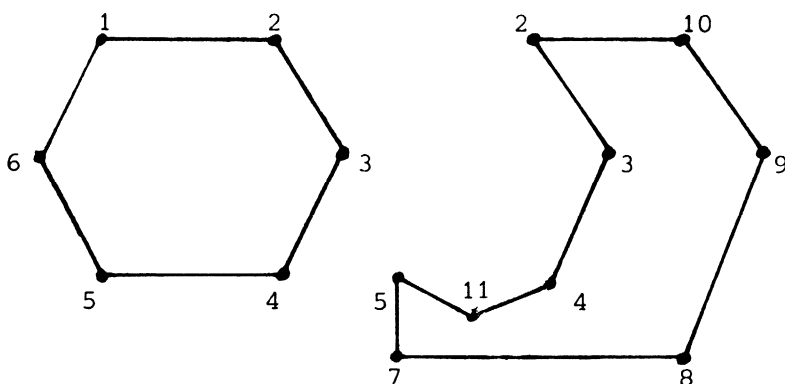
Bepaal met de kortste-pad-algoritme de afstand tussen i en f .

Geef een duidelijk overzicht van de labelwaarden die de knopen tijdens de uitvoering van de algoritme krijgen.

21. Geef de definitie van de volledige bipartite graaf (ongericht) $K_{n,m}$.
 Leid een uitdrukking af voor het totale aantal opspannende bomen van $K_{n,2}$.
 Aanwijzing: Gebruik bij voorkeur niet de stelling van Kirchhoff, maar ga na welke gedaanten de opspannende bomen kunnen hebben.
22. We definiëren een k -hoek in een ongerichte graaf als een circuit met k verschillende knopen en k verschillende takken.
- Hoeveel verschillende driehoeken ($k = 3$) bevat de volledig graaf K_n , als de omlooprichting in een driehoek er niet toe doet?
 - Dezelfde vraag voor het aantal verschillende vierhoeken in K_n ($k = 4$).

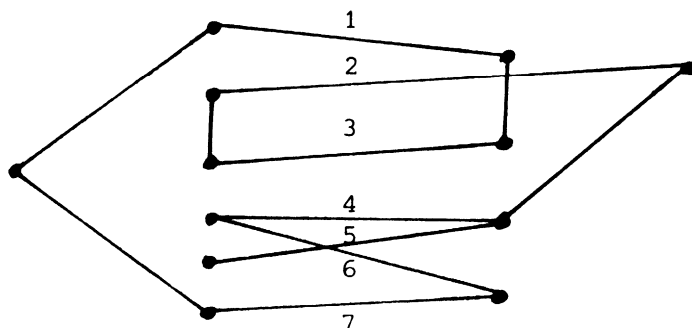
Aanwijzing : Men hoeft geen gebruik te maken van de nabuurmatrijs A .

23. Gegeven zijn de volgende circuits C_1 en C_2 van een graaf G :



Pas de algoritme beschreven in het bewijs van stelling (4.16) toe om $C_1 \oplus C_2$ te schrijven als de vereniging van takdisjunkte elementaire circuits.

24. Gegeven is de volgende graaf G :



Laat zien dat de scheidende verzameling takken $\{1,2,3,4,5,6,7\}$ te schrijven is als een vereniging van takdisjunkte sneden (Pas de algoritme uit het bewijs van stelling (4.21), deel (ii) toe.)

25. Zij G een ongerichte samenhangende, enkelvoudige graaf, met n knopen, waarvan de takken gelabeld zijn met (positieve) getallen $l(e)$, $e \in EG$. Gegeven is verder dat al deze gewichten verschillend zijn.

