Hoe verbind je een stel steden met zo weinig mogelijk kilometers asfalt? Hoe maak je een optimaal computernetwerk met kabels die maar een beperkte capaciteit hebben? Veel van zulke problemen zijn op te lossen met het *gretige algoritme*. Hoewel het principe voor de hand ligt, werkt het niet altijd. In dit vierde thema-artikel legt Klaas Pieter Hart dit algoritme uit de discrete wiskunde uit. Ga er even goed voor zitten; hier en daar is enig doorzettingsvermogen nodig!

■ door Klaas Pieter Hart

HEBZUCHT LOONT – NIET ALTIJD

Het kiezen van een beperkt aantal koekjes uit een trommel is een ernstige zaak; je wilt natuurlijk zo veel mogelijk koek verzamelen. Een natuurlijke strategie is: eerst het grootste koekje pakken, dan het op één na grootste, enzovoort. Dit voor de hand liggende idee ziet er ook eenvoudig te programmeren uit: je laat je programma telkens de overgebleven mogelijkheden doorzoeken en de beste overgebleven mogelijkheid kiezen.

Soms is deze strategie echter niet optimaal, omdat er randvoorwaarden aan de uiteindelijke verzameling worden gesteld. Een flauw voorbeeld waarin gretig zijn niet helpt, is het volgende: je hebt drie ballen b_1, b_2 en b_3 die respectievelijk 3, 2 en 2 kilogram wegen. Je taak is ballen te kiezen zó dat het totale gewicht maximaal is. Er zit één adder onder het gras: je keuze moet een deelverzameling van $\{b_1\}$ zijn of van $\{b_2, b_3\}$. Nu keert gretig zijn zich tegen je: meteen de zwaarste bal, b_1, kiezen geeft je in totaal 3 kilogram omdat je niet meer ballen mag kiezen. Het was beter geweest eerst b_2 en dan b_3 te pakken, dan heb je in totaal 4 kilogram te pakken.

Dit voorbeeld lijkt zo flauw omdat je ook zo wel ziet dat $\{b_2, b_3\}$ de beste keuze is. Als je echter een programma schrijft dat die optimale keuze moet opsporen, zul je merken dat het probleem helemaal niet zo eenvoudig is: wat een mens in één oogopslag ziet en doet, kost in een programma heel wat stappen. Je moet bij elke bal kijken in welke toegelaten verzameling hij zit en welke van die verzamelingen de zwaarste is. En denk daarbij niet aan een handvol ballen maar aan miljoenen ballen waar een optimale deelverzameling uit gekozen moet worden.

Nu zijn er problemen waarbij gretig zijn wel werkt en het mooie is dat je die problemen kunt karakteriseren. Het verhaal wordt hier en daar wel wat abstract, maar met pen en papier bij de hand zijn de redeneringen goed te volgen.

Als voorbeeld laten we zien hoe je in een samenhangende graaf optimale opspannende bomen kunt opsporen.

Een *pad* is een rij verschillende punten $v_1, v_2, \ldots, v_k$ waarbij opeenvolgende punten zijn verbonden door een lijn. Als v_1 en v_k ook verbonden zijn door een lijn, dan vormen de lijnen $v_1v_2, v_2v_3, \ldots, v_{k-1}v_k, v_kv_1$ een *circuit*.

Een *boom* is een samenhangende graaf zonder circuits.

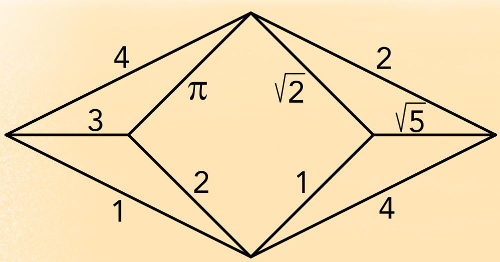
Een *opspannende boom* in een graaf is een verzameling lijnen die een boom vormen en zó dat elk punt eindpunt van een van die lijnen is.

OPSPANNENDE BOMEN Stel je een groep steden voor waartussen wegen lopen, of computers die met kabels verbonden zijn, of pompstations in een netwerk van oliepijpleidingen. Je kunt zo'n situatie beschrijven als een samenhangende graaf (V, E) met een functie $w : E \to [0, \infty)$, zie ook de vorige *Pythagoras*. Hier komt V overeen met de knooppunten, E met de verbindingen en w geeft een eigenschap van de verbindingen aan: bijvoorbeeld het aantal auto's dat per uur over een weg kan, of het aantal liters olie per seconde dat door een leiding gepompt kan worden.

