1.2 Bomen

1.2.1 Algemeen

Beschouw eerst een niet-gerichte graaf. Een *boom* is een samenhangende graaf die geen kringen bevat. Een boom wordt meestal genoteerd met de letter T (tree). Een *bos* is een graaf waarvan de componenten bomen zijn. De volgende stelling geeft zes equivalente karakteriseringen van het begrip boom.

Stelling 1.4
Voor een normale graaf T zijn de volgende zes beweringen equivalent:
1. *T is een boom.*
2. *T heeft geen kringen en heeft $n-1$ takken.*
3. *T is samenhangend en heeft $n-1$ takken.*
4. *T is samenhangend en na het weglaten van een willekeurige tak niet meer.*
5. *Ieder paar knooppunten van T is door precies één keten verbonden.*
6. *T heeft geen kringen, maar door het toevoegen van een willekeurige tak ontstaat precies één kring.*

Bewijs
Indien $n = 1$ dan worden de beweringen triviaal. We nemen daarom aan dat $n \geq 2$.

$1 \Rightarrow 2$: We moeten bewijzen dat T $n-1$ takken heeft en doen dat met inductie ($n = 2$ triviaal). Door een tak weg te laten ontstaan twee deelgrafen (anders was er een kring) die weer een boom zijn, met n_1 en n_2 knooppunten, waarbij $n_1 + n_2 = n$. Volgens de inductieveronderstelling hebben deze $n_1 - 1$ resp. $n_2 - 1$ takken.
T heeft dus $(n_1 - 1) + (n_2 - 1) + 1 = n - 1$ takken.

$2 \Rightarrow 3$: Indien T niet samenhangend is, dan heeft T $k \geq 2$ componenten, die geen kringen bevatten, dus een boom zijn. Omdat uit bewering 1 bewering 2 volgt, heeft iedere component één tak minder dan het aantal knooppunten, d.w.z. T heeft $n - k$ takken. Anderzijds is het aantal takken van T gelijk aan $n - 1$, waaruit volgt dat $k = 1$, wat de tegenspraak geeft.

$3 \Rightarrow 4$: Stel tak e wordt verwijderd en er is nog steeds samenhang. Dan bevat $T - e$ een boom, dus heeft $T - e$ minstens $n - 1$ takken, wat de tegenspraak geeft.

$4 \Rightarrow 5$: Omdat T samenhangend is, is ieder paar knooppunten door minstens één keten verbonden. Als v en w door twee ketens zijn verbonden, dan is er een kring en dus een tak weg te laten zonder de samenhang te verstoren, wat in tegenspraak is met bewering 4.

$5 \Rightarrow 6$: Als T kring heeft, dan is er een paar knooppunten dat door twee ketens verbonden is: tegenspraak met bewering 5, dus T heeft geen kringen. Als een tak e wordt toegevoegd, dan ontstaat een kring, want de begin- en eindpunten van e waren al verbonden. Er kunnen ook geen twee kringen ontstaan, want deze zouden beide e bevatten en door beide kringen zonder e te doorlopen zou een kring in T ontstaan.

$6 \Rightarrow 1$: Stel T niet samenhangend. Tak toevoegen tussen knooppunten uit twee verschillende componenten geeft geen kring: tegenspraak. □

N.B. Waar gebruiken we normaliteit in bovenstaand bewijs?

De unieke kring die ontstaat door aan de boom een tak toe te voegen heet een met de boom *geassocieerde kring*. Een deelgraaf van een samenhangende graaf die een boom is en alle knooppunten van de graaf bevat heet een *opspannende boom*.

Vervolgens beschouwen we een gerichte graaf. Dit is een *boom* als de bijbehorende niet-gerichte graaf een boom is. Als in een boom voor een knooppunt r geldt dat er een pad is van r naar alle andere knooppunten, dan spreken we over een *in r gewortelde gerichte boom*.
In een in r gewortelde gerichte boom heet knooppunt i een *voorganger* van knooppunt j als er een pad is van i naar j; in dat geval heet j een *afstammeling* van i. Als de boom de pijl (i, j) bevat, dan heet i de *ouder* van j en j een *kind* van i. Een knooppunt zonder uitgaande pijlen heet een *blad*, en een knooppunt dat geen blad is heet een *inwendig knooppunt*.

Stelling 1.5
Zij de gerichte graaf $T = (V, A)$ een boom.
T is in r gewortelde gerichte boom d.e.s.d. als $d^-(r) = 0$ en $d^-(i) = 1$ voor alle $i \neq r$.

Bewijs
$\Rightarrow$ Iedere $i \neq r$ heeft minstens één inkomende pijl. Omdat er precies $n - 1$ pijlen zijn moet dus gelden: $d^-(r) = 0$ en $d^-(i) = 1$ voor alle $i \neq r$.
$\Leftarrow$ Neem een $i \neq r$. Omdat we vanuit ieder knooppunt $j \neq r$ i.v.m. $d^-(j) = 1$ kunnen teruglopen, eindigt zo'n wandeling in r, d.w.z. in de boom is een pad van r naar i. □

Vraag 1.7
Laat T een boom zijn.
a. Bewijs dat T minstens twee knooppunten van de graad 1 heeft.
b. Veronderstel dat het aantal takken van T even is.
Bewijs dat er minstens één knooppunt met even graad is.

