

# Vorbereidend materiaal Wiskundetoernooi 2009: Grafentheorie

Dit jaar is *Grafentheorie* het thema van de middagwedstrijd *Sum of Us* van het Wiskundetoernooi. Dit boekje bevat het voorbereidend materiaal dat je nodig hebt om met succes aan dit onderdeel van het toernooi mee te doen.

Tijdens het toernooi treed je samen met je team op als een onderzoeksteam van *Het Spoor* en mogen jullie een aantal knelpunten oplossen rondom het spoorwegnetwerk. De knelpunten zijn allemaal op te lossen met behulp van grafentheorie, al is dit op het eerste gezicht misschien niet altijd even duidelijk.

Het eerste artikel over grafentheorie werd in 1736 geschreven door de Zwitserse wiskundige Leonhard Euler. Dit jaar vieren we het vijftigjarig bestaan van een belangrijke Nederlandse bijdrage aan de grafentheorie: het Dijkstra Algoritme (zie Hoofdstuk 3) bedacht door Edsger Dijkstra (1930-2002), die op het voorblad is afgebeeld.

Deze pagina's zijn in eerste instantie bedoeld om goed voor de dag te komen op het Wiskundetoernooi, waar vooral de eer en de spectaculaire prijzen op het spel staan. Het voorbereidend materiaal geeft je echter ook een goed beeld van wat je met wiskunde - in dit geval grafentheorie - allemaal kunt doen. Zo zul je in de voorbeelden zien dat veel dagelijkse problemen met behulp van grafentheorie kunnen worden opgelost.

Elk hoofdstuk behandelt een onderwerp binnen de grafentheorie. Nieuwe begrippen, algoritmen en methoden worden uitgelegd aan de hand van voorbeelden. Je kunt met vragen over dit voorbereidend materiaal bij ons terecht via [toernooi@math.ru.nl](mailto:toernooi@math.ru.nl).

Meer over grafentheorie kun je vinden in:

*Grafen: kleuren en routeren* door Alexander Schrijver  
<http://homepages.cwi.nl/~lex/files/graphs1.3.pdf>

*Grafen in de praktijk* door Hajo Broersma  
Zebra reeks, Epsilon uitgaven, ISBN:90-5041-078-2

*Postbodes, handelsreizigers en kortste paden* door Alex van den Brandhof  
Pythagoras, nummer 6 juni 2009, pg 6-11



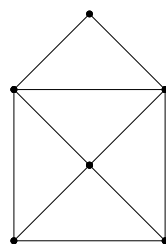
# Inhoudsopgave

<b>1</b>	<b>Eulergrafen</b>	<b>3</b>
<b>2</b>	<b>Kleuringsen</b>	<b>6</b>
<b>3</b>	<b>Kortste paden</b>	<b>10</b>
<b>4</b>	<b>Het Chinese postbodeprobleem</b>	<b>13</b>
<b>5</b>	<b>Minimale opspannende bomen</b>	<b>15</b>
<b>6</b>	<b>Stromen</b>	<b>18</b>



# Eulergrafen

Een bekend puzzeltje is het "kruishuis", ook wel de "geopende envelop" genoemd (zie hieronder). De bedoeling bij deze puzzel is om het plaatje te tekenen zonder je potlood van het papier te halen en zonder zijdes dubbel te doorlopen. Probeer het maar (er zijn meerdere oplossingen mogelijk).



Het kruishuis

Deze figuur is een voorbeeld van een **graaf**. Een graaf bestaat uit punten waarvan sommige verbonden zijn door lijnen. De punten in een graaf noemen we **knoep** en de verbindingslijnen heten **kanten**. De **graad** van een knoop is het aantal kanten dat deze knoop als uiteinde heeft. In het kruishuis bijvoorbeeld heeft de knoop linksonder graad 3 en de knoop in de top graad 2.

Een **pad** tussen twee knopen is een aaneenschakeling van kanten beginnend bij de ene knoop en eindigend in de andere. Hierbij mogen begin- en eindpunt hetzelfde zijn. We noemen een graaf **samenhangend** als er vanuit elke knoop in de graaf een pad bestaat naar elke andere knoop. We zullen alleen met samenhangende grafen werken.

Een **Eulerpad** in een graaf is een pad dat elke kant van de graaf precies één keer doorloopt.

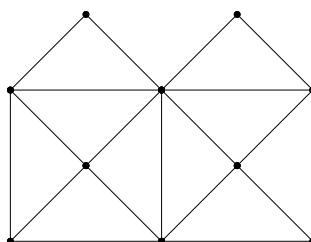
We noemen een graaf waarin een Eulerpad bestaat een **Eulergraaf**. Hierboven heb je gezien dat het kruishuis een Eulergraaf is.

Leonhard Euler (1707-1783) was een Zwitserse wis- en natuurkundige. Euler loste in 1736 het Königsberger zevenbruggenprobleem op. Dit is een bekend probleem uit de grafentheorie.



L. Euler

**Opgave 1.1** Is het "dubbele kruishuis" (zie hieronder) een Eulergraaf?



Het dubbele kruishuis

Als er te veel knopen zijn waarvan de graad oneven is, dan kun je geen Eulerpad vinden. Laten we hier eens wat beter naar kijken. Merk allereerst op dat het aantal knopen van oneven graad altijd even is (zie je waarom?). De volgende stelling vertelt je wanneer een graaf een Eulergraaf is. Ook staat er in het bewijs bij onderdeel ii) een algoritme (stappenplan) om in een graaf een Eulerpad te vinden.

### Stelling 1.1

- i) Een Eulergraaf heeft nul of twee knopen van oneven graad.
- ii) Een samenhangende graaf met nul of twee knopen van oneven graad is een Eulergraaf.

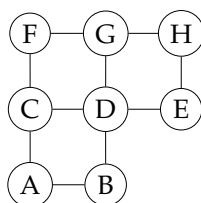
#### Bewijs:

- i) In een Eulergraaf bestaat een Eulerpad. Dit pad gaat precies één keer door alle kanten van de graaf. Kijkend naar de knopen die niet aan de uiteinden van dit pad zitten, zien we dat iedere keer als het pad via een kant een knoop binnenkomt, het via een nog ongebruikte kant de knoop weer verlaat. Dus voor deze knopen geldt dat zij een even graad hebben. Alleen de twee eindpunten hebben eventueel oneven graad.
- ii) Voor een graaf met nul of twee knopen van oneven graad bestaat een algoritme om een Eulerpad in de graaf te vinden. Voor een graaf met twee knopen van oneven graad werkt dit algoritme in grote lijnen als volgt:
  1. Vind de twee knopen van oneven graad.
  2. Bepaal een pad tussen deze twee knopen.
  3. Indien het pad nog niet alle kanten doorloopt, voeg dan net zo lang omwegen in totdat alle kanten precies één keer in het pad voorkomen.

