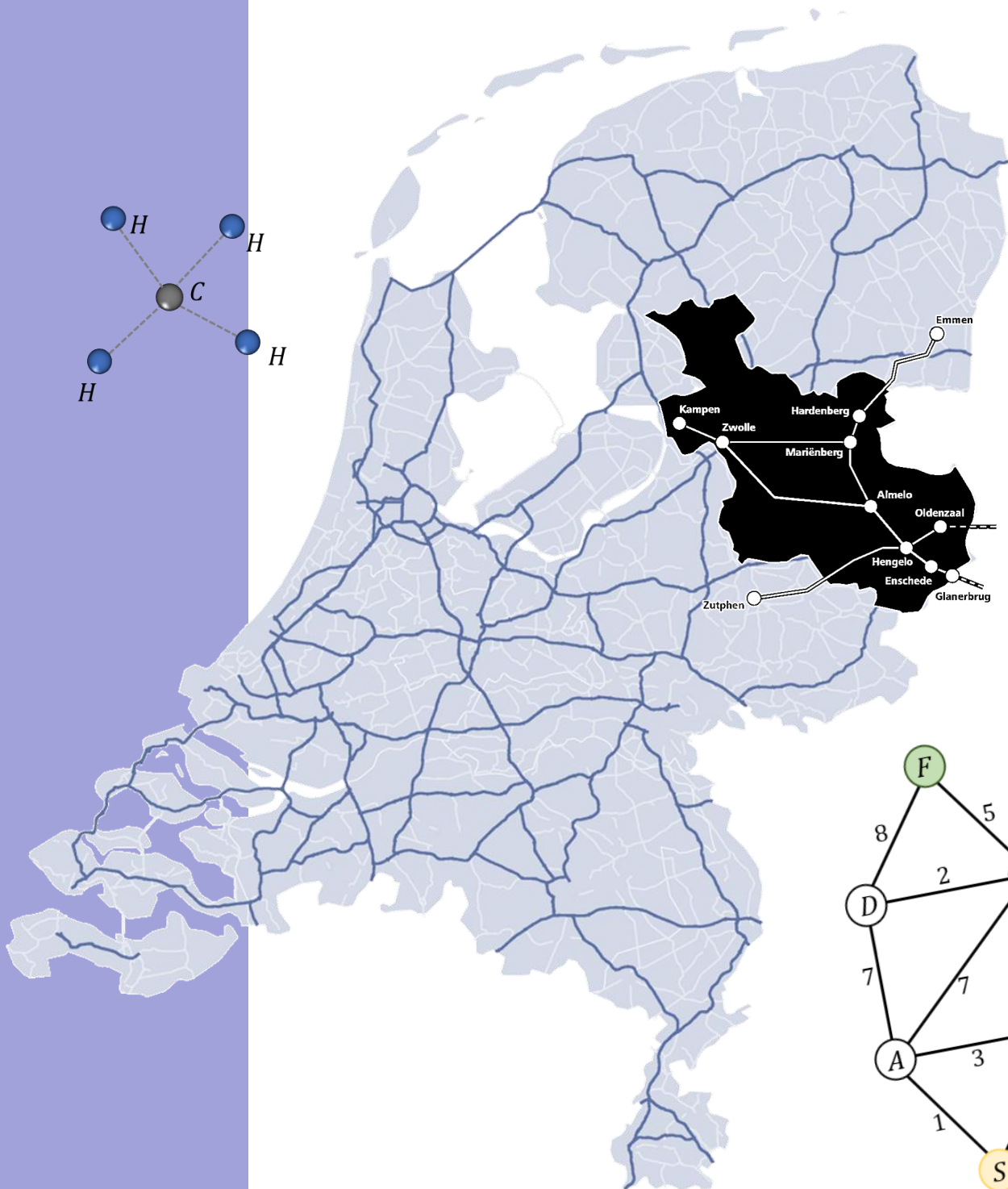


Computational thinking, grafentheorie en routes



Wat gaan we doen?

Beste leerlingen,

Voor je ligt de mini-lessenserie *Computational thinking, grafentheorie en routes*. Deze begrippen worden gedurende de lessenserie duidelijk. De mini-lessenserie bestaat uit 3 lessen waarvoor jullie in totaal 200 minuten hebben. Voor les 1 staan 50 minuten, voor les 2 staan 90 minuten, voor les 3 nog 60. Je gaat aan het werk in *tweetallen*.

Les 1 gaat over grafentheorie. Je leert hier wat een graaf is, wat je met een graaf kan en we leren hoe we een kaart kunnen modelleren als graaf. Ook ga je in deze les bezig met routes en berekeningen in grafen. In les 2 ga je bezig met algoritmes om de lengte van de kortste route tussen twee plekken te berekenen. In les 3 ga je algoritmes gebruiken om de kortste route tussen twee plekken te bepalen, zoals dat ook gedaan wordt in Google Maps. Bovendien zie je waarom algoritmes nuttig zijn.

Enkele regels:

- Je werkt in tweetallen. Overleg tussen groepjes is niet toegestaan. Je kunt wel hulp vragen aan de docent.
- De beoordeling vindt per tweetal plaats, tenzij een individuele beoordeling meer op zijn plaats lijkt. We beoordelen jullie werk op (wiskundig) niveau en op je aanpak. Het gaat dus niet alleen om het eindantwoord, maar juist ook om het proces. *Antwoorden zonder berekening of verklaring leveren dan ook weinig of zelfs geen punten op.*
- Deel de opdrachten niet op. Je zult samen door de opgaven heen moeten werken om de mini-lessenserie te kunnen volbrengen.

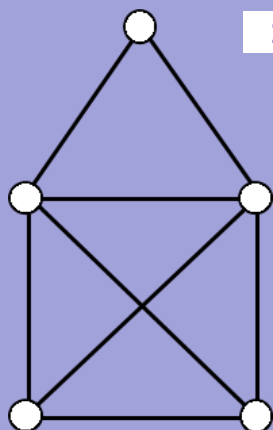
Veel plezier en succes!