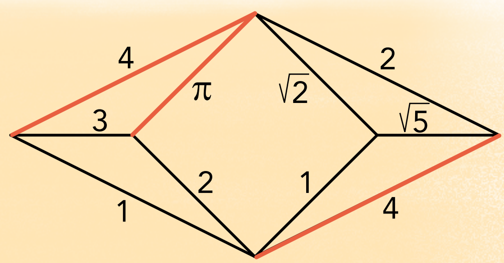
De functie w wordt meestal een *gewichtsfunctie* genoemd en het (totaal)gewicht van een verzameling lijnen is de som van de gewichten van die lijnen.

Een *opspannende boom* verbindt alle knooppunten in een graaf, maar zodanig dat er nergens een circuit (rondwandeling) ontstaat. Het optimaliseringsprobleem is nu, om een opspannende boom te vinden met een totaalgewicht dat zo klein mogelijk (de eerste twee voorbeelden) is, of juist zo groot mogelijk (in het derde voorbeeld).

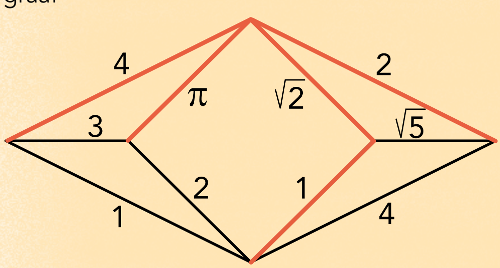
In figuur 1 zie je een voorbeeld van een samenhangende graaf. In figuur 2 staat een opspannende boom getekend met een totaal gewicht van



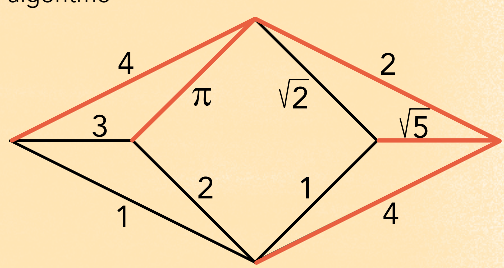
Figuur 1 Voorbeeld van een samenhangende graaf



Figuur 3 De eerste zetten van het gretige algoritme



Figuur 2 Een opspannende boom van de graaf uit figuur 1



Figuur 4 Een opspannende boom via het gretige algoritme

$4 + \pi + \sqrt{2} + 2 + 1 = 7 + \pi + \sqrt{2}$. Deze is niet de zwaarste en ook niet de lichtste.

Opgave 1. Maak een lichtere en een zwaardere opspannende boom.

Laten we eens kijken wat het gretige algoritme oplevert: in dit geval kiezen we telkens een lijn met het hoogste gewicht die we toe kunnen voegen *zonder een circuit te maken*. We kunnen zonder gevaar een lijn van gewicht 4 kiezen, want één lijn maakt geen circuit; de tweede lijn van gewicht 4 ligt los van de eerste en geeft dus nog steeds geen circuit. Daarna kunnen we ook nog de lijn van gewicht π kiezen zonder een circuit te maken, zie figuur 3.

Nu moeten we oppassen: de lijn van gewicht 3 is het zwaarst, maar die mogen we niet meer kiezen want dan maken we een circuit. De lijn van gewicht $\sqrt{5}$ is de op één na zwaarste en die kan veilig toegevoegd worden. Vervolgens kan een van de lijnen van gewicht 2 nog worden toegevoegd om een opspannende boom te maken, zie figuur 4.

De rode lijnen in figuur 4 vormen een opspannende boom: als we een lijn toevoegen, creëren we een circuit. Deze heeft ook het maximale totale gewicht. Dat is in dit geval nog wel met de hand na te gaan, maar voor (veel) grotere problemen is dat niet praktisch. We zullen gaan bewijzen dat het gretige algoritme het probleem van de opspannende bomen altijd optimaal oplost.

Opgave 2. Gebruik het gretige algoritme om een opspannende boom van minimaal gewicht te vinden (wees niet gretig maar bescheiden).

Het bewijs van de bewering dat het gretige algoritme altijd werkt, doen we in twee stappen. We beschrijven een abstracte structuur (matroïd) en bewijzen dat in een dergelijke structuur het algoritme altijd werkt; vervolgens laten we zien dat het opspannende-bomenprobleem zich in een matroïd afspeelt.

MATROÏDEN Een matroïd is een familie deelverzamelingen, $\mathcal{M}$, van een vaste eindige verzameling X. (Het woord 'matroïd' komt van 'matrix'. Voor wie iets van lineaire algebra weet: als je een matrix hebt, dan vormen de lineair onafhankelijke verzamelingen kolommen een matroïd.)