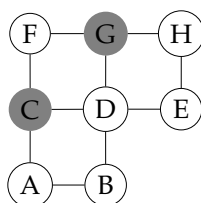
We illustreren het algoritme aan de hand van een voorbeeld.

#### Voorbeeld 1.1

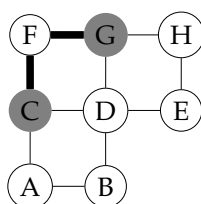
Bekijk onderstaande graaf.



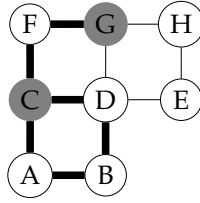
Stap 1: De knopen van oneven graad zijn knoop C en G.



Stap 2: Een pad tussen C en G is CFG.



Stap 3: De kanten AB, BD, DC, CA, DE, EH, HG en GD zijn nog niet doorlopen. We maken eerst de omweg CABDC en voegen die aan ons pad toe. We krijgen dan: CABDCFG



Nu zijn de kanten DE, EH, HG en GD nog niet doorlopen. We maken de omweg GDEHG en voegen die toe aan ons pad; dit levert CABDCFGDEHG. Nu hebben we precies alle kanten eenmaal doorlopen. Dit algoritme werkt altijd. Zie je waarom?

**Opgave 1.2** *Bewijs dat er een Eulerpad bestaat in een graaf zonder knopen van oneven graad.*

Als het goed is heb je ontdekt dat in een graaf zonder knopen van oneven graad, de begin- en eindknoop van het Eulerpad hetzelfde zijn. Dit is een bijzonder soort pad.

**Definitie 1.1** *Een pad waarvan begin- en eindknoop hetzelfde zijn en dat géén kant twee keer doorloopt noemen we een **cykel**. (Uitzondering: een losse knoop is geen cykel.)*

In het geval van een Eulergraaf zonder knopen van oneven graad spreken we dan ook van een **Eulercykel**.

# 2

## Kleuringen

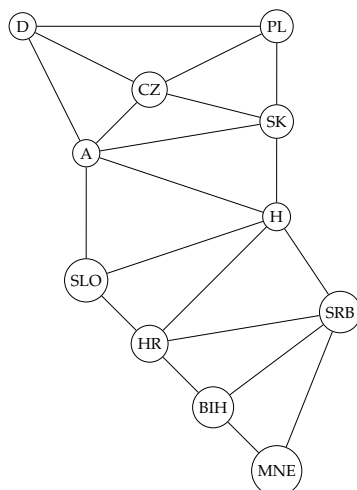
In een atlas zijn buurlanden altijd verschillend gekleurd. Hoeveel kleuren heb je nodig om dat bij elke kaart voor elkaar te krijgen?

**Opgave 2.1** Hieronder zie je de kaart van een deel van centraal Europa. Kleur de kaart met zo min mogelijk kleuren, zó dat twee aan elkaar grenzende landen verschillende kleuren hebben.



Centraal Europa

Laten we dit kleurprobleem eens wiskundig bekijken. Allereerst maken we een graaf bij de kaart. De knopen zijn de landen. Als twee landen aan elkaar grenzen, worden de bijbehorende knopen verbonden door een kant, anders niet.



De vertaling van het kleurprobleem is dan: kleur de knopen van de graaf zo dat twee knopen die met elkaar verbonden zijn verschillende kleuren krijgen. Gebruik hiervoor zo min mogelijk kleuren.



Er is helaas geen efficiënt algoritme bekend om dit voor een willekeurige graaf te doen met het minimale aantal kleuren. Wel zijn er snelle algoritmen die vaak, maar niet altijd, een minimale kleuring vinden, bijvoorbeeld het volgende algoritme:

- Nummer de knopen op volgorde van hun graad; de knoop met de hoogste graad eerst. Voor knopen met gelijke graad kies je een willekeurige onderlinge volgorde.
- Kleur de knopen in de volgorde die je zojuist hebt gemaakt. En doe dat zo zuinig mogelijk. Als knoop nummer 2 niet verbonden is met knoop nummer 1, dan krijgt hij dezelfde kleur als knoop nummer 1. Als knoop 3 wel met 2, maar niet met 1 verbonden is, dan krijgt hij dezelfde kleur als 1, behalve als knoop 2 al dezelfde kleur had als 1; dan krijgt 3 een nieuwe kleur. Enzovoort...

**Opgave 2.2** Kleur de voorgaande graaf met dit algoritme.

#### Over het vierkleurenprobleem

Het kleuren van een landkaart heeft wiskundigen lange tijd beziggehouden. In 1852 stelde Francis Guthrie zich de vraag of elke landkaart met vier kleuren zó ingekleurd kan worden, dat landen die aan elkaar grenzen verschillende kleuren krijgen. (Als twee landen slechts één grenspunt gemeen hebben, grenzen ze niet aan elkaar.) Velen hebben geprobeerd te bewijzen dat dit inderdaad altijd met vier kleuren kan; soms werd pas jaren later ontdekt dat zo'n bewijs niet helemaal deugde. Pas 125 jaar later is door Appel en Haken, met behulp van een computer, bewezen dat vier kleuren inderdaad volstaan.

Voor het vinden van een kleuring van een landkaart hoef je niet per se eerst een graaf te maken. Je kunt even goed de kleuring direct uitvoeren. Het maken van een graaf kan wel helpen om het probleem overzichtelijk weer te geven.

Bij de volgende opgave is het nuttig om het probleem eerst te vertalen naar een graaf en vervolgens de knopen te kleuren. Bedenk eerst wat de knopen en kanten van de graaf moeten worden en daarna wat de kleuring betekent.