Les 1

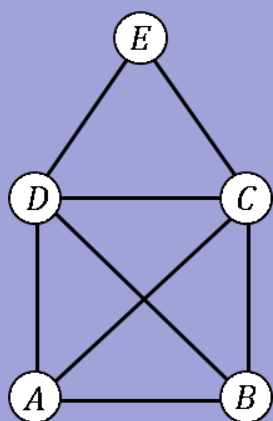
In deze les leer je wat een graaf is en bekijken we een toepassing. We beginnen met een puzzel die je misschien wel kent. In Figuur 1 is een huisje met een kruis erdoorheen getekend. De bedoeling van de puzzel is om dit plaatje te tekenen zonder je pen van het papier te halen en zonder lijnen dubbel te doorlopen.

OPDRACHT

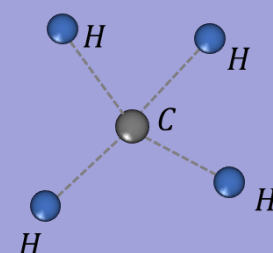
Probeer het huisje met een kruis zelf te tekenen zonder je pen van het papier te halen en zonder lijnen dubbel te doorlopen. Geef aan in welk punt je bent begonnen.



Figuur 1: het huisje met een kruis.



Figuur 2: het huisje met een kruis als graaf.



Figuur 3: structuurformule van CH_4 .

Het huisje met een kruis is een voorbeeld van een *graaf*. Een graaf bestaat uit *punten* en *links* tussen die punten. Elk punt in een graaf heeft een *graad*: het aantal links dat bij het punt aankomen. Zo heeft het punt linksonder een graad van 3 en het punt bovenin een graad van 2.

In grafen gebruiken we letters om de punten aan te geven. In Figuur 2 is het huisje met een kruis nog een keer getekend, deze keer met letters in de punten. De volgende notatie wordt gebruikt voor links. De link tussen de punten A en B geven we aan met $\{A, B\}$. Zo is de link tussen de punten E en C gegeven door $\{E, C\}$.

SAMENVATTING

Een graaf is een wiskundig object en bestaat uit:

- **Punten**, weergegeven met hoofdletters. Voorbeeld: A.
- **Links** tussen de punten. Notatie: $\{A, B\}$.

De **graad** van een punt is het aantal links dat bij het punt aankomt.

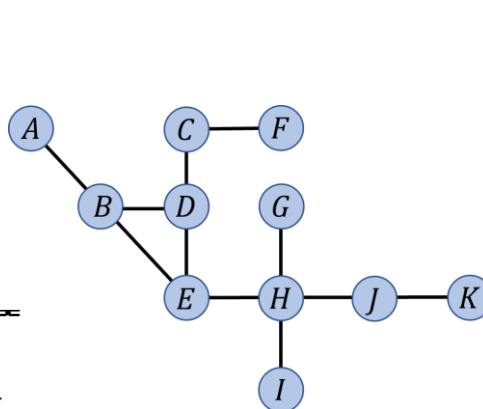
Grafen hebben verschillende toepassingen. Eén toepassing ken je misschien al wel als je scheikunde hebt. De bouw van een molecuul wordt meestal weergegeven met behulp van een structuurformule. Dit is eigenlijk een graaf met als punten de atomen en als links de atoombindingen, zie Figuur 3.

In deze lessenserie gaan we met een andere toepassing bezig: routes. Een graaf kan gebruikt worden om een wegennetwerk grafisch weer te geven. Hiermee wordt het mogelijk om wiskunde in te zetten voor het vinden van de snelste route van Enschede naar Amsterdam of om juist de mooiste fietsroute te vinden van Enschede naar Oldenzaal.

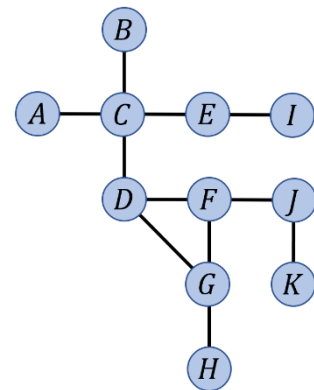
In Figuur 4 is een kaartje weergegeven van Overijssel met daarop de regionale treintrajecten. Dit is eigenlijk een voorbeeld van een graaf. De stations zijn hierin *de punten* en *de links* representeren de spoorlijnen tussen de stations. In Figuur 4b is een corresponderende graaf getekend. Let op dat de afstand tussen de punten en de ligging van de punten geen betekenis meer hebben.



Figuur 4a: treinnetwerk Overijssel.



Figuur 4b: corresponderende graaf.



Figuur 4c

OPDRACHT

2

Bekijk het netwerk in Figuur 4a en de corresponderende graaf in Figuur 4b goed.

- Welk punt in de graaf in Figuur 4b correspondeert met station Almelo? En welk station hoort bij punt J? *Leg je antwoord uit door het begrip graad te gebruiken.*
- Gebruik Figuur 4b. Met welke link correspondeert het stuk spoor tussen Zwolle en Mariënborg? En welke 2 stukken spoor kunnen horen bij link $\{H, I\}$? *Gebruik het begrip graad.*
- Gebruik Figuur 4c. Beargumenteer waarom deze graaf ook het treinnetwerk van Overijssel kan representeren. *Hint: match elk punt uit de graaf in Figuur 4c met een plaats uit het kaartje in Figuur 4a.*

Om van Kampen naar Hengelo te reizen moet je een route afleggen die verschillende stations passeert. Een mogelijke route gaat via Mariënborg. Deze route kunnen we met behulp van de graaf in Figuur 4b als volgt opschrijven:

$$R = \langle A, B, D, E, H \rangle.$$

Of korter:

$$R = \langle ABDEH \rangle.$$

Door het maken van een graaf verlies je de informatie over afstanden of reistijden tussen plaatsen. Dit kan worden opgelost door de introductie van een *gewogen* graaf. Dit is een graaf met bij elke link een getal. Dit getal kan bijvoorbeeld de afstand of de reistijd tussen twee punten zijn.

In Figuur 5 hieronder is een gewogen graaf bij het treinnetwerk uit Figuur 4 getekend. Bij elke link vind je een getal dat vertelt hoe lang het duurt om over die link te reizen. Zo kost een treinrit over het spoor tussen Zwolle (B) en Almelo (E) 38 minuten. Wiskundig noteren we dit als $c(B, E) = 38$.