De familie $\mathcal{M}$ moet aan twee eisen voldoen:
1. als $M \in \mathcal{M}$ en $N \subseteq M$, dan ook $N \in \mathcal{M}$;
2. als $M, N \in \mathcal{M}$ en $|M| < |N|$, dan is er een $x \in N \setminus M$ zó dat $M \cup \{x\} \in \mathcal{M}$.

Met $N \setminus M$ wordt bedoeld: de verzameling N minus de elementen die ook in M zitten. De familie $\mathcal{M}$ speelt de rol van de 'toegelaten' verzamelingen; het legt beperkingen op aan de deelverzamelingen van X die je kunt kiezen. In de definitie staat dat als een verzameling toegelaten is, elke deelverzameling daarvan ook toegelaten is, en als je twee toegelaten verzamelingen hebt van verschillende grootte, dan kun je uit de grotere extra elementen lenen om de kleinere uit te breiden tot toegelaten verzamelingen met meer elementen.

Deze definitie (uit 1935) is het resultaat van heel wat denkwerk en brengt een heel scala aan verschijnselen onder één noemer.

In de volgende twee paragrafen laten we zien: als $\mathcal{M}$ een matroïd op een verzameling X is en $w : X \to [0, \infty)$ een gewichtsfunctie is, dan levert toepassing van het gretige algoritme altijd een verzameling in $\mathcal{M}$ van maximaal gewicht; en *omgekeerd*: als het gretige algoritme bij elke functie w de optimale oplossing vindt, dan hebben we met een matroïd te maken.

Het is informeel wel duidelijk hoe het gretige algoritme werkt, maar voor het komende bewijs moeten we wel een nette beschrijving hebben.

Stap 1: kies $x_1 \in X$ met $\{x_1\} \in \mathcal{M}$ en $w(x_1)$ zo groot mogelijk.
Stap i + 1: als $x_1, ..., x_i$ gekozen zijn, kies dan $x_{i+1} \in X$ (ongelijk aan de reeds gevonden punten) zó dat $\{x_1, ..., x_i, x_{i+1}\} \in \mathcal{M}$ en met $w(x_{i+1})$ zo groot mogelijk.
Als zo'n keuze niet mogelijk is, stop dan.

Merk op dat dit precies is wat we bij de opspannende boom gedaan hebben! Het flauwe voorbeeld, waarin het gretige algoritme niet werkte, is ook geen matroïd: aan eis 1 kunnen we nog wel voldoen door $\mathcal{M} = \{\emptyset, \{b_1\}, \{b_2\}, \{b_3\}, \{b_2, b_3\}\}$ te nemen, maar eis 2 lukt niet: $\{b_1\}$ heeft één element en $\{b_2, b_3\}$ heeft er twee, maar we kunnen geen element lenen om $\{b_1\}$ uit te breiden *en* binnen $\mathcal{M}$ te blijven.

GRETIG ZIJN WERKT We moeten eerst een extra notie invoeren, die van *basis*. Sommige verzamelingen in een matroïd zijn maximaal, dat wil zeggen dat er geen grotere verzamelingen in de matroïd zijn; zo'n maximale verzameling noemen we een basis. Wat opmerkelijk is, is dat elk tweetal bases even groot is. Kijk maar naar eis 2: als $|M| < |N|$, dan kunnen we uit N een element lenen om M te vergroten, dus M is geen basis.

Het gretige algoritme produceert altijd een basis: we stoppen pas als we onze verzameling niet kunnen uitbreiden. Maar waarom is die basis zeker van het grootste gewicht? Dat gaan we nu bewijzen.

Noem de gevonden basis M. We gaan M vergelijken met een basis van maximaal gewicht; dergelijke bases bestaan, want we kunnen in principe van elke basis zijn totaalgewicht uitrekenen en omdat er maar eindig veel bases zijn, is er een maximum totaalgewicht. Neem dus een basis N van maximaal gewicht en wel zo dat de doorsnede $M \cap N$ zo veel mogelijk elementen heeft. Als $M = N$ zijn we klaar; neem eens aan dat $M \neq N$. Omdat M en N bases zijn, hebben ze even veel elementen en dat betekent dat $M \setminus N$ en $N \setminus M$ allebei niet leeg zijn. Neem de eerste i met $x_i \in M \setminus N$. We kijken naar $Y = N \cup \{x_i\}$ en naar $\mathcal{N} = \{A \in \mathcal{M} : A \subseteq Y\}$.