**Opgave 2.3** De stichting Educatie verzorgt op dinsdag, woensdag, donderdag en vrijdag cursussen Wiskunde, Natuurkunde, Aardrijkskunde, Geschiedenis, Engels, Frans en Duits. Elke cursus duurt een volle dag. Er zijn drie docenten: één geeft zowel Wis- als Natuurkunde, een ander geeft Geschiedenis en Aardrijkskunde, en de derde geeft de moderne vreemde talen. Sommige cursisten willen twee vakken volgen. Zo zijn er cursisten die Wiskunde willen combineren met Frans, of met Aardrijkskunde, of met Geschiedenis. Verder zijn er bij elke taal cursisten die ook Aardrijkskunde of Geschiedenis willen volgen. Tot slot zijn er nog cursisten die naast Natuurkunde ook Engels willen leren. Andere combinaties komen niet voor.

Lukt het de zeven vakken in vier dagen te geven, zo dat alle cursisten die twee vakken willen volgen dat ook kunnen?

We bekijken nog een ander roosterprobleem dat je met kleuringen kunt oplossen.

Er zijn zes docenten: A t/m F, zes klassen: I t/m VI en drie lesuren: 1, 2 en 3. In onderstaand schema geven de lege hokjes aan wie welke klassen moet lesgeven, bijvoorbeeld klas I krijgt les van docent C, D en F. Door in elk leeg hokje één van de cijfers 1, 2 of 3 in te vullen ontstaat een lesrooster. Het cijfer 2 in hokje C-V zou betekenen dat docent C in lesuur 2 lesgeeft aan klas V.

	A	B	C	D	E	F
I	×	×			×	
II		×	×			×
III	×				×	×
IV			×	×	×	
V	×	×		×		
VI			×	×		×

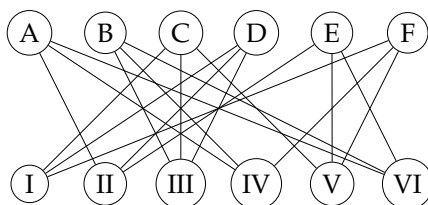
Er zijn een paar voorwaarden:

1. Een docent kan maar aan één klas tegelijk lesgeven.
2. Een klas kan maar van één docent tegelijk les krijgen.

**Opgave 2.4** Wat betekenen deze voorwaarden voor de kolommen en rijen van het schema? Probeer een passend lesrooster te maken.

Zoals je waarschijnlijk hebt gemerkt is het vinden van een passend rooster niet gemakkelijk als je zomaar wat probeert. Gelukkig bestaat er een algoritme dat je kan helpen.

Om dit algoritme uit te leggen, is het handig het roosterprobleem te vertalen naar een graaf:



De knopen zijn de leraren en de klassen; de kanten zijn de mogelijke combinaties. Zo geeft docent A les aan klas II, IV en VI en is knoop A dus ook met de knopen II, IV en VI verbonden. Merk op dat de docenten onderling niet verbonden zijn en de klassen evenmin. We noemen dit een **bipartiete graaf** (de knopen zijn verdeeld in *twee parten*).

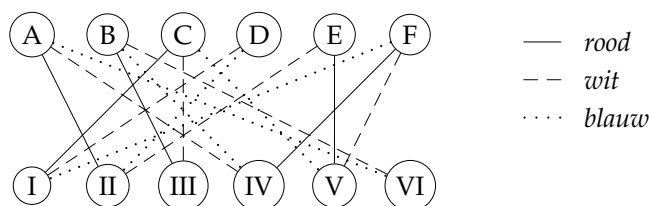
We gaan de *kanten* kleuren. Elke kleur staat voor één van de lessen 1, 2 of 3, als blauw staat voor lesuur 3 en de kant die I met C verbindt is blauw betekent dit dus dat klas I het lesuur 3 les krijgt van docent C.

**Opgave 2.5** Wat betekenen de voorwaarden 1. en 2. voor de kleuring van de kanten?

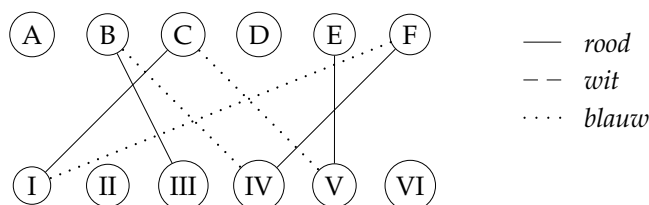
De vraag is of we bovenstaande graaf kunnen kantkleuren met drie kleuren, zodanig dat aan voorwaarden 1. en 2. is voldaan. Het antwoord wordt gegeven door de Stelling van König, die zegt dat een bipartiete graaf met maximale graad  $k$  kantkleurbaar is met  $k$  kleuren. De graad van elke knoop in ons probleem is drie en dus volgt uit de Stelling van König dat er een oplossing bestaat voor ons probleem.

Het volgende algoritme laat zien hoe je die oplossing kunt vinden.

We beginnen gewoon op hoop van zegen te kleuren, rekening houdend met de twee voorwaarden en dan komen we een heel eind. In onderstaande poging bleven slechts twee kanten over die niet gekleurd konden worden: D-III en E-VI.



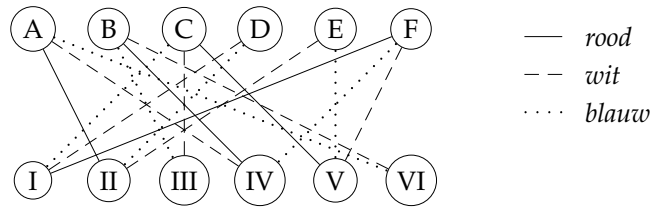
We gaan zorgen dat ook E-VI gekleurd wordt. In E ontbreekt de kleur blauw en in VI de kleur rood. We maken nu een zo lang mogelijk pad, te beginnen bij E, van afwisselend de kleuren rood en blauw. Dat pad staat hieronder:



Omdat de graaf bipartiet is gaat dit pad op en neer tussen docenten en klassen. Hieruit volgt dat het pad niet in VI eindigt, of door VI gaat. Zie je waarom?

In dit pad verwisselen we de kleuren rood en blauw. In een tussenknoop van het pad waren de kleuren rood en blauw aanwezig en dat zijn ze na de verwisseling nog steeds. In de eindknoop III was er een rode kant, en nu een blauwe. Die was er blijkbaar nog niet anders was dit niet het langste pad!