Vraag 1.8
Bewijs dat een boom een bipartiete graaf is.

Vraag 1.9
Zij $T = (V, F)$ een opspannende boom van de normale, samenhangende graaf $G = (V, E)$.
Toon aan dat geldt:
a. Iedere co-kring van G heeft een tak gemeen met F.
b. Iedere kring van G heeft een tak gemeen met $E \backslash F$.

ga verder met $VISIT(5)$: omdat $Z[5] = N[5]$ krijgen we:

$S = \emptyset;\ S^*[5] = 0;\ COMP[5] = 4;\ k = 5.$

De streng samenhangende componenten zijn: $C_1 = \{2, 4, 3\};\ C_2 = \{1\};\ C_3 = \{6\}$ en $C_4 = \{5\}$.

Stelling 1.19
Algoritme 1.6 heeft complexiteit $\mathcal{O}(p)$, waarbij $p = max(n, m)$.

Bewijs
Uit Stelling 1.11 volgt dat, zonder de bewerkingen aan Z, S en S^* mee te tellen, de complexiteit $\mathcal{O}(p)$ is. De bewerkingen aan Z zijn: $\mathcal{O}(n)$ bij de eerste toekenning in stap 3; stap 4 geeft voor het totale algoritme $\mathcal{O}(m)$ en stap 5 $\mathcal{O}(n)$. Totaal geeft dit $\mathcal{O}(p)$ voor het werk aan Z.
Voor het werk aan S en S^* geldt het volgende. Ieder knooppunt wordt éénmaal op S gezet en er ook éénmaal afgehaald; ieder knooppunt krijgt bij initialisatie de S^*-waarde 0, dan zodra het bereikt wordt de waarde 1 en tenslotte, nadat het van S is afgehaald weer de waarde 0.
Verder wordt voor iedere pijl (v, w) in stap 4b getest of $S^*[w] = 1$. De totale hoeveelheid werk aan S en S^* is dus $\mathcal{O}(n)$. $\square$

Vraag 1.15
Beschouw een gerichte samenhangende graaf G.
Toon aan dat G streng samenhangend is d.e.s.d. als iedere pijl van G tot een ronde behoort.

1.2.6 Minimale opspannende boom

Beschouw een niet-gerichte normale samenhangende graaf $G = (V, E)$ waarin aan iedere tak (i, j) een reëel getal l_{ij}, de *lengte* geheten, is toegekend. Het probleem is om in deze graaf een opspannende boom te vinden met minimale lengte.
We zullen twee methoden behandelen, de methode van Prim en de methode van Kruskal. Beide methoden zijn van het type *gretig algoritme*, d.w.z. dat in iedere iteratiestap een keuze wordt gemaakt die, kortzichtig gezien, het beste is, d.w.z. *lokaal optimaal*. Het zal blijken dat deze keuzes ook leiden tot een *globaal optimum*.

Voorbeeld 1.12
Beschouw een aantal plaatsen waarvan de onderlinge lengtes in onderstaande tabel staan. Hoe moeten deze plaatsen door electriciteitskabels worden verbonden zódat iedere plaats is aangesloten en er zo weinig mogelijk kabel wordt gebruikt?

	1	2	3	4	5	6	7
1	-	43	27	60	40	70	50
2	43	-	28	20	40	35	24
3	27	28	-	26	14	35	50
4	60	20	26	-	19	18	40
5	40	40	14	19	-	24	40
6	70	35	35	18	24	-	22
7	50	24	50	40	40	22	-

Voorbeeld 1.13
Beschouw een inlichtingendienst die n agenten heeft zitten in vijandelijk gebied. Als agent i een bericht doorstuurt naar agent j, dan valt dit met kans p_{ij} in vijandelijke handen. Via welke verbindingen moet een bericht worden gecommuniceerd zodat het iedere agent bereikt, terwijl de kans dat het in vijandelijke handen komt minimaal is?
Om iedere agent te bereiken moeten de verbindingen minimaal een opspannende boom vormen. Extra verbindingen vergroten de kans dat het bericht in vijandelijke handen valt, dus we zoeken naar de beste opspannende boom T. Voor een gegeven boom T is de kans dat het bericht niet in handen van de vijand komt gelijk aan: $\prod_{(i,j)\in T}(1-p_{ij})$. Het probleem luidt dus:

$$\begin{array}{llll} \max_T \prod_{(i,j)\in T}(1-p_{ij}) & \Leftrightarrow & \max_T \; log \prod_{(i,j)\in T}(1-p_{ij}) & \Leftrightarrow \\ \max_T \sum_{(i,j)\in T} log(1-p_{ij}) & \Leftrightarrow & \min_T \sum_{(i,j)\in T} l_{ij}, & \end{array}$$

waarbij $l_{ij} = -log(1-p_{ij})$, zodat $l_{ij} \geq 0$ voor alle (i,j). Dit is dus het probleem van de minimale opspannende boom met lengtefunctie $l_{ij} = -log(1-p_{ij}) \geq 0$ voor alle (i,j).