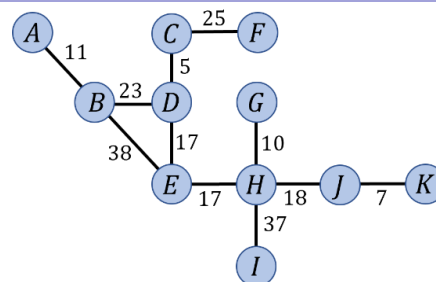
De lengte van een route R van een punt x naar y wordt aangegeven met $L(R)$. Zo geldt voor de route $R = \langle ABD \rangle$ van A naar D via B (Kampen – Zwolle – Mariënborg) dat de lengte $L(R)$ gelijk is aan:

$$L(R) = c(A, B) + c(B, D) = 11 + 23 = 34 \text{ minuten.}$$

De lengte R is dus gelijk aan de som van de lengtes van alle links in de route. In dit geval zijn dat de links $\{A, B\}$ en $\{B, D\}$.

NOTATIE

- Een route R van x_0 naar x_n via de punten $x_1, x_2, \dots, x_{n-1}$ geven we aan met $R = \langle x_0, x_1, x_2, \dots, x_{n-1}, x_n \rangle$.
- De lengte van een link $\{x, y\}$ geven we aan met $c(x, y)$.
- De lengte van een route schrijven we als $L(R)$.



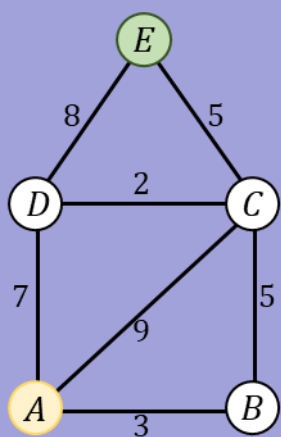
Figuur 5: gewogen graaf van het regionaal treinnetwerk Overijssel.

OPDRACHT

3

Gebruik Figuur 4a en Figuur 5.

- Bereken exact de reistijd tussen Kampen (A) en Enschede (J) als je de route $K = \langle A, B, D, E, H, J \rangle$ neemt. *Gebruik de juiste notatie.*
- Is de route zoals je die berekend hebt in a de snelste route? Zo nee, geef een snellere route en bereken hoeveel minuten het scheelt om die route te nemen.



Figuur 6: een gewogen graaf. Merk op dat we geen eenheden meer gebruiken net als in de meeste assenstelsels.

4

Bekijk de graaf in Figuur 6.

- Vind alle mogelijke routes van A naar E zonder punten vaker dan één keer te bezoeken.
- Gebruik de methode van *brute force* om de kortste route van A naar E te vinden. Gebruik de juiste notatie.

OPDRACHT

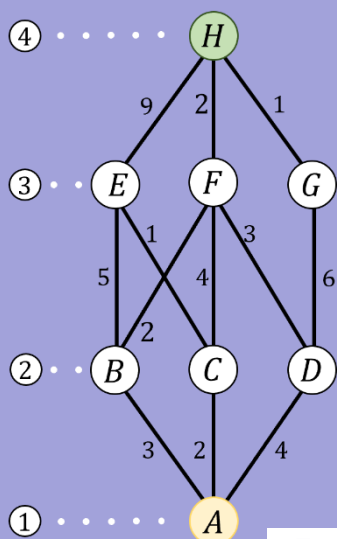
In de vorige opdracht heb je gezien dat een *brute force* aanpak niet erg efficiënt is. Je moest immers eerst alle routes vinden en daarvan de lengtes berekenen. Als je genoeg neemt met een snelle route en niet per se op zoek bent naar de snelste route dan kan een *heuristiek* soms uitkomst bieden. Een heuristiek is een snelle manier om iets redelijk goed op te lossen, maar zonder de garantie dat het optimaal is. Voor de graaf in Figuur 7 is de volgende heuristiek mogelijk:

- Start in het punt A op hoogte 1 en neem de kortste link uit A naar hoogte 2.
- In een punt kies de link naar boven uit dit punt met de laagste waarde.
- Herhaal 2 totdat je op hoogte 4 bent.

OPDRACHT

Bekijk de graaf in Figuur 7. **Werk opgave 5b uit op het werkblad.**

- Pas de hierboven beschreven heuristiek toe. Geeft de heuristiek de kortste route? Is het wel een goede oplossing? **Leg uit.**
- Verzin *andere getallen* op de links waardoor de heuristiek perfect (*linkergraaf*) of dramatisch slecht werkt (*rechtergraaf*).



5

Figuur 7: graaf met hoogtes. A ligt op hoogte 1, B, C, D op hoogte 2 etc.

Op het werkblad is dezelfde graaf 2 keer weergegeven zonder getallen bij de links (zie opgave 5b).

Les 2



Figuur 8: Edsger W. Dijkstra.

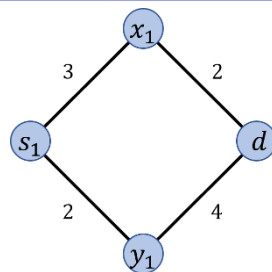
In deze les gaan we door met het rekenen aan grafen en leren we een algoritme dat gebruikt wordt om kortste routes te berekenen: Dijkstra's algoritme. Dit algoritme is in 1959 bedacht door Edsger Dijkstra (zie Figuur 8), een Nederlandse wiskundige en informaticus, terwijl hij met zijn vrouw op een terras in Amsterdam zat.

SAMENVATTING

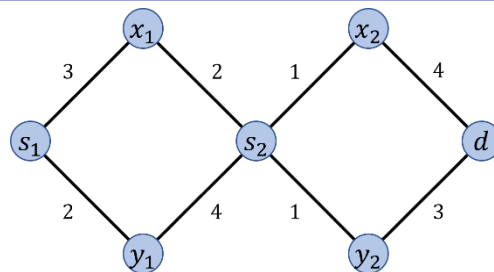
Vorige les heb je twee methodes geleerd om in grafen korte routes te berekenen:

- **Brute force:** vind alle mogelijke routes en bereken van elk van die routes de lengte.
- **Heuristiek:** een vuistregel om goede routes te berekenen.