Als we de kleurwisseling van dit pad in de graaf uitvoeren, krijgen we:



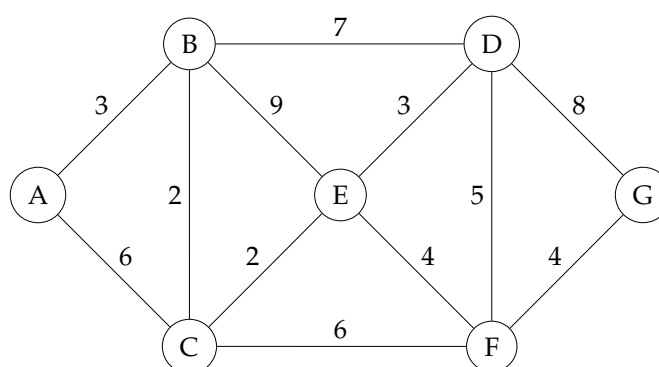
In knoop E ontbreekt nu de kleur rood, die ontbrak ook in VI, dus de kant E-VI kan nu rood gekleurd worden.

Het probleem met de andere kant die we nog niet gekleurd hadden (D-III) is en passant ook meteen opgelost. Want in zowel knoop D, als in knoop III ontbreekt nu de kleur rood. Was dit niet het geval geweest dan hadden we het algoritme moeten herhalen.

Je kunt het algoritme van König ook vertalen en direct op het lesrooster toepassen. De graaf helpt om de gegevens overzichtelijk weer te geven.

## Kortste paden

Een fietskoerier moet een pakketje afleveren. Hij wil dit zo snel mogelijk doen. Hieronder is de situatie schematisch weergegeven in een graaf. De fietskoerier vertrekt vanuit het postkantoor A en moet het pakketje afleveren bij G, de knopen zijn kruispunten en de getallen langs de kanten zijn de lengtes van de wegen.



**Opgave 3.1** *Probeer een kortste route van A naar G te vinden.*

Een algoritme dat het kortste pad vindt, is het Dijkstra Algoritme. Het idee van dit algoritme is dat we de knopen labelen en daarbij telkens het kleinste label kiezen. We maken daarbij tijdelijke (lichtgrijze) en permanente (donkergrijze) labels. Een tijdelijk label geeft de kortste afstand van A tot die knoop, die we tot dan toe gevonden hebben. Bij iedere stap die we doen kan dit tijdelijke label kleiner worden. We maken een tijdelijk label permanent als we hebben vastgesteld dat er geen korter pad naar die knoop bestaat.

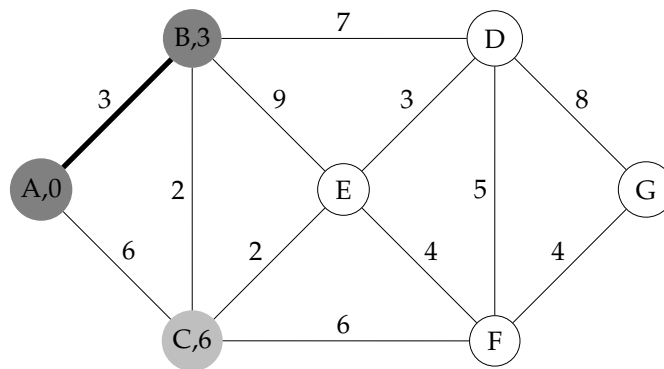
Nu het daadwerkelijke algoritme:

1. Geef knoop A het permanente label 0.
2. Bekijk alle burenen van knoop A. De **burenen** van een knoop zijn de knopen die door één kant met de betreffende knoop zijn verbonden. De burenen van A zijn dus B en C. Deze burenen krijgen een tijdelijk label: de lengte van de kant tussen A en deze buur. Dus B krijgt het label 3 en C het label 6. Kies van deze tijdelijke labels het kleinste (indien er meerdere labels het kleinst zijn, kies dan één van deze kleinste labels). In het voorbeeld is het label 3 van B het kleinst, maak dit permanent en leg de bijbehorende kant AB vast.

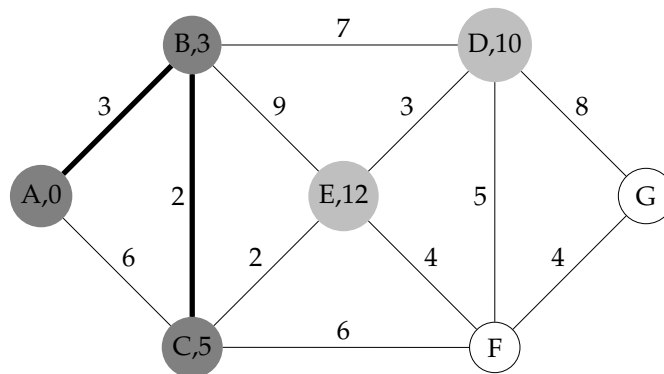
Edsger Dijkstra (1930-2002) was een Nederlandse wiskundige. Hij heeft verscheidene algoritmen in de grafentheorie bedacht.



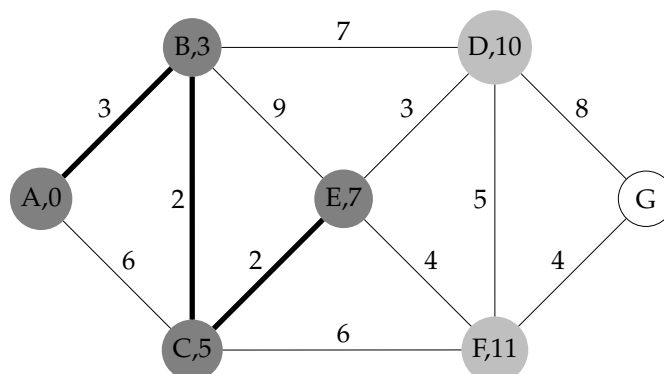
*E. Dijkstra*



3. Kijk naar de buren van B, die nog geen permanent label hebben (C,D en E). Geef deze nieuwe tijdelijke labels. Als een buur nog geen label heeft, dan wordt het tijdelijke label de som van het label van B (3) en de lengte van de kant tussen deze buur en B. Als een buur al een label heeft dan wordt het nieuwe tijdelijk label het minimum van deze som en het oude tijdelijke label. Voor C geldt dus dat het nieuwe tijdelijke label 5 wordt. De knopen die buur zijn van A, maar niet van B (in dit geval zijn dat er geen) behouden hun tijdelijke label. Maak weer het kleinste tijdelijke label (in de h le graaf) en de bijbehorende kant permanent, in dit geval heeft C het kleinste tijdelijke label en de bijbehorende kant is BC.