Zij t_i de lichtste tak incident met knoop v_i , dus

$$l(t_i) = \min\{l(e) \mid e \text{ incident met } v_i\}.$$

De takkenverzameling F is de volledige verzameling van "lichtste" takken:

$$F := \{t_i \mid 1 \leq i \leq n\}.$$

Bewijs dat G precies één minimaalskelet heeft door achtereenvolgens aan te tonen :

- (i) een minimaalskelet bevat alle takken van F ;
- (ii) F bevat geen deelverzameling takken, die een circuit vormen;
- (iii) uitgaande van de takken van F kan men op slechts één manier een minimaalskelet van G konstrueren.

26. Gegeven is een ongerichte enkelvoudige samenhangende graaf G .

Er is gegeven dat G planair is, dat wil zeggen dat G zo is af te beelden dat de takken elkaar slechts "snijden" in de knopen.

Een gebied is een deel van het platte vlak dat begrensd is door takken en dat in het inwendige géén takken of knopen bevat.

De contour van een gebied is de verzameling takken die dat gebied begrenzen.

Het aantal knopen van G is n , het aantal takken m en het aantal gebieden r .

- (i) Toon aan dat de contouren van de eindige gebieden een basis van $\mathcal{C}$ vormen.
- (ii) Bewijs dat voor elke planaire, samenhangende graaf (enkelvoudig of niet enkelvoudig) geldt $n - m + r = 2$. (relatie van Euler)

27. Leid met behulp van stelling (4.22) de uitdrukking af voor het aantal elementen van S .

Doe hetzelfde met behulp van stelling (4.21).

28. Hoe groot is het totale aantal enkelvoudige circuits in K_6 ?

ANTWOORDEN

Hoofdstuk 1

1. 720 ; 2520
 2. $9!$; $5 \cdot 8!$
 3. $(5!) \cdot (22!)$
 4. 106656
 5. $4!$; 751.515.144
 6. 444440
 7. 30
 - 8a. $(n-1)!$
 - b. $\frac{1}{2}(n-1)!$
 9. 2520
 10. 1296 ; 204
 11. $\frac{(8!)^2}{(3!)^2 \cdot 5!}$
 12. $\binom{n}{k} (n-1)^{n-k}$
 13. 30
 - 14a. $\binom{m}{2} \binom{n}{2}$
 - b. $2 \binom{m}{2} \binom{n}{2}$
 15. 9.106.006.728.000 ; 6.650.280.000
 16. 1.485.100
 - 17a. $\frac{(2n)!}{n!}$
 - b. $(2^m)!$
 18. 792 ; 496
 19. 160
 20. 286
 21. $\frac{(2n)!}{n! 2^n}$
 22. 255
 23. 435 ; 390 ; 393
 24. 56
 25. $\frac{(rn)!}{n! (r!)^n}$
 26. 270 ; $(r+1)^2$
 28. a. 100
 - b. 45
 - c. 55
 - d. 45
 - e. 55
 29. 1953 ; 3 ; 153
 30. 3^{61} ; $3 \binom{61}{21} \binom{40}{20}$;
- $$\sum_{k_1+k_2+k_3=61} k_1! k_2! k_3! = \frac{61!}{k_1! k_2! k_3!}.$$

Hoofdstuk 2

$$1. a_n = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^{n+2} - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2} \right)^{n+2}$$

$$2. t_n = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^{n+1} - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2} \right)^{n+1}$$

$$3. a_n = \frac{3+2\sqrt{3}}{6} (1+\sqrt{3})^n + \frac{3-2\sqrt{3}}{6} (1-\sqrt{3})^n$$

$$4. a_n = \frac{1}{16} [(12n+3)3^n + 13(-1)^n]$$

$$5. a_n = \frac{1}{2} (4^n - 2^n)$$

$$6. a_n = \frac{1}{2} (n^2 + n + 2)$$

$$7. d_n = \frac{1}{6} (n^3 + 5n + 6)$$

$$9. a_{11} = 360, a_{12} = 210, a_{14} = 840, a_{30} = 2731$$

$$10. b_n = \begin{cases} n+1, & 0 \leq n \leq q-1, \\ 2q-n-1, & n \geq q \end{cases}$$