We hebben al gezien dat een *brute force* aanpak zorgt voor optimale routes, maar dat dit ingewikkeld kan worden als de netwerken groter worden. We gaan nu bekijken hoe problematisch dit kan zijn! We gaan daarvoor aan de slag met de grafen in Figuur 9.



Figuur 9a



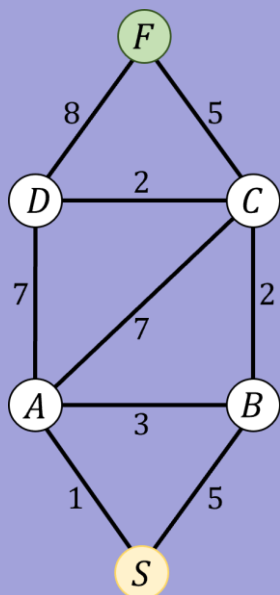
Figuur 9b

OPDRACHT

6

In Figuur 9 zijn twee grafen getekend die erg op elkaar lijken. We kunnen zeggen dat de linker graaf bestaat uit de component (s_1, x_1, y_1, d) en dat de rechtergraaf bestaat uit twee vergelijkbare componenten (er is een component toegevoegd aan de rechterkant van de graaf in Figuur 9a om de graaf in Figuur 9b te krijgen).

- Geef voor beide grafen de kortste route R van s_1 naar d en hun lengte $L(R)$.
- Hoeveel mogelijke routes van s_1 naar d zijn er in de linkergraaf? En hoeveel in de rechter?
- Stel dat we een component toevoegen (aan de rechterkant van de graaf), hoeveel mogelijke routes zijn er dan van s_1 naar d ?



Figuur 11: een gewogen graaf.

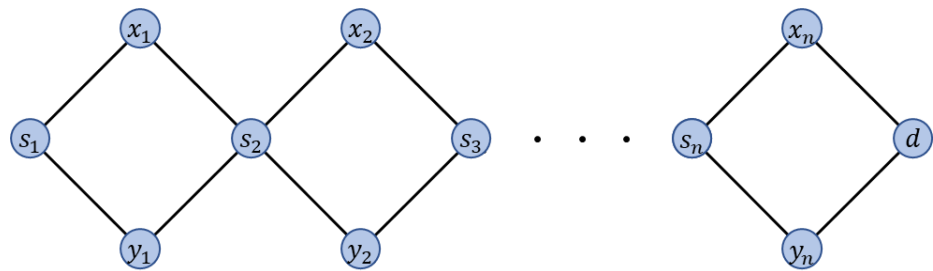
- d Neem nu aan dat de graaf niet bestaat uit 1, 2 of 3 componenten, maar dat de graaf uit n componenten bestaat (zie Figuur 10). Hoeveel mogelijke routes zijn er te vinden?

We gebruiken een oude computer uit het jaar 2001. We nemen aan dat de computer die we gebruiken ongeveer 0,0004 seconden nodig heeft om een route te vinden en de lengte ervan te berekenen. Om vervolgens de kortste route te vinden, berekent de computer het minimum van de gevonden lengtes. We verwaarlozen de tijd die de computer daarover doet.

- e Neem aan dat de graaf uit 36 componenten bestaat. Hoe lang duurt het om met een *brute force* aanpak de kortste route te vinden? Geef je antwoord in *gehele dagen*.

Tussen 2001 en 2021 zijn de computers ongeveer 2000 keer zo snel geworden. Dit betekent dat de rekestijd van de computer ook 2000 keer zo kort is geworden.

- f Kan deze moderne computer een graaf berekenen waarin er 2000 keer zo veel componenten zijn als de graaf uit opgave e? Leg je antwoord uit.



Figuur 10: impressie van een graaf met n componenten.

We hebben nu gezien dat een *brute force* aanpak al op relatief kleine netwerken problematisch kan zijn. Om systematischer te werk te gaan zetten we de volgende notatie in:

- De (optimale) afstand van het startpunt S naar een punt X in de graaf noteren we met $\delta(X)$.

OPDRACHT

7

Bekijk de gewogen graaf in Figuur 11.

- a Bereken $\delta(A)$ en $\delta(B)$.
- b Gegeven het punt F en een route R van S naar F , wanneer geldt dat $L(R) = \delta(F)$? *Hint: geef R voor de graaf in Figuur 11 en bedenk een algemene regel.*

8

Bekijk opnieuw de gewogen graaf in Figuur 11.

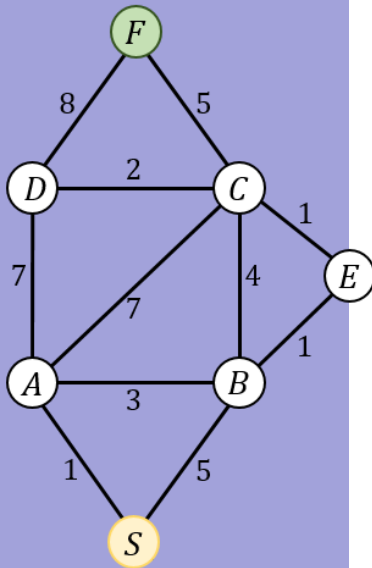
- a** Bereken $\delta(C)$, $\delta(A) + c(A, C)$ en $\delta(B) + c(B, C)$.
- b** Stelling: het is onmogelijk om een punt X te vinden zodat $\delta(X) + c(X, C) < \delta(C)$.

Klopt deze uitspraak? *Zo ja*, leg uit waarom. *Zo nee*, geef een punt X waarvoor de ongelijkheid geldt.

- c** Wanneer geldt $\delta(X) + c(X, C) = \delta(C)$?



Checkpoint: laat bovenstaande vraag aftekenen om verder te mogen naar de volgende pagina.