We geven allereerst twee (duale) karakteriseringen van het begrip minimale opspannende boom.

Stelling 1.20
Een opspannende boom T heeft minimale lengte d.e.s.d. als voor iedere tak e die niet tot T behoort geldt dat $l(e) \geq l(f)$ voor alle takken f in de kring $C_T(e)$ die ontstaat door e aan T toe te voegen.

Bewijs
$\Rightarrow$ Stel $l(e) < l(f)$ voor een $f \in C_T(e)$. Maar dan is de opspannende boom $T - \{f\} + \{e\}$ (ga zelf na dat dit een opspannende boom is) een boom met kleinere lengte dan $l(T)$: tegenspraak.

$\Leftarrow$ Laat T^* een opspannende boom zijn met minimale lengte. We moeten aantonen dat $l(T) = l(T^*)$ en doen dat met inductie naar k, het aantal takken in $T^*\backslash T$.
Als $k = 0$, dan is $T^* = T$ en klopt de bewering. Nu de algemene inductiestap.
Neem $e^* \in T^*\backslash T$, dan valt $T^* - \{e^*\}$ in twee deelbomen uiteen, zeg T_1 en T_2.
De keten $C_T(e^*)\backslash\{e^*\}$ heeft één eindpunt in T_1 en één eindpunt in T_2.

Dus er is een $e \in T \backslash T^*$ en $e \in C_T(e^*)$ met één eindpunt in T_1 en één eindpunt in T_2.

Dus $T^* - \{e^*\} + \{e\}$ is weer een opspannende boom. Uit de minimaliteit van T^* volgt $l(e) \geq l(e^*)$, terwijl uit het gegeven van de stelling volgt $l(e) \leq l(e^*)$, zodat $l(e) = l(e^*)$.

We kunnen dus een tak van T^* vervangen door een tak van T zonder de lengte van de boom te veranderen. Met het inductie argument is hiermee het bewijs geleverd. $\square$

Stelling 1.21

Een opspannende boom T heeft minimale lengte d.e.s.d. als voor iedere tak e die tot T behoort geldt dat $l(e) \leq l(f)$ voor alle takken f in de snede $S_T(e)$ die ontstaat door e uit T te verwijderen.

Bewijs

$\Rightarrow$ Stel $l(e) > l(f)$ voor een $f \in S_T(e)$. Nu is de opspannende boom $T - \{e\} + \{f\}$ een opspannende boom met kleinere lengte dan $l(T)$: tegenspraak.

$\Leftarrow$ We zullen nu laten zien dat aan de voorwaarde van Stelling 1.20 is voldaan, waaruit volgt dat de opspannende boom minimaal is.
Zij e een tak die niet tot T behoort en $f \neq e$ een element van $C_T(e)$. Omdat $e \in S_T(f)$, is volgens het gegeven $l(f) \leq l(e)$. $\square$

We zullen nu een generiek algoritme opstellen voor het bepalen van een opspannende boom met minimale lengte.

Algoritme 1.7 *Opspannende boom met minimale lengte (generieke versie)*

1. Voor $i = 1, 2, \ldots, n$ doe: $V_i = \{i\}$ en $T_i = \emptyset$.
2. Voor $k = 1, 2, \ldots, n-1$ doe:
   a. Kies een i met $V_i \neq \emptyset$.
   b. Kies de kleinste tak die precies één uiteinde in V_i heeft, zeg tak e, en laat het andere uiteinde in V_j zitten.
   c. $V_i := V_i \cup V_j$; $V_j = \emptyset$; $T_i := T_i \cup T_j \cup \{e\}$; $T_j = \emptyset$.
3. $T = T_i$ en STOP.

Stelling 1.22

Algoritme 1.7 is correct, d.w.z. de laatste boom T is een opspannende boom met minimale lengte.

Bewijs

We zullen met inductie naar k bewijzen dat alle takken van de deelbomen T_i, $1 \leq i \leq n$ tot een opspannende boom met minimale lengte behoren. Bij de start klopt dit en als de inductiestap is bewezen, dan geldt het ook voor de eindsituatie, d.w.z. voor de boom T.

Veronderstel dat in stap k de tak e wordt gekozen met één uiteinde in V_i en dat e niet tot een minimale opspannende boom T^* behoort. De kring $C_{T^*}(e)$ bevat een tak $f \neq e$ met één uiteinde in V_i. Volgens Stelling 1.20 geldt: $l(e) \geq l(f)$.
Anderzijds is e zo gekozen dat $l(e) \leq l(f)$, zodat $l(e) = l(f)$. Dus $T' = T^* + \{e\} - \{f\}$ is weer een minimale opspannende boom die e bevat: tegenspraak. $\square$

Opmerking
Door in Algoritme 1.7 de mogelijke keuze voor V_i nader te specificeren ontstaan diverse algoritmen, waarvan ook de complexiteit kan worden bepaald.