$$11.a. \sum_{k=0}^n \binom{r}{k} \binom{s}{n-k} = \binom{r+s}{n}$$

$$b. \sum_{k=0}^n \binom{r+k-1}{r-1} \binom{s+n-k-1}{s-1} = \binom{r+s+n-1}{r+s-1}$$

$$12.a. a_0 = -\frac{3}{2}, a_1 = -\frac{7}{4}, a_2 = -\frac{15}{8}, a_3 = -\frac{31}{16}$$

$$b. a_{50} = 2^{-51} - 2$$

$$13.a. \sum_k \binom{50}{k} \binom{50-k}{k}; 3^{50}$$

$$b. \sum_k \binom{50}{k} \binom{k}{50-k} 3^k; 7^{50}$$

$$14. 68$$

$$15. 3n^2 + 3n + 1$$

$$16. GF(x) = \frac{1}{(1-x)^2(1-x^2)} ;$$

$$a_n = \begin{cases} \frac{1}{4}(n+1)2^{n+3} & , \text{ als } n \text{ is oneven,} \\ \frac{1}{4}(n+2)2^n & , \text{ als } n \text{ is even} \end{cases}$$

$$17.a. GF(x) = x^2 \frac{(1-x^6)^2}{(1-x)^2}$$

$$b. GF(x) = x^2 \frac{(1-x^6)^2}{(1-x)^2}$$

$$c. GF(x) = \frac{1}{1-x-x^2-x^3-x^4-x^5-x^6}$$

$$18.b \quad \binom{r-1}{k-1} ; 2^{r-1}$$

$$19. \quad a_n = \frac{1}{5} (2 \cos n \frac{\pi}{2} - \sin n \frac{\pi}{2} + 2^{n-1}) ; \quad A(x) = \frac{1-2x+x^2-x^3}{1-2x+x^2-2x^3}$$

$$20. \quad B(x) = \frac{x^3}{1-2x+x^2-x^3}$$

$$21. \quad B(x) = 1 - \frac{1}{A(x)}$$

$$22. \quad \frac{1}{4} n^2 (n+1)^2$$

$$23.c. GF(x) = \prod_{i \text{ oneven}} \frac{1}{1-x^i}$$

$$d. GF(x) = \frac{1}{(1-x)(1-x^2) \dots (1-x^k)}$$

$$24.b. GF(x) = \frac{x^k}{(1-x)(1-x^2) \dots (1-x^k)}$$

$$26.a. EF(x) = (e^x - 1)^n$$

$$a_r = \sum_{i=0}^n (-1)^i \binom{n}{i} (n-i)^r = n! S(r, n)$$

Hoofdstuk 3

2.a. (0,0,2,2,1,5, 2,6,6,2)

b. $10! - 1$

4. 169.314 ; 350.464

5. 5 2 7 1 9 8 3 6 4. ;
 5 2 7 9 1 8 3 4 6

6. 5 2 7 1 9 6 8 4 3 ;
 5 2 7 1 8 9 3 4 6

7. 1 2 3 6 8 11 12 5 9 10 4 7 ;
 8 1 2 3 12 4 9 11 5 10 7 6

8.a. 2113 ;

 5 1 3 7 6 4 2 ,
 5 1 4 2 3 6 7

b. 1800

14. 16

15. $a_n = 2^n - 1$

16. $a_n = 2^{n+1} - n - 2$

17. 908

Hoofdstuk 4

4. $t = \frac{1}{6}5$

11. 24

12c. D_i , met $D = \text{diag } (\rho(v)) - A$ en D_i de minor van D die ontstaat door schrappen van i^{de} rij en i^{de} kolom.

13. 45 , 12

17. 32

19. 4

20. $d(i,f) = 6$

21. $n \cdot 2^{n-1}$

22. $\binom{n}{3}$; $3\binom{n}{4}$