Figuur 12

Bekijk nu de graaf in Figuur 12. De kortste afstand van S naar C is gelijk aan $\delta(C) = 6$. De kortste route is $R^* = \langle S, A, B, E, C \rangle$. De kortste route is aangegeven met een sterretje, *. De link $\{E, C\}$ zit in R^* . Omdat de link $\{E, C\}$ in een kortste route van S naar C zit geldt:

$$\delta(E) + c(E, C) = \delta(C).$$

Waarom? De kortste afstand om van S naar E te komen is $\delta(E)$. De kortste route naar C loopt via E . Dus:

$$\delta(C) = c(S, A) + c(A, B) + c(B, E) + c(E, C) = \delta(E) + c(E, C).$$

SAMENVATTING

- De (optimale) afstand van het startpunt S naar een punt X in de graaf noteren we met $\delta(X)$.
- De vergelijking $\delta(X) + c(X, Y) = \delta(Y)$ geldt als de link $\{X, Y\}$ in een kortste pad zit van S naar Y .
- Er geldt $\delta(X) + c(X, Y) \geq \delta(Y)$ omdat de lengte van elke route naar Y langer is dan $\delta(Y)$ of gelijk is aan $\delta(Y)$. Zo ook de route met daarin de link $\{X, Y\}$.

Het algoritme van Dijkstra om kortste routes te berekenen gebruikt voor elk *punt* in de graaf een label $d(X)$. Dit label is gelijk aan de voorlopig kortste afstand tot het punt X . Dit is de afstand die de computer tot op dat moment kan zien als de kortste afstand tot het punt X vanuit het startpunt. Wat we hiermee bedoelen wordt duidelijk in de volgende opdracht.

OPDRACHT

9

Bekijk de graaf in Figuur 12 opnieuw. De labels $d(X)$ houden de voorlopig kortste afstand bij tot een punt X . Dit is de afstand die de computer tot op dat moment kan zien als de kortste afstand vanuit het startpunt S . De computer kijkt niet verder dan de burens van een punt bij het zoeken naar kortste afstanden. Neem aan dat $d(S) = 0$ omdat S het startpunt is.

- Begin in het punt S . Beredeneer dat $d(A) = 1$ en $d(B) = 5$. We hebben nu gekeken naar alle links die uit S vertrekken. De computer gaat nu naar een volgend punt.
- Waarom is het handig nu met punt A verder te gaan? Ga naar het punt A . Opnieuw gaan we de labels d updaten.
- Laat zien dat na deze stap geldt dat $d(B) = 4$.
- Waarom geldt $d(C) = 8$? En $d(D) = 8$?

In de vorige opdracht heb je gekeken naar voorlopige labels d voor punten in de graaf. In de volgende opdrachten gaan we een stap verder en gebruiken we de voorlopige labels d om het algoritme van Dijkstra te maken. Dit algoritme laat ons de kortste route van S naar F berekenen. Om te beginnen gaan we kijken wat een algoritme is.

Een **algoritme** is een stappenplan om een bepaald doel te bereiken. Een mooi voorbeeld uit het dagelijkse leven is het volgen van een recept. Bij het bereiden van een maaltijd volg je de stappen die in het recept zijn opgeschreven en bereid je op deze manier het gerecht. Een voorbeeld uit het vak wiskunde is het algebraïsch berekenen van extreme waarden van een functie $f(x)$.

10

a Beschrijf een ander algoritme dat je wel eens gebruikt in de wiskundeles.

Het is belangrijk dat alle stappen in een algoritme duidelijk zijn. In de volgende opdracht bekijk je een algoritme voor een huishoudrobot. De taak van de robot is om de afwas te doen. Hierbij voert de robot het algoritme uit zonder zelf na te denken. De robot doet dus *precies* wat je zegt. Het algoritme bestaat uit 3 stappen, zie Figuur 13.



Maak de vieze afwas schoon.

Laat de schone afwas drogen.

Plaats de droge afwas in de kasten.

Figuur 13: schoonmaakrobot met een algoritme.

De robot heeft het algoritme steeds precies gevolgd. Hierbij is een aantal dingen misgegaan. Op het werkblad staan vier afbeeldingen op chronologische volgorde. In elke afbeelding is het algoritme voor de afwasrobot verbeterd, omdat duidelijk werd dat de robot zijn taak niet zoals verwacht uitvoerde.

b Geef bij elke afbeelding aan wat de robot fout gedaan zou kunnen hebben waardoor het algoritme aangepast moest worden (noem per afbeelding 1 fout).

Hint: voor afbeelding 1 is de fout al gegeven.

Het algoritme van Dijkstra is een stappenplan dat in stapjes de kortste afstand van S naar F berekent. Je hebt al een stukje van dit algoritme gedaan in opdracht 9. De resultaten van opdracht 9 zijn in Tabel 1 gezet (zie pagina 13). We gaan nu in stappen de tabel verder aanvullen. Daarvoor gebruiken we Dijkstra's algoritme, waarvan een deel hieronder beschreven staat:

DIJKSTRA'S ALGORITME

1. Begin in het punt S .
2. Bereken $d(X)$ voor alle punten X die verbonden zijn met S via een link.
3. Ga naar het punt met de laagste waarde voor d dat nog niet eerder aan bod is geweest. Zijn er meer punten mogelijk, kies er dan één. Dit punt heet X_c (de c komt van het Engelse woord *current*, het huidige punt). Bestaat dit punt niet, ga dan naar stap 5.
4. ...
5. Geef $d(F)$.

OPDRACHT

Bekijk de graaf in Figuur 12a (dit is dezelfde graaf als in Figuur 12). Dit is dezelfde graaf als in opdracht 9. In de vragen 9a en 9b heb je de stappen 1, 2 en 3 van Dijkstra's algoritme al toegepast:

1. Je bent begonnen in het punt S .
2. De burenen van S zijn de punten A en B . Vanuit S gezien is de kortste afstand tot A en B voorlopig gelijk aan $d(A) = 1$ en $d(B) = 5$.
3. Het punt met de laagste waarde voor d is A . Dus $X_c = A$. We gaan daarheen.

In vraag 9c heb je laten zien dat $d(B) = 4$. Daarnaast heb je laten zien in 9d dat $d(C) = 8$ en $d(D) = 8$. We gaan nu een regel verzinnen voor stap 4.

- a** Vul stap 4 hieronder aan door de gestreepte lijnen in te vullen:
Kies uit: $d(X)$, $d(X_c)$, $\delta(X)$, $\delta(X_c)$.