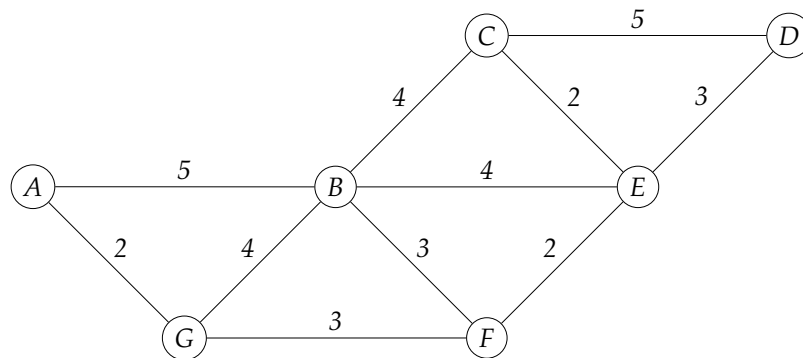
Herhaal nu de vorige stap. Geef nieuwe tijdelijke labels aan de buren van C en maak het kleinste tijdelijke label in de graaf permanent evenals de bijbehorende kant.



Herhaal stap 3 totdat de bestemmingsknoop G een permanent label heeft. Let op: bij een permanent label hoort dus een pad, dit pad is de tot dan toe kortste route tot de betreffende knoop. Het permanente label van G is nu de lengte van het kortste pad van A naar G. En het vastgelegde pad is het kortste pad. Soms zijn er meerdere oplossingen, een kortste route van A naar D is ABD met lengte 10, maar ABCED heeft ook lengte 10 en is dus ook een oplossing.

**Opgave 3.2** Vind nu zelf een kortste route van A naar G.

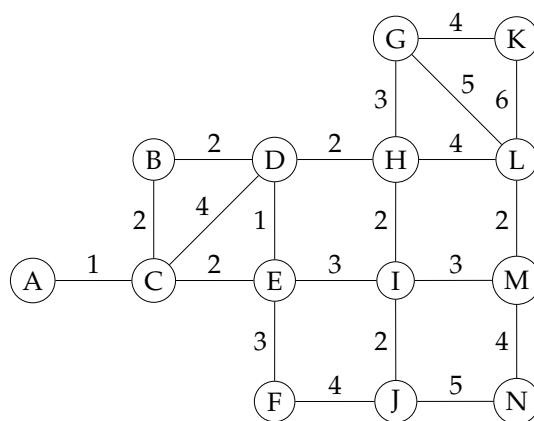
**Opgave 3.3** De fietskoerier moet vervolgens in de onderstaande graaf een pakketje van knoop A naar knoop D brengen. Wat is een kortste route en hoe lang is die?



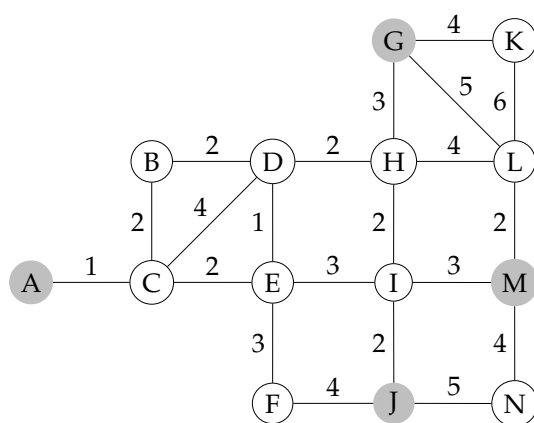
## Het Chinese postbodeprobleem

Een postbode wil zo snel mogelijk klaar zijn met zijn werk. Daarom zoekt hij de kortste route die ten minste één keer door alle straten (waar hij de post moet bezorgen) gaat en begint en eindigt op het postkantoor. Om dit probleem op te lossen combineren we de theorie uit hoofdstuk 1 (over Eulergrafen) en hoofdstuk 3 (over kortste paden).

Hieronder is de situatie schematisch weergegeven. Net als in de vorige paragraaf zijn de knopen de kruispunten en de getallen langs de kanten de lengtes van de wegen. We zoeken een **rondwandeling** in deze graaf. Dit is een pad dat alle kanten *ten minste* één keer doorloopt en waarvan begin- en eindknoop hetzelfde zijn. Elke Eulercykel is een rondwandeling, aangezien je daarbij alle zijdes *precies* één keer doorloopt. Onderstaande graaf heeft echter ook knopen van oneven graad, deze graaf bevat dus geen Eulercykel: er moeten kanten dubbel worden doorlopen in de rondwandeling. Aangezien we op zoek zijn naar een zo kort mogelijke rondwandeling, willen we de afstand die we dubbel lopen minimaliseren. Het volgende algoritme geeft een minimale rondwandeling.



1. Vind alle knopen van oneven graad. Dat zijn A, G, J en M.



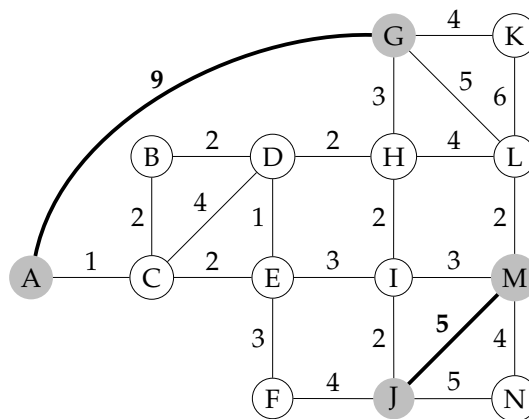
2. Bepaal (met Dijkstra's algoritme) voor elk tweetal knopen van oneven graad de lengte van de kortste pad tussen de twee knopen. Deze afstanden staan in de volgende tabel:

	A	G	J	M
A	-	9	8	9
G	9	-	7	7
J	8	7	-	5
M	9	7	5	-

3. Bekijk op welke manieren je de punten van oneven graad kunt opsplitsen in paren. Een voorbeeld van zo'n opsplitsing is AG, JM. Aangezien de afstand tussen A en G 9 en tussen J en M 5 is, zeggen we dat de lengte van deze opsplitsing 14 ( $=9+5$ ) is. De mogelijke opsplitsingen (en hun lengtes) zijn:

Opsplitsing	Lengte
AG, JM	14
AJ, GM	15
AM, GJ	16

4. Kies een opsplitsing met de kortste lengte. Dat is AG, JM.  
Voeg een kant tussen A en G en tussen J en M toe aan de graaf. Dat levert de volgende nieuwe graaf:



5. De nieuwe graaf heeft alleen knopen van even graad. Vind een Eulercykel in deze graaf. Er zijn meerdere Eulercyclen mogelijk, wij kiezen:

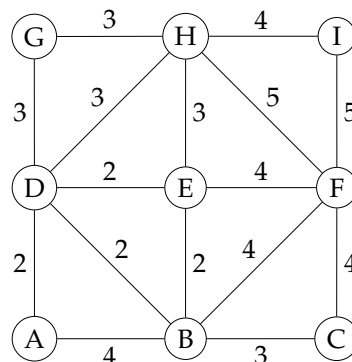
ACBDCEFJNMJIEDHIMLKGLHGA.