Opgave 3. Ga na dat $\mathcal{N}$ een matroïd op Y is.

Opgave 4. Ga na dat $\{x_1, ..., x_i\}$ ook gekozen wordt als we het gretige algoritme toepassen binnen Y en met gebruik van de matroïd $\mathcal{N}$.

Nu komt de cruciale opmerking: als we het gretige algoritme binnen Y voortzetten, zullen we alle punten van $N \setminus M$, op één na, kiezen. Het algoritme produceert namelijk een basis B in $\mathcal{N}$ en die moet even groot zijn als N, maar omdat $Y = N \cup \{x_i\}$ kan B maar één element met N verschillen. Het niet gekozen punt noemen we y. Nu geldt $w(x_i) \geq w(y)$, want x_i werd boven y verkozen en y was een legitieme keuze geweest want $\{x_1, ..., x_{i-1}, y\} \subseteq N$. Als we nu het totaalgewicht $w(N)$ van $w(B)$ aftrekken, houden we $w(x_i) - w(y)$ over en dat is niet negatief. Dus $w(N) \leq w(B)$ en dus $w(B) = w(N)$, omdat $w(N)$ maximaal is. Maar dan heeft B ook een maximaal totaalgewicht en in $M \cap B$ zit één punt meer dan in $M \cap N$ (namelijk x_i); tegenspraak, dus toch $M = N$.

ALLEEN MATROÏDEN ZIJN GESCHIKT Stel nu eens dat $\mathcal{M}$ een familie deelverzamelingen van een verzameling X is die aan eis 1 voldoet en met de eigenschap dat, ongeacht $w : X \to [0, \infty)$, het gretige algoritme altijd een element van $\mathcal{M}$ van maximaal totaalgewicht oplevert.

We gaan eis 2 bewijzen. Neem dus $M, N \in \mathcal{M}$ met $|M| < |N|$. Om de gedachten te bepalen, nemen we aan dat M vijf elementen heeft en N acht. Dan heeft $N \setminus M$ drie elementen meer dan $M \setminus N$ (trek $|M \cap N|$ maar van 8 en 5 af). We kunnen dan twee getallen s en t met $s > t > 0$ vinden zó dat

$$s|M \setminus N| < t|N \setminus M|.$$

Opgave 5. Loop de mogelijkheden voor $|M \cap N|$ na en bepaal telkens geschikte s en t.

Nu definiëren we een functie $w : X \to [0, \infty)$ door

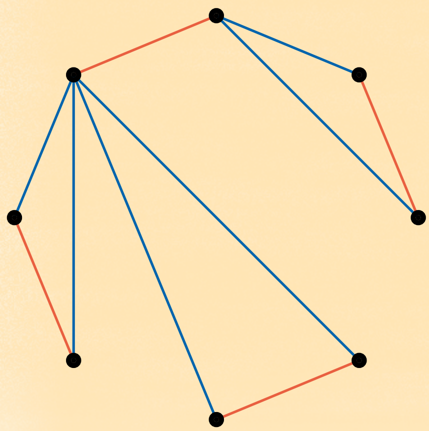
$$w(x) = \begin{cases} s & \text{als } x \in M \\ t & \text{als } x \in N \setminus M \\ 0 & \text{als } x \notin M \cup N \end{cases}$$

Opgave 6. Ga na dat $w(M) = s|M|$ en $w(N) = s|M \cap N| + t|N \setminus M|$ en dat $w(M) < w(N)$.

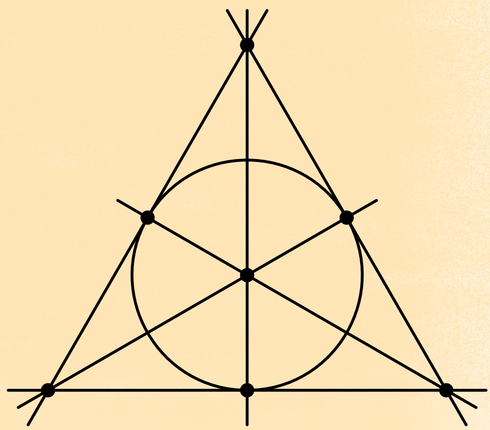
Het gretige algoritme kiest eerst alle elementen van M, maar omdat het totaalgewicht van M niet maximaal is, loopt het kennelijk door en de enige manier om een groter gewicht te kiezen is door een $x \in N \setminus M$ te nemen. Dan zit $M \cup \{x\}$ blijkbaar in $\mathcal{M}$.