4. Bekijk elk punt X dat direct grenst aan het punt waar je nu bent, X_c .

Als het punt X nog geen label heeft:

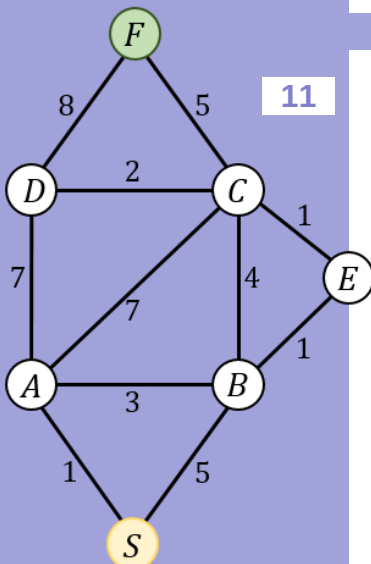
↳ Maak $d(X) = \text{_____} + c(X_c, X)$.

Als het punt X al een label $d(X)$ heeft:

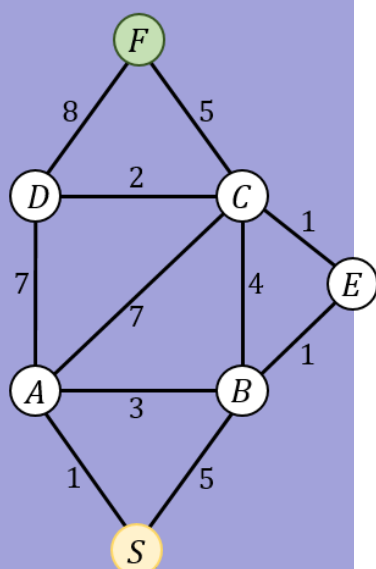
↳ Update $d(X)$ naar $d(X) = d(X_c) + c(X_c, X)$
als $d(X_c) + c(X_c, X) < \text{_____}$.

Klaar? Ga dan terug naar stap 3.

Hint: gebruik het voorbeeld uit 9d, waar $X_c = A$.



Figuur 12a: kopie van de gewogen graaf uit Figuur 12.



Figuur 12b: kopie van de gewogen graaf uit Figuur 12.

- b** Als we het voorbeeld uit opdracht 9 verder volgen, waarom geldt $X_c = B$ als je teruggaat naar stap 3 aan het eind van stap 4?

Nadat we het punt A hebben bezocht gaan we verder. We hebben de labels d van de burens van $X_c = A$ geüpdatet. Dit waren de labels $d(B) = 4$, $d(C) = 8$ en $d(D) = 8$. Vervolgens is aan het eind van stap 4 X_c geüpdatet naar B .

In Tabel 1 hieronder staan de voorlopige labels voor elk punt ingevuld voor Dijkstra's algoritme.

- In de **groene regel** zien we het resultaat van stap 1 en 2 van het algoritme.
- Omcirkeld zie je het punt X_c .
- In de **blauwe regel** zie je het resultaat van stap 3 en 4 na 1 herhaling.

	S	A	B	C	D	E	F
0	0	1	5	×	×	×	×
1	0	1	4	8	8	×	×
2	0	1	4				
3	0	1	4				
4	0	1	4				
5	0	1	4				
6	0	1	4				

Tabel 1: Dijkstra's algoritme op de graaf uit Figuur 12.

- c** Herhaal stap 4 totdat de hele tabel is ingevuld.
d Wat is de kortste afstand van S naar F ?
e Ga na dat voor elk punt X geldt dat het getal uit de onderste rij in de tabel gelijk is aan $\delta(X)$. Voorbeeld: $\delta(A) = 1$ komt overeen met de waarde uit de tabel.



Checkpoint: laat bovenstaande vraag aftekenen om verder te mogen naar de volgende pagina.

Les 3

In de vorige les heb je kennism gemaakt met het algoritme van Dijkstra. In deze les herhalen we het algoritme en gaan we er verder mee werken.

SAMENVATTING

Hieronder vind je het complete algoritme van Dijkstra.

1. Begin in het punt S .
2. Bereken $d(X)$ voor alle punten X die verbonden zijn met S via een link.
3. Ga naar het punt met de laagste waarde voor d dat nog niet eerder aan bod is geweest. Zijn er meer punten mogelijk, kies er dan één. Dit punt heet X_c . Bestaat dit punt niet, ga dan naar stap 5.
4. Bekijk elk punt X dat direct grenst aan het punt waar je nu bent, X_c .

Als het punt X nog geen label heeft:

↳ Maak $d(X) = d(X_c) + c(X_c, X)$.

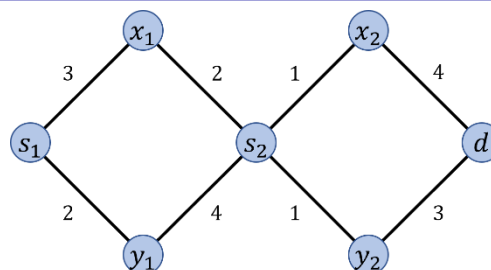
Als het punt X al een label $d(X)$ heeft:

↳ Update $d(X)$ naar $d(X) = d(X_c) + c(X_c, X)$ als $d(X_c) + c(X_c, X) < d(X)$.

Klaar? Ga dan terug naar stap 3.

5. Geef $d(F)$.

VOORBEELD 1



Figuur 13: voorbeeldgraaf met twee componenten.

Bereken met behulp van Dijkstra's algoritme de kortste afstand van s_1 naar d .

Uitwerking

1. We beginnen in het punt s_1 .
2. $d(x_1) = 3, d(y_1) = 2$.
3. $X_c = y_1$, want $d(y_1)$ heeft de laagste waarde.
4. $d(s_2) = d(y_1) + c(y_1, s_2) = 2 + 4 = 6$.
3. We gaan naar x_1 , want $d(x_1) \leq d(s_2)$. Dus $X_c = x_1$.

- Het vervolg is samengevat in Tabel 2.