6. Vervang in de Eulercykel de toegevoegde kanten (MJ en GA) door een kortste pad (in de oorspronkelijke graaf) tussen de betreffende knopen. We vervangen MJ door MIJ en GA door GHDECA. Dit levert als kortste rondwandeling:

ACBDCEFJNMIJIEDHIMLKGLHGHDECA

met lengte: 64 (totale lengte oorspronkelijke graaf) + 14 (dubbel gelopen) = 78.

**Opgave 4.1** In onderstaande wijk moet de post bezorgd worden. De postbode kan beginnen waar hij wil, maar hij moet wel op dezelfde plek eindigen. Wat is de minimale afstand die de postbode moet afleggen om in de hele wijk de post te bezorgen? Welke route moet hij dan lopen?

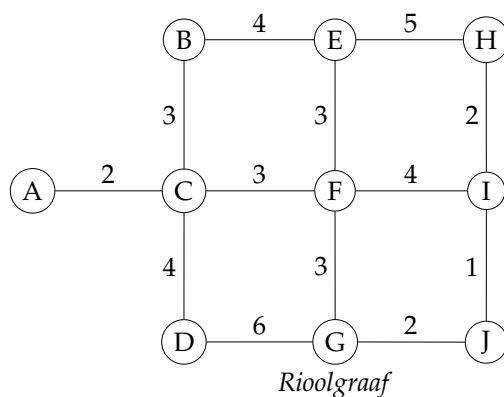




## Minimale opspannende bomen

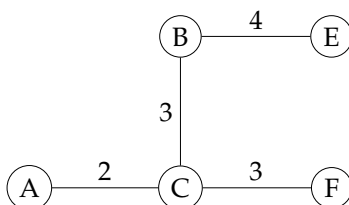
In Nijmegen wordt een nieuwe wijk aangelegd en in deze nieuwbouwwijk moet ook een rioolnetwerk komen. Een aannemersbedrijf is gevraagd om een offerte op te stellen voor dit rioolnetwerk. Het netwerk moet verbonden zijn met de centrale afvoer in de wijk en alle huizen moeten zijn aangesloten op het netwerk.

Hieronder is de situatie schematisch weergegeven met behulp van een graaf. De knopen zijn de aansluitpunten voor de huizen (knoop A is de centrale afvoer) en de kanten de mogelijke rioolbuizen met daarbij de afstanden tussen de aansluitpunten.



Om de kosten zo laag mogelijk te houden, wil het aannemersbedrijf zo min mogelijk rioolbuis gebruiken. Welke buizen moet het bedrijf leggen om dit te bereiken? Om dit probleem op te lossen zoeken we een deelgraaf van deze graaf, die aan een aantal eisen voldoet.

Een **deelgraaf** is, zoals het woord al zegt, een deel van de oorspronkelijke graaf, waarbij knopen en/of kanten zijn weggelaten (het geheel moet natuurlijk wel weer een graaf zijn). Bijvoorbeeld:



is een deelgraaf van de rioolgraaf.

De deelgraaf die we zoeken, moet aan de volgende eisen voldoen:

- De deelgraaf bevat alle knopen van de oorspronkelijke graaf.
- De deelgraaf is samenhangend.
- De deelgraaf bevat geen cyclen.

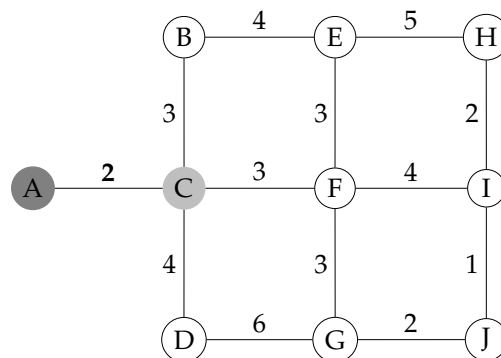
Een deelgraaf die aan deze voorwaarden voldoet, noemen we een **opspannende boom**. Om zo min mogelijk rioolbuis te gebruiken zoeken we een **minimale** opspannende boom, dit wil zeggen een opspannende boom met de kleinst mogelijke totale lengte. Het algoritme van Prim vertelt je hoe je zo'n minimale opspannende boom in een gegeven graaf kunt vinden.

Het algoritme werkt als volgt:

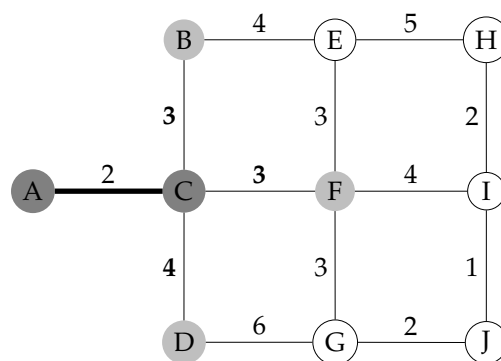
1. Kies een knoop van de graaf en markeer die.
2. Bekijk van elke gemarkeerde knopen zijn ongemarkeerde buren. Bepaal van elk van deze buren de minimale afstand, via één kant, tot een gemarkeerde knoop. Kies een ongemarkeerde knoop met de kortste minimale afstand. Markeer deze knoop en de bijbehorende kant. Herhaal deze stap totdat alle knopen gemarkeerd zijn.