OPSPANNENDE BOMEN Onze laatste taak is het volgende te laten zien: als (V, E) een graaf is, dan is $\mathcal{B} = \{M \subseteq E : M \text{ bevat geen circuit}\}$ een matroïd. Aan eis 1 is natuurlijk voldaan: als in M geen circuit zit, dan bevat een deel van M zeker geen circuit.

Nu eis 2 nog. We nemen twee verzamelingen lij-



Figuur 5



Figuur 6 Het Fanovlak

nen, M en N, die geen circuits bevatten en wel zo dat $|M| < |N|$. De verzamelingen eindpunten van die lijnen noemen we V_M en V_N. Als er een $v \in V_N \setminus V_M$ bestaat, zijn we klaar: neem een lijn $e \in N$ met v als eindpunt; dan zit e niet in M en $M \cup \{e\}$ bevat geen circuits, omdat e met zijn nieuwe eindpunt aan geen lijn uit M vast zit.

Blijft over het geval dat $V_N \subseteq V_M$, dus elk eindpunt van elke lijn in N is ook eindpunt van een lijn uit M. In figuur 5 zie je hoe zo'n situatie kan optreden. Er zijn vier rode lijnen en zes blauwe die dezelfde acht punten als eindpunten hebben.

Bekijk de blauwe lijnen: die vormen twee samenhangende stukken en 6 + 2 = 8: het aantal lijnen plus het aantal stukken is gelijk aan het aantal punten.

Dat geldt ook voor de rode lijnen: vier lijnen en ook vier stukken, 4 + 4 = 8.

Je ziet in figuur 5 veel paren rode lijnen die door een blauwe verbonden worden; neem een blauwe lijn die twee rode verbindt, die kun je zonder gevaar aan de rode verzameling toevoegen: één enkele brug tussen twee rode stukken kan geen circuit veroorzaken.

Opgave 7. Stel dat (V, E) een boom is, een samenhangende graaf zonder circuits dus. Bewijs dat $|V| = |E| + 1$. *Aanwijzing*: toon aan dat er een eindpunt is, een punt dat eindpunt van maar één lijn is. Als je dat punt en zijn lijn weglaat, hou je een boom over, waarin je weer een eindpunt kunt vinden...

Laat nu c_M het aantal samenhangende stukken (componenten) van M zijn en c_N dat van N. Pas opgave 7 toe om in te zien dat $|V_M| = |M| + c_M$ en $|V_N| = |N| + c_N$ (net als in de rode en blauwe voorbeelden). Uit $|M| < |N|$ en $V_N \subseteq V_M$ volgt nu dat $c_M > c_N$.

Nu volgt met een eenvoudig telargument dat er een component D van N is die aan twee (of meer) componenten van M vast zit: elke component van M zit aan een component van N vast en M heeft meer componenten dan N. Neem twee van die componenten, C_1 en C_2, en punten v_1 en v_2 in respectievelijk $C_1 \cap D$ en $C_2 \cap D$. Er is een pad door D met lijnen uit N dat v_1 en v_2 verbindt. Bedenk nu zelf waarom een van die lijnen aan M kan worden toegevoegd zonder een circuit te maken.

EEN PAAR VOORBEELDEN Er zijn veel soorten matroïden. We geven een paar eenvoudige voorbeelden.

Figuur 6 heet het Fanovlak; het bestaat uit zeven punten en zeven lijnen (ook de cirkel telt als lijn) en het voldoet aan enkele meetkunde-axioma's, zoals: 'door elk tweetal punten gaat een lijn', maar ook aan 'elk tweetal lijnen snijdt elkaar'. Er is een heel deelgebied van de discrete wiskunde dat over dit soort *eindige* meetkunde gaat.

De familie van alle driepuntsverzamelingen die niet op één lijn liggen, bepalen een matroïd waarvan zij precies de bases zijn.

Als X een verzameling is en $A_1, A_2, ..., A_k$ is een stel deelverzamelingen, dan heet $\{x_1, x_2..., x_k\}$ een *transversaal* voor de A_i als de x_i verschillend zijn en $x_i \in A_i$ voor elke i; een partiële transversaal is een transversaal voor een deel van de A_i's. De familie van alle (partiële) transversalen is een matroïd op X.

Opgave 8. Lees het artikel in de vorige *Pythagoras* over de Huwelijksstelling van Hall nog een keer en onderzoek hoe de transversalenmatroïd de vrouwen kan helpen het totale kapitaal van hun (aanstaande) echtgenoten te optimaliseren. ■