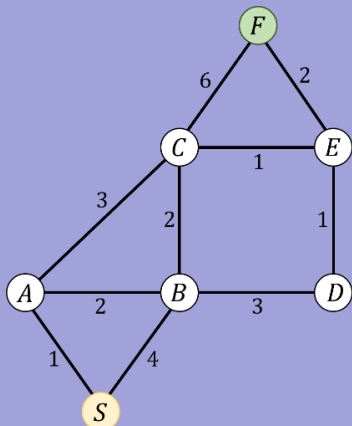
	s_1	x_1	y_1	s_2	x_2	y_2	d
0	0	3	2	×	×	×	×
1	0	3	2	6	×	×	×
2	0	3	2	5	×	×	×
3	0	3	2	5	6	6	×
4	0	3	2	5	6	6	10
5	0	3	2	5	6	6	9
6	0	3	2	5	6	6	9

Tabel 2: Dijkstra's algoritme op de graaf uit Figuur 13.

OPDRACHT

12

Bereken voor de graaf in Figuur 14 de lengte van de kortste route van S naar F met behulp van Dijkstra's algoritme. Gebruik de tabel op het werkblad.



Figuur 14

Er mist nog 1 stap in ons doel om de kortste route van een startpunt naar het eindpunt te vinden. We hebben immers de kortste afstand gevonden, nog niet de precieze route. Gelukkig heb je al wel alle ingrediënten gevonden om de kortste route te vinden.

In de onderste regel van Tabel 2 heb je de afstandslabellen $\delta(s_1)$, $\delta(x_1)$, $\delta(y_1)$, etc. gevonden (*controleer maar!*). Verder heb je geleerd dat de vergelijking $\delta(X) + c(X, Y) = \delta(Y)$ geldt als de link $\{X, Y\}$ in een kortste pad zit van het startpunt naar Y .

Laten we eens beginnen in het punt d van Figuur 13 (op de vorige pagina). We hebben:

$$\delta(d) = 9.$$

Hoe komen we in het punt d ? Er zijn 2 opties: via link $\{x_2, d\}$ of $\{y_2, d\}$. We hebben:

$$\delta(x_2) + c(x_2, d) = 6 + 4 = 10 > \delta(d) = 9.$$

$$\delta(y_2) + c(y_2, d) = 6 + 3 = 9 = \delta(d).$$

Dus $\{y_2, d\}$ zit in de kortste route. We gaan dus naar y_2 .

Vanuit y_2 is het simpel, aangezien er maar één mogelijkheid is. De link $\{s_2, y_2\}$ hoort in de kortste route.

13

Gebruik de graaf in Figuur 13 (zie pagina 14).

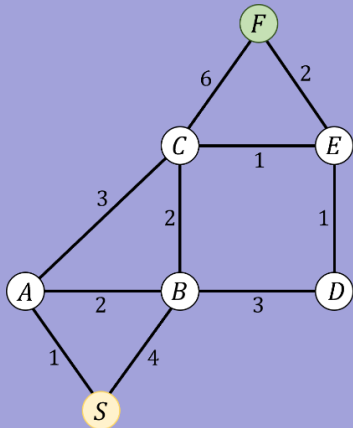
- Laat op dezelfde manier zien dat vanuit s_2 de link $\{x_1, s_2\}$ moet worden toegevoegd.
- Geef nu de kortste route van s_1 naar d .
- Vul het onderstaande algoritme aan om de kortste route te vinden. *Hint: gebruik labels $\delta(X)$, $\delta(X_c)$, $d(X)$ en/of $d(X_c)$.*
 - Laat X_c het eindpunt zijn.
 - Voor elk punt X grenzend aan X_c

Als $\delta(X) + c(X, X_c)$ _____
Voeg $\{X, X_c\}$ toe aan de kortste route.
Onthoud het punt X .

Als $\delta(X) + c(X, X_c)$ _____
Voeg $\{X, X_c\}$ **niet** toe aan de kortste route.
 - Maak het punt X dat je onthouden hebt X_c tenzij X het startpunt is. In dat geval heb je de kortste route gevonden.

Gebruik de graaf in Figuur 14a (kopie van Figuur 14).

- Gebruik het algoritme bij c om de kortste route van S naar F te vinden voor de graaf in Figuur 14a.



Figuur 14a: kopie van Figuur 14.

14

Voor de graaf in Figuur 13 is Dijkstra's algoritme toegepast (zie pagina's 14 en 15). Hierin zagen we dat het algoritme van Dijkstra 6 stappen nodig heeft om de lengte van de kortste route te vinden (het aantal rijen in de tabel). In opgave 13 heb je laten zien dat de kortste route vervolgens in 4 stappen te vinden is.

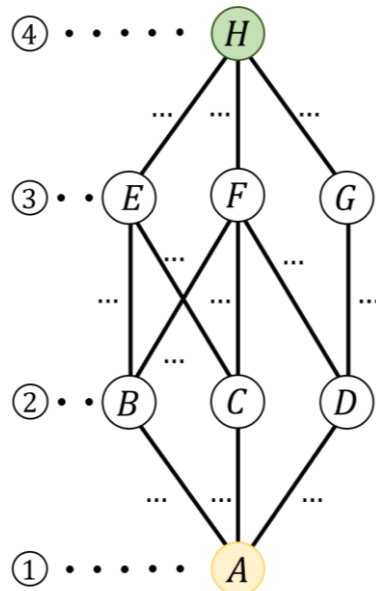
In deze opgave gebruiken we dezelfde trage computer uit 2001. Deze computer heeft 0,0004 seconden nodig om stap 4 van Dijkstra's algoritme uit te voeren voor één punt X . Op dezelfde manier doet de computer er 0,0004 seconden over om stap 2 van het algoritme uit opdracht 13c uit te voeren voor één punt X . De rest van de stappen zijn verwaarloosbaar in tijd.

- Laat zien dat de computer minder dan $7 \cdot 6 \cdot 0,0004 + 6 \cdot 6 \cdot 0,0004 = 0,0312$ seconden nodig heeft om de kortste route te vinden.
- Ga nu uit van n componenten. Laat zien dat een computer voor die graaf minder dan $4n \cdot 4n \cdot 0,0004 + 4n \cdot 4n \cdot 0,0004$ seconden nodig heeft en bereken voor $n = 36$ hoe lang de computer erover rekent. *Hint: gebruik opgave a.*
- Hoeveel eerder is de computer klaar door Dijkstra's algoritme te gebruiken dan met een *brute force* aanpak? *Vergelijk met je antwoord op opgave 6e.*
- Wat is belangrijker denk je, de introductie van betere algoritmes of snellere computers?