Voor het rioolnetwerk gaat het algoritme als volgt:

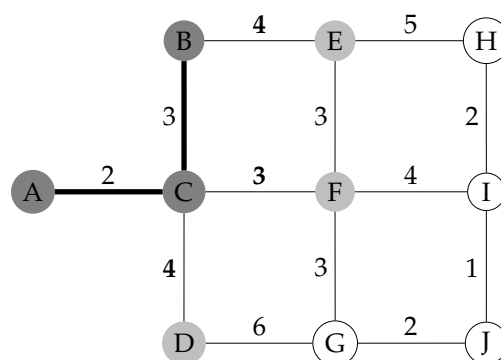
1. We kiezen als beginpunt de centrale afvoer, knoop A.



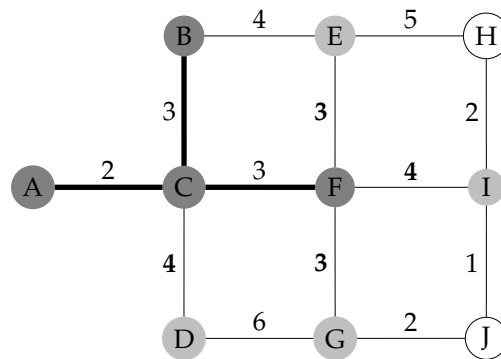
2. De enige knoop die met de centrale afvoer verbonden is, is knoop C. Die markeren we, evenals de kant tussen A en C.



2. De knopen die met een gemarkeerde knoop verbonden zijn, zijn B, D en F. Zowel B als F heeft de kortste afstand, 3, tot een gemarkeerde knoop. We kiezen ervoor om B en de kant tussen B en C te markeren.

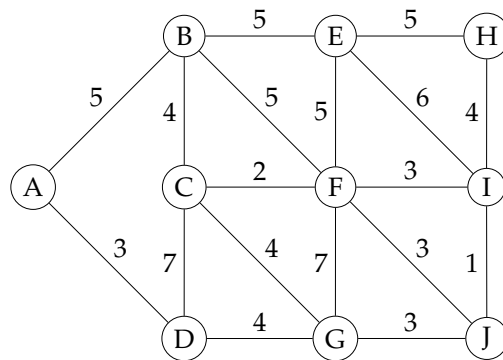


2. De knopen die met een gemarkeerde knoop verbonden zijn, zijn D, E en F. Alleen knoop F heeft afstand 3 (de minimale afstand) tot een gemarkeerde knoop. We markeren dus F en de kant tussen C en F.



**Opgave 5.1** Maak het algoritme af en vind zodoende een minimale opspannende boom. Als je het goed hebt gedaan vind je een boom van lengte 23.

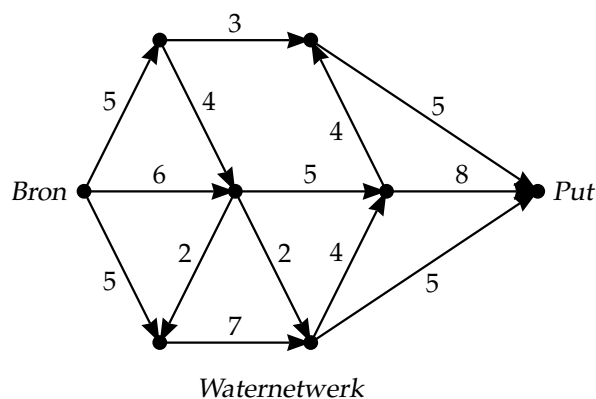
**Opgave 5.2** In de nieuwbouwwijk moeten ook telefoonlijnen komen. In onderstaande graaf zie je waar telefoonkabels kunnen komen met de bijbehorende afstanden. Vind een netwerk waarin zo min mogelijk telefoonkabel wordt gebruikt.



# 6

## Stromen

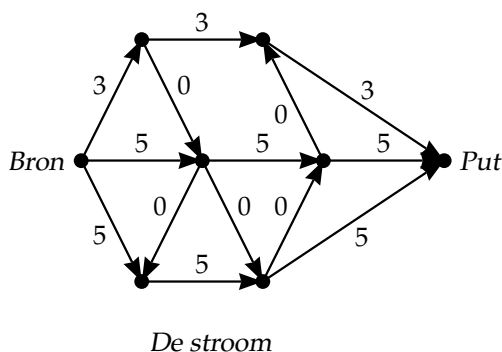
Een waterbron is via een netwerk van kanalen verbonden met een put. Van elk kanaal is de capaciteit bekend; dat is de hoeveelheid water die er hoogstens per minuut doorheen kan. Bovendien is van elk kanaal gegeven in welke richting het water stroomt. De situatie is hieronder in een graaf weergegeven. We hebben hier met een **gerichte graaf** te maken: elke kant (kanaal) heeft een richting.



Hoeveel water kan er per minuut maximaal door het netwerk stromen? De stroom moet voldoen aan de volgende eisen:

- in elk kanaal is de stroom ten hoogste de capaciteit,
- in elke knoop (behalve bron en put) is de instroom gelijk aan de uitstroom (er kan zich geen water onderweg ophopen),
- in elk kanaal moet de stroomrichting gelijk zijn aan de voorgeschreven richting.

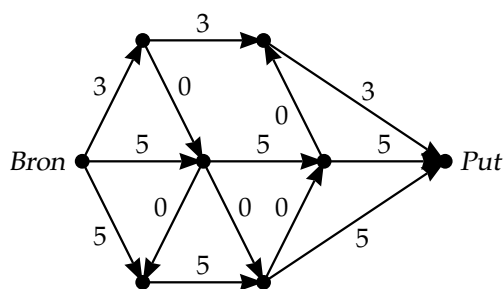
Een voorbeeld van een stroom:



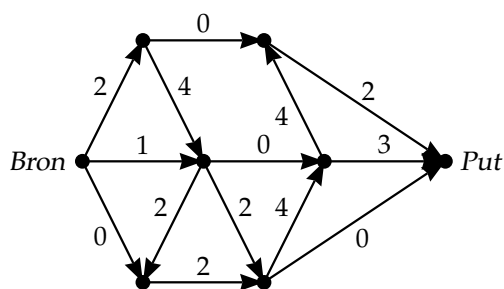
**Opgave 6.1** Ga na dat aan de eisen is voldaan.

Merk op dat de instroom van de put gelijk is aan de uitstroom van de bron. Dit volgt uit de tweede voorwaarde.

Om te bepalen hoeveel water er maximaal van bron naar put kan stromen, nemen we de stroom hierboven als uitgangspunt.