De methode van Prim

Bij de methode van Prim[4] beginnen we met V_1 en gaan we V_1 steeds uitbreiden met het knooppunt dat er het dichtste bij ligt. De andere V_i's blijven dus steeds uit $\{i\}$ bestaan. De afstand van zo'n i tot V_1 noemen we d_i en het knooppunt van V_1 dat het dichtste bij i ligt noteren we met w_i. Deze d_i's en w_i's worden in iedere iteratie herberekend. Het algoritme is nu als volgt.

Algoritme 1.8 *Opspannende boom met minimale lengte (Prim)*

1. Laat $W = \{1\},\ T = \emptyset;\ w_i = 1,\ d_i = l_{i1},\ i = 2, 3, \ldots, n.$
2. Zolang $W \neq V$ doe:
   a. Laat $k \notin W$ met k zdd. $d_k = min_{i \notin W}\ d_i$.
   b. $W := W \cup \{k\};\ T := T \cup \{(k, w_k)\}$.
   c. Voor iedere $i \notin W$ doe: als $l_{ik} < d_i:\ d_i := l_{ik}$ en $w_i := k$.

Voorbeeld 1.14
Beschouw het probleem uit Voorbeeld 1.12
Iteratie 1:
$W = \{1\},\ T = \emptyset;\ w_i = 1,\ 2 \leq i \leq 7;\ d_2 = 43,\ d_3 = 27,\ d_4 = 60,\ d_5 = 40,\ d_6 = 70,\ d_7 = 50.$
Iteratie 2:
$k = 3;\ W = \{1, 3\},\ T = \{(3, 1)\};\ w_2 = 3,\ d_2 = 28,\ w_4 = 3,\ d_4 = 26,\ w_5 = 3,\ d_5 = 14,$
$w_6 = 3,\ d_6 = 35.$
Iteratie 3:
$k = 5;\ W = \{1, 3, 5\},\ T = \{(3, 1), (5, 3)\};\ w_4 = 5,\ d_4 = 19,\ w_6 = 5,\ d_6 = 24,\ w_7 = 5,\ d_7 = 40.$
Iteratie 4:
$k = 4;\ W = \{1, 3, 5, 4\},\ T = \{(3, 1), (5, 3), (4, 5)\};\ w_2 = 4,\ d_2 = 20,\ w_6 = 4,\ d_6 = 18.$
Iteratie 5:

[4] R.C. Prim, *Shortest connection networks and some generalizations*, Bell Systems Technical Journal 36 (1957) 1389–1401.

$k = 6;\ W = \{1, 3, 5, 4, 6\},\ T = \{(3, 1), (5, 3), (4, 5), (6, 4)\};\ w_7 = 6,\ d_7 = 22.$
Iteratie 6:
$k = 2;\ W = \{1, 3, 5, 4, 6, 2\},\ T = \{(3, 1), (5, 3), (4, 5), (6, 4), (2, 4)\}.$
Iteratie 7:
$k = 7;\ W = \{1, 3, 5, 4, 6, 2, 7\},\ T = \{(3, 1), (5, 3), (4, 5), (6, 4), (2, 4), (7, 6)\}.$
De minimale opspannende boom bevat dus de verbindingen $(3, 1), (5, 3), (4, 5), (6, 4), (2, 4)$ en $(7, 6)$. De lengte van de boom is 120.

Stelling 1.23
De methode van Prim geeft een minimale opspannende boom en heeft complexiteit $\mathcal{O}(n^2)$.

Bewijs
De correctheid volgt uit Stelling 1.22. Stap 1 van het algoritme heeft complexiteit $\mathcal{O}(n)$.
Stap 2 heeft voor iedere iteratie complexiteit $\mathcal{O}(n)$, immers:
onderdeel a: $\mathcal{O}(n)$, onderdeel b: $\mathcal{O}(1)$ en onderdeel c: $\mathcal{O}(n)$. □

De methode van Kruskal

Bij deze methode[5] worden V_i en e zó gekozen dat $l(e)$ minimaal is voor alle mogelijke keuzes. Mogelijke keuzes zijn takken die geen kringen genereren en dus niet beide uiteinden in dezelfde V-verz. hebben. Om dit laatste na te gaan, houden we van alle knooppunten bij in welke component deze zitten: $c_i = c$ d.e.s.d. als knooppunt i in de c-de component zit.
Het bijhouden van deze componenten gebeurt, indien (i, j) aan T toegevoegd gaat worden (dan bevat $T \cup (i, j)$ dus geen kring), als volgt:
1. als i en j beide niet tot T behoren: i en j komen beide in een nieuwe component;
2. als precies één van de punten i en j tot T behoort, zeg i: j in dezelfde component als i;
3. als i en j beide tot T behoren (en dus in verschillende componenten zitten):
   alle knooppunten uit de component van j komen in dezelfde component als die van i.

Verder sorteren we de takken e_k naar oplopende lengte: $l(e_1) \leq l(e_2) \leq \cdots \leq l(e_m)$.
Dit geeft het volgende algoritme.

Algoritme 1.9 *Opspannende boom met minimale lengte (Kruskal)*

1. a. $W = T = \emptyset;\ c_i = 0,\ 1 \leq i \leq n;\ k = c = 1.$
   b. Sorteer de takken zdd. $l(e_1) \leq l(e_2) \leq \cdots \leq l(e_m)$.
2. Zolang $W \neq V$ doe:
   a. Neem e_k, zeg $e_k = (i, j)$.

[5] J.B. Kruskal, *On the shortest spanning subtree of a graph and the traveling salesman problem*, Proceedings of the American Mathematical Society 7 (1956) 48–50.

b. Als $c_i = c_j \neq 0$: $k := k+1$ en herhaal stap 2.

c. Als $c_i = c_j = 0$: $T := T \cup \{(i,j)\}$: $W := W \cup \{i\} \cup \{j\}$; $c_i = c_j = c$; $c := c+1$; $k := k+1$ en herhaal stap 2.

d. Als $c_i \neq 0$ en $c_j = 0$: $T := T \cup \{(i,j)\}$; $W := W \cup \{j\}$; $c_j := c_i$; $k := k+1$ en herhaal stap 2.

e. Als $c_i = 0$ en $c_j \neq 0$: $T := T \cup \{(i,j)\}$; $W := W \cup \{i\}$; $c_i := c_j$; $k := k+1$ en herhaal stap 2.

f. Als $c_i \neq c_j$: $T := T \cup \{(i,j)\}$; $c_p := c_i$ voor alle $p \in W$ met $c_p = c_j$; $k := k+1$ en herhaal stap 2.

Voorbeeld 1.15
Beschouw het probleem uit Voorbeeld 1.12
Initialisatie:
$W = T = \emptyset$; $c_i = 0$, $i = 1, 2, \ldots, 7$; $k = c = 1$.
$e_1 = (3,5)$, $e_2 = (4,6)$, $e_3 = (4,5)$, $e_4 = (2,4)$, $e_5 = (6,7)$, $e_6 = (2,7)$, $e_7 = (5,6)$, $e_8 = (3,4)$, $e_9 = (1,3)$, $e_{10} = (2,3)$, $e_{11} = (2,6)$, $e_{12} = (3,6)$, $e_{13} = (1,5)$, $e_{14} = (2,5)$, $e_{15} = (4,7)$, $e_{16} = (5,7)$, $e_{17} = (1,2)$, $e_{18} = (1,7)$, $e_{19} = (3,7)$, $e_{20} = (1,4)$, $e_{21} = (1,6)$.
Iteratie 1: $e_1 = (3,5)$; $T = \{(3,5)\}$; $W = \{3,5\}$; ;$c_3 = c_5 = 1$; $c = 2$; $k = 2$.
Iteratie 2: $e_2 = (4,6)$; $T = \{(3,5),(4,6)\}$; $W = \{3,5,4,6\}$; $c_4 = c_6 = 2$; $c = 3$; $k = 3$.
Iteratie 3: $e_3 = (4,5)$; $T = \{(3,5),(4,6),(4,5)\}$; $c_3 = c_5 = 2$; $k = 4$.
Iteratie 4: $e_4 = (2,4)$; $T = \{(3,5),(4,6),(4,5),(2,4)\}$; $W = \{3,5,4,6,2\}$; $c_2 = 2$; $k = 5$.
Iteratie 5: $e_5 = (6,7)$; $T = \{(3,5),(4,6),(4,5),(2,4),(6,7)\}$; $W = \{3,5,4,6,2,7\}$; $c_7 = 2$; $k = 6$.
Iteratie 6: $e_6 = (2,7)$; $k = 7$.
Iteratie 7: $e_7 = (5,6)$; $k = 8$.
Iteratie 8: $e_8 = (3,4)$; $k = 9$.
Iteratie 9: $e_9 = (1,3)$; $T = \{(3,5),(4,6),(4,5),(2,4),(6,7),(1,3)\}$; $W = \{3,5,4,6,2,7,1\}$; $c_1 = 2$; $k = 10$.

Stelling 1.24
Algoritme 1.9 geeft een minimale opspannende boom en heeft complexiteit $\mathcal{O}\{max(m\,log\,n, n^2)\}$.

Bewijs
De correctheid volgt weer uit Stelling 1.22.
De initialisatie heeft complexiteit $\mathcal{O}(m\,log\,m)$ (vanwege het sorteren van m elementen).[6]
Omdat $log\,m \leq log\,n^2 = 2log\,n$, is dit ook te schrijven als $\mathcal{O}(m\,log\,n)$.

[6]Zie hiervoor bijvoorbeeld: D.E. Knuth: *The art of computer programming*, Volume 3: *Sorting and searching*, Second edition, Addison-Wesley (1998).