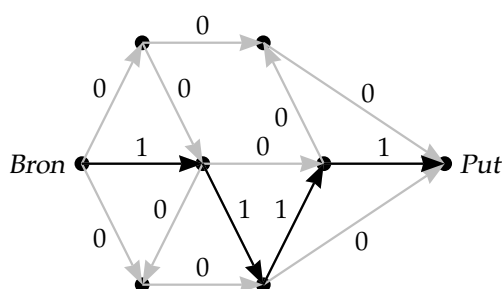


De stroom



De resterende capaciteit

Er is nog capaciteit over; zie het rechter plaatje. Je kunt die benutten door een **stroomvermeerderend** pad van bron naar put te maken, rekening houdend met de eisen. Bijvoorbeeld:



Een stroomvermeerderend pad

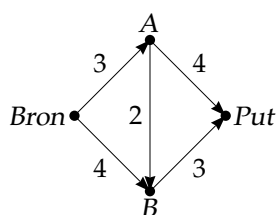
De extra stroom kun je optellen bij de eerdere stroom zonder de capaciteit te overschrijden.

**Opgave 6.2** Wat wordt de nieuwe stroom en wat is de resterende capaciteit?

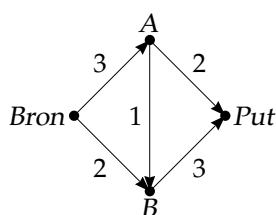
Het is duidelijk dat, wanneer je zo'n stroomvermeerderend pad kunt vinden, je de stroom in het netwerk kunt vergroten. Doe dit net zo lang totdat er geen stroomvermeerderend pad meer gevonden kan worden.

**Opgave 6.3** Vind de maximale stroom in het netwerk.

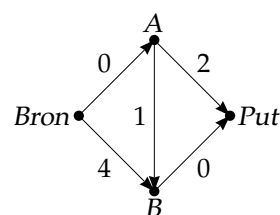
Er is een kleine complicatie waar we rekening mee moeten houden: soms kun je de stroom vermeerderen door een stroom in de verkeerde richting te verminderen! In het volgende (flauwe) voorbeeld zie je links een netwerk met de capaciteiten, in het midden een stroom ter grootte 5 die we daarin al gemaakt hebben en rechts de resterende capaciteit. Omdat de beide kanten met capaciteit 3 al 'verzadigd' zijn, kun je geen nieuwe stroom toevoegen. Toch is de maximale stroom nog niet bereikt (want je vindt makkelijk een stroom van grootte 6)!



De capaciteit



De stroom



De resterende capaciteit

Dat komt doordat de stroom ter grootte 1 door de kant A-B een slecht idee is. We kunnen de stroom groter maken door via het pad BRON-B-A-PUT een stroom ter grootte 1 te laten lopen; op kant A-B gaat dat 'tegen

de stroom in', en is het resultaat een stroom ter grootte 0. Op BRON-A en B-PUT gaat dit met de stroom mee en kunnen we gewoon 1 bij de stroom die we al hadden optellen. Dit **stroomvermeerderend pad** geeft dus extra stroom 1. Bij de kanten die in de richting van de pijl doorlopen worden kunnen we 1 optellen bij de stroom die we al hadden. en bij de kanten die tegen de pijl in doorlopen worden moeten we de stroom die we al hadden met 1 verminderen. Kortor: bij de mee-pijlen de extra stroom optellen, bij de tegen-pijlen de extra stroom aftrekken. Dat kan natuurlijk alleen als op de mee-pijlen voldoende capaciteit over is en op de tegen-pijlen voldoende stroom aanwezig is. Je hebt de maximale stroom in een graaf gevonden, als je geen stroomvermeerderend pad meer kunt vinden, dat is wanneer elk pad van BRON naar PUT ofwel een kant doorloopt in de richting van de pijl waar de stroom al net zo groot is als de capaciteit, ofwel een kant doorloopt tegen de richting van de pijl in waar de stroom al 0 is.

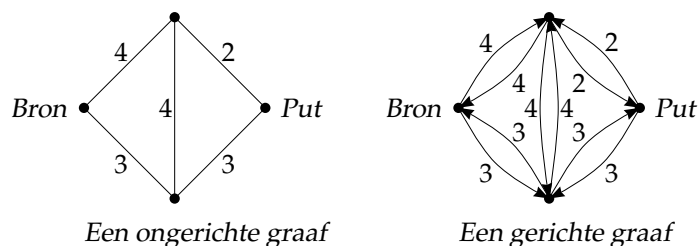
Er is nog een belangrijk en eenvoudig hulpmiddel om te zien of je klaar bent met het vinden van de maximale stroom. Het maakt gebruik van het begrip snede. Een **snede** in een netwerk is een stel kanten dat na verwijdering geen gericht pad van de BRON naar de PUT in de graaf overlaat; de grootte van zo'n snede is de som van de capaciteiten van de kanten erin. In bovenstaand voorbeeld vormen de kanten Bron-A en B-PUT samen een snede van grootte 6.

In elk netwerk geldt: *de maximale stroom is gelijk aan de minimale snede* (de 'max flow-min cut' stelling). Intuïtief is dit vrij duidelijk: als je wilt voorkomen dat er een stroom kan lopen, is het nodig en voldoende dat je een snede maakt (dus de kanten van de snede verwijdert), die de betreffende kanalen blokkeert. De minimale manier om dat te doen correspondeert precies met een maximale stroom.

**Opgave 6.4** Vind de minimale snede bij de oplossing van het eerste voorbeeld.

Wanneer je een maximale stroom hebt gevonden, zullen de kanten waar de gehele capaciteit wordt benut een snede vormen!

Ten slotte nog dit. In veel toepassingen is het helemaal niet logisch dat stromen slechts in één richting door kanten kunnen stromen. Denk bijvoorbeeld aan een systeem van buizen waar pompen in beide richtingen water kunnen pompen. Een model hiervan bestaat uit een gewone, ongerichte graaf (nog steeds met bron, put en capaciteiten). Het lijkt misschien moeilijker om hierin de maximale stroom te vinden omdat je niet van tevoren weet in welke richting stromen door kanten zullen gaan. Dit lossen we als volgt op: We vervangen elke kant in de graaf door twee gerichte kanten (één in elke richting) beide met de gegeven capaciteit. Zie het voorbeeld hieronder.



Dan krijg je een netwerk in een gerichte graaf, waarin je een maximale stroom kunt vinden. En die kun je dan terugvertalen naar de ongerichte situatie.

**Opgave 6.5** Vind de maximale stroom in de ongerichte graaf hieronder.