Stap 2 wordt $\mathcal{O}(m)$ keer uitgevoerd en bestaat uit onderdeel 2a en één van de onderdelen 2b t/m 2f. Afgezien van de opdracht in 2f om de c's aan te passen heeft ieder onderdeel complexiteit $\mathcal{O}(1)$. In 2f worden twee bestaande deelbomen tot één boom samengevoegd. Dit kan slechts $\mathcal{O}(n)$ keer gebeuren. Het aanpassen van de c-waarden heeft ook complexiteit $\mathcal{O}(n)$. Hieruit volgt dat de totale complexiteit van het algoritme $\mathcal{O}\{max(m\,log\,n, n^2)\}$ is. □

Opmerking:
Door een geschikte datastructuur te kiezen is er een implementatie van het algoritme van Kruskal met complexiteit $\mathcal{O}(m\,log\,n)$.[7] Voor spaarse netwerken, d.w.z. $m = \mathcal{O}(n)$ geeft dit een betere complexiteit.

Vraag 1.16
Stel dat $l(e) < l(f)$ voor alle $f \in E \backslash \{e\}$.
Bewijs dat e in iedere opspannende boom met minimale lengte zit.

Vraag 1.17
Beschouw een netwerk met 10 knooppunten, waarvan de lengtes in onderstaande tabel staan (een - betekent dat er geen verbinding is).

	1	2	3	4	5	6	7	8	9	10
1	-	18	-	4	11	-	-	-	-	-
2	18	-	17	-	20	16	-	-	-	-
3	-	17	-	-	-	15	12	-	-	-
4	4	-	-	-	19	-	-	10	-	-
5	11	20	-	19	-	7	-	8	13	-
6	-	16	15	-	7	-	14	-	5	2
7	-	-	12	-	-	14	-	-	-	9
8	-	-	-	10	8	-	-	-	3	-
9	-	-	-	-	13	5	-	3	-	6
10	-	-	-	-	-	2	9	-	6	-

a. Bepaal een minimale opspannende boom met de methode van Prim.
b. Bepaal een minimale opspannende boom met de methode van Kruskal.

1.2.7 Opgaven

Opgave 1.4
Een samenhangende graaf $G = (V, E)$ met n knooppunten heet *éénkringig* als G precies één kring bevat.

[7] Zie hiervoor paragraaf 23.2 in: T.H. Cormen, C.E. Leiserson , R.L. Rovest and C. Stein: *Introduction to algorithms*, MIT Press, Cambridge (2001).

Toon aan dat de volgende uitspraken equivalent zijn:
1. G is éénkringig.
2. $G\backslash e$ is een opspannende boom voor een geschikt gekozen tak e.
3. G is samenhangend en heeft n takken.
4. G is samenhangend en de verz. $C = \{e \in E \mid G\backslash e \text{ is samenhangend}\}$ is een kring.

Opgave 1.5
Laten T_1 en T_2 twee opspannende bomen van G zijn. Bewijs dat als e een tak is van T_1, er een tak f van T_2 is zdd. $T_1 - \{e\} + \{f\}$ ook een opspannende boom van G is.

Opgave 1.6
Zij n_k het aantal knooppunten met graad k in een boom T die minstens twee knooppunten bevat.

Bewijs dat $n_1 \geq n_3 + 2$.

Opgave 1.7
a. Gebruik de gewichten 2, 3, 5, 10 en 10 om te laten zien dat de hoogte van een Huffman boom niet uniek is.
b. Hoe kan het algoritme worden aangepast om een Huffman boom met minimale hoogte te krijgen?

Opgave 1.8
Zij G een niet-gerichte samenhangende graaf en T de opspannende gerichte boom bestaande uit de pijlen van F, verkregen met DFS. Laat G_1 een deelgraaf van G zijn die volledig is.
a. Toon aan dat er in T een pad is waar alle knooppunten van G_1 op liggen.
b. Ga na of op dit pad ook andere knooppunten kunnen liggen.

Opgave 1.9
a. Bewijs dat een niet-gerichte graaf bipartiet is d.e.s.d. als iedere kring een even aantal takken heeft.
b. Stel een $\mathcal{O}(p)$ algoritme op, waarbij $p = max(n, m)$, om na te gaan of een niet-gerichte graaf bipartiet.

Opgave 1.10
Zij G een niet-gerichte graaf en T het opspannende gerichte bos bestaande uit de pijlen van F, verkregen met DFS. Zij $T[v]$ het aantal knooppunten dat in T vanuit v kan worden bereikt, inclusief v zelf.
a. Toon aan dat w in T vanuit v bereikbaar is d.e.s.d. als $N[v] \leq N[w] < N[v] + T[v]$.
b. Pas algoritme 1.3 aan zdd. daarmee ook de getallen $T[v],\ v \in V$, worden berekend.

Opgave 1.11
Beschouw een gerichte graaf $G = (V, A)$.

a. Toon aan dat G geen ronde bevat d.e.s.d. als de verz. D uit DFS leeg is.

b. Veronderstel dat $D = \emptyset$. Toon aan dat $(v, w) \in A$ d.e.s.d. $R[w] < R[v]$.

c. Stel een $\mathcal{O}(p)$ algoritme op, waarbij $p = max(n, m)$, om de knooppunten zó te nummeren dat als $(v, w) \in A$, dan is het nummer van v kleiner dan het nummer van w, indien zo'n nummering mogelijk is, d.w.z. als G geen ronde heeft; als dit niet mogelijk is, d.w.z. G heeft wel een ronde, implementeer deze eigenschap dan in het algoritme door een variabele "ronde"de waarde 1 te geven.

Opgave 1.12
Stel een algoritme op voor Breadth-First Search in een niet-gerichte graaf.

Opgave 1.13
Pas algoritme 1.6 toe op de gerichte graaf, gerepresenteerd door de volgende lijsten:
$L[1] = \{2\}$; $L[2] = \{4, 8\}$; $L[3] = \{9\}$; $L[4] = \{6, 7\}$; $L[5] = \{3, 4\}$; $L[6] = \{4\}$; $L[7] = \{6\}$; $L[8] = \{1, 5\}$; $L[9] = \{10\}$; $L[10] = \{3, 11\}$; $L[11] = \{9\}$.

Opgave 1.14
In deze opgave bespreken we een ander algoritme om de streng samenhangende componenten van een gerichte graaf $G = (V, A)$ te bepalen. Dit algoritme gaat als volgt.

Algoritme 1.10 *Bepaling streng samenhangende componenten (versie 2)*

1. Voer DFS uit en nummer de knooppunten in de volgorde waarin ze worden afgehandeld (het eerst afgehandelde knooppunt krijgt nummer 1 etc).

2. Construeer een andere gerichte graaf $G_1 = (V, A_1)$ met $(v, w) \in A_1$ d.e.s.d. als $(w, v) \in A$.

3. Voer DFS uit in G_1, beginnend met het hoogst genummerde knooppunt (volgens de in stap 1 aangebrachte nummering). Indien met een nieuwe boom moet worden gestart, dan beginnen we weer met het nog niet bezochte knooppunt met het hoogste nummer.

4. Neem als s.s.c. de samenhangende componenten die in stap 3 worden gevonden.

Beantwoord de volgende vragen over dit algoritme:

a. Pas het algoritme toe op Voorbeeld 1.8.

b. Bepaal de complexiteit van de methode.

c. Toon de correctheid van het algoritme aan.

Opgave 1.15
Een bank heeft 5 bijkantoren die elk over een computer terminal beschikken, die verbonden moet worden met de centrale computer in het hoofdkantoor. Deze telecommunicatie geschiedt met speciale telefoonlijnen. Een bijkantoor hoeft niet rechtstreeks met het hoofdkantoor verbonden te zijn, de verbinding kan ook via andere bijkantoren lopen.
De kosten van de telefoonlijnen zijn recht evenredig met de lengte van de lijnen. De afstanden tussen de verschillende kantoren staan in onderstaande tabel.

	hoofdkantoor	bijkantoor 1	bijkantoor 2	bijkantoor 3	bijkantoor 4	bijkantoor 5
hoofdkantoor	-	160	270	115	70	190
bijkantoor 1	160	-	310	80	220	50
bijkantoor 2	270	310	-	175	120	215
bijkantoor 3	115	80	175	-	140	240
bijkantoor 4	70	220	120	140	-	100
bijkantoor 5	190	50	215	240	100	-

Het probleem is welke verbindingen er gemaakt moeten worden om de kosten te minimaliseren. Los dit probleem op.

Opgave 1.16 Een vertrouwelijke boodschap moet door 7 mensen gelezen worden. De boodschap wordt met kans p_{ij} onderschept, als deze regelrecht van persoon i naar j of omgekeerd wordt verstuurd. Hoe sturen we die boodschap rond met een minimale kans dat hij onderschept wordt? De kansen staan in onderstaande tabel gegeven:

	1	2	3	4	5	6	7
1	-	0,05	0,07	0,03	0,08	0,02	0,05
2		-	0,09	0,04	0,05	0,05	0,04
3			-	0,07	0,04	0,08	0,06
4				-	0,06	0,05	0,05
5					-	0,07	0,03
6						-	0,06
7							-

Opgave 1.17
Neem een tweetal knooppunten, zeg s en t, van een graaf $G = (V, E)$ met n knooppunten en een lengtefunctie op de takken. Voor een keten K van s naar t is de *waarde van* K het maximum van de lengtes op deze keten.
Het *minmax-probleem* is: bepaal een keten van s naar t met minimale waarde.
Stel een $\mathcal{O}(n^2)$ algoritme op om dit probleem op te lossen en bewijs de correctheid van dit algoritme.

Opgave 1.18

De *boomgraaf* $T(G)$ van een samenhangende graaf G heeft als knooppunten de opspannende bomen van G. Twee knooppunten van $T(G)$ worden verbonden als de bijbehorende opspannende bomen $n-2$ takken gemeen hebben.

a. Toon aan dat voor iedere lengtefunctie l de deelgraaf van $T(G)$ die behoort bij de minimale opspannende bomen samenhangend is.

b. Toon aan dat $T(G)$ samenhangend is.

Opgave 1.19

Het onderstaande algoritme[8] begint met een bos T met alleen n losse knooppunten, d.w.z. er zijn n componenten. In ieder knooppunt wordt de tak met de kortste lengte gekozen. Deze takken worden aan T toegevoegd. Dit herhaalt zich waarbij in plaats van knooppunten componenten moeten worden genomen en voor iedere component de kortste tak wordt gekozen met precies één uiteinde in die component.

Algoritme 1.11 *Opspannende boom met minimale lengte (Boruvka)*

1. $T = (V, \emptyset)$; $p = n$.
2. Zolang $p > 1$:
   (i) Voor iedere deelboom T_k $(1 \leq k \leq p)$ van T: bepaal de kortste tak met precies één uiteinde in T_k, zeg e_k, $1 \leq k \leq p$.
   (ii) Voor $k = 1, 2, \ldots, p$: als $e_k \notin T$: $T := T \cup e_k$ en $p := p - 1$.

a. Voer bovenstaand algoritme uit op de graaf uit Voorbeeld 1.12.

b. Toon aan dat als alle lengtes verschillend zijn in T geen kringen voorkomen en het algoritme correct is.

c. Geef een voorbeeld waaruit blijkt dat als de lengtes niet verschillend zijn er een kring kan voorkomen en het algoritme dus niet correct is.

d. Toon aan dat er hoogstens $log_2 n$ iteraties (het uitvoeren van stap 2) zijn.

e. Toon aan dat iedere iteratie met complexiteit $\mathcal{O}(m)$ kan worden uitgevoerd, zodat de totale complexiteit $\mathcal{O}(m \, log_2 n)$ is.

Opgave 1.20

Beschouw het volgende *gerandomiseerde algoritme* dat de takken van de graaf met rood en blauw gaat kleuren. Aan het einde van het algoritme vormen de blauwe takken een opspannende boom met minimale lengte.

Algoritme 1.12 *Opspannende boom met minimale lengte (Kleuralgoritme)*

Herhaal totdat $n-1$ takken blauw gekleurd zijn:

[8] Dit algoritme stamt al uit 1926 en is ontwikkeld door O. Boruvka (het artikel is in het Tjechisch geschreven).

1. Kies random of wordt verder gegaan met stap 2 of stap 3.

2. Neem een snede S zonder blauwe takken, kies een tak uit S met minimaal gewicht, kleur deze blauw en ga naar stap 1.

3. Neem een kring C zonder rode takken, kies een tak uit C met maximaal gewicht, kleur deze rood en ga naar stap 1.

a. Voer bovenstaand algoritme uit op de graaf uit Voorbeeld 1.12 en kies daarbij om en om stap 2 en stap 3.

b. Bewijs dat het algoritme een opspannende boom met minimale lengte oplevert.

c. Laat zien dat een variant van bovenstaand algoritme het algoritme van Prim is als we iedere keer stap 2 kiezen met als snede S de takken die grenzen aan een blauw gekleurde tak (start met voor S de takken die knooppunt 1 met de rest van de graaf verbinden).

d. Laat zien dat een variant van bovenstaand algoritme het algoritme van Kruskal is als we het volgende doen, nadat de takken in niet-dalende volgorde zijn genummerd:
neem de volgende tak; als beide uiteinden blauw in dezelfde blauwe deelboom zitten:
voer stap 3 uit met voor C de kring die door het toevoegen van de laatste tak ontstaat;
anders voeren we stap 2 uit met voor S de takken tussen de twee deelbomen die door de tak worden verbonden.

e. Laat zien dat een variant van bovenstaand algoritme het algoritme van Boruvka is als we het volgende doen: voer stap 2 uit op de snedes die iedere blauwe boom met de rest van de graaf verbinden (start met als blauwe deelbomen alle afzonderlijke knooppunten).

1.3 Euler en Hamilton grafen

1.3.1 Euler grafen

Niet-gerichte Euler grafen

Königsberger bruggenprobleem

Door de voormalige Pruisische stad Königsberg stroomt de rivier de Pregel. De oevers van deze rivier en twee eilandjes in de rivier zijn verbonden door zeven bruggen (zie onderstaande figuur). Toen de beroemde Zwitserse wiskundige Euler in 1736 daar op bezoek was werd hem de vraag gesteld of het mogelijk was een wandeling te maken waarbij iedere brug precies één keer wordt aangedaan en bij het beginpunt terugkeert.

Door elke rivieroever en elk eiland te representeren door een knooppunt en elke brug door een tak kunnen we er een probleem in een niet-gerichte graaf van maken. De vraag is dan of de graaf een Euler graaf is. Euler[9] wist op deze wijze het probleem op te lossen.

[9] L. Euler: *Solutio problematis ad geometriam situs pertinentis*, Comment. Acad. Sci. Petropol. 8 (1736) 128-140. Zie ook http://turnbull.mcs.st-and.ac.uk/~history/Mathematicians/Euler.html