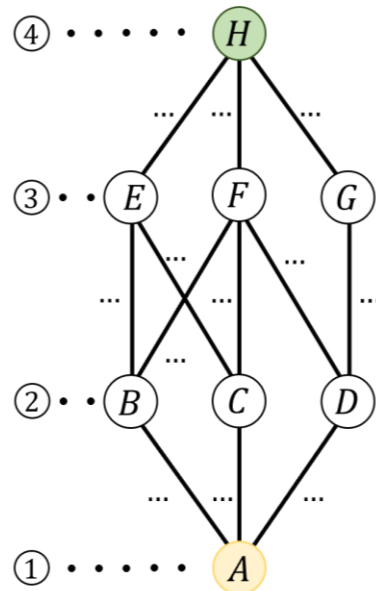
Werkblad

Gebruik dit werkblad bij het oplossen van opgaves 5, 10, 11 en 12.

BIJLAGE OPDRACHT 5b

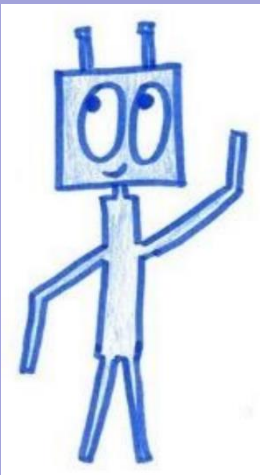


Vul getallen in op de links zodat de heuristiek **perfect** werkt.



Vul getallen in op de links zodat de heuristiek **dramatisch slecht** werkt.

BIJLAGE OPDRACHT 10



Figuur: de afwasrobot.

MaaK de vieze afwas schoon.

Laat de schone afwas drogen.

*Plaats de droge afwas in de kasten.**

**BEHALVE in het uur voor de maaltijd;
in dat geval laat je ze staan*

Figuur 1: de robot heeft borden opgestapeld in de kast die al op tafel stonden voor het avondeten.

Zie volgende pagina voor het vervolg

Maak de vieze afwas schoon.

↳ maar wacht totdat alle grote etensresten weg zijn

Laat de schone afwas drogen.

Plaats de droge afwas in de kasten.*

* BEHALVE in het uur voor de maaltijd;
in dat geval laat je ze staan

Figuur 2: beschrijf in je uitwerking van opdracht 10 wat er fout gaat.

Maak de vieze afwas schoon.

↳ maar wacht totdat alle grote etensresten weg zijn*
* of totdat het weghalen van eten gestopt is

Laat de schone afwas drogen.

Plaats de droge afwas in de kasten.*

* BEHALVE in het uur voor de maaltijd;
in dat geval laat je ze staan

Figuur 3: beschrijf in je uitwerking van opdracht 10 wat er fout gaat.

Zie volgende pagina voor het vervolg

Maak de vieze afwas schoon.

↳ maar wacht totdat alle grote etensresten weg zijn *

* of totdat het weghalen van eten gestopt is

voor MINSTENS 5 MINUTEN

Laat de schone afwas drogen.

↳ behalve wanneer het vlak voor de maaltijd is en er niet genoeg borden zijn, droog ze in dit geval

Plaats de droge afwas in de kasten.*

* BEHALVE in het uur voor de maaltijd;
in dat geval laat je ze staan als ze

op de tafel staan; die in het droogrek ruim je op.

↳ of op het aanrecht

Figuur 4: beschrijf in je uitwerking van opdracht 10 wat er fout gaat.

BIJLAGE OPDRACHT 11

	S	A	B	C	D	E	F
0	0	1	5	×	×	×	×
1	0	1	4	8	8	×	×
2	0	1	4				
3	0	1	4				
4	0	1	4				
5	0	1	4				
6	0	1	4				

Tabel 1: tabel voor opdracht 11c.

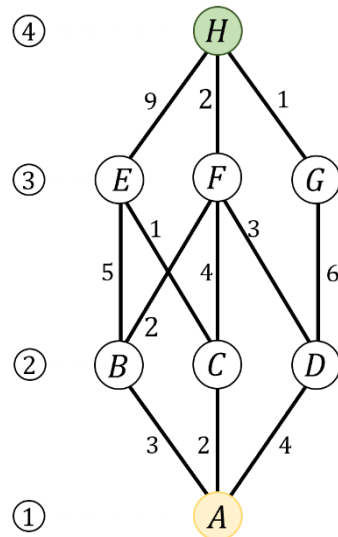
BIJLAGE OPDRACHT 12

	S	A	B	C	D	E	F
0	0						
1							
2							
3							
4							
5							
6							

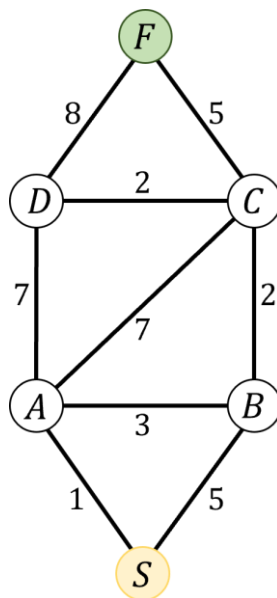
Tabel 2: tabel voor opdracht 12.

Figurenoverzicht

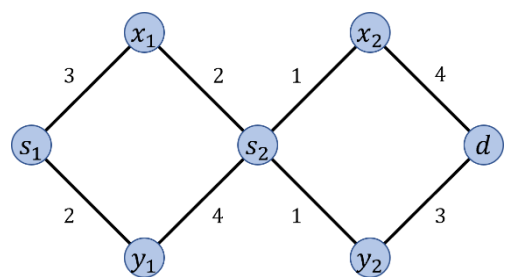
Op deze pagina vind je de figuren 7, 11 en 13 die meerdere keren terugkomen zodat je niet hoeft te bladeren.



Figuur 7: graaf met hoogtes. A ligt op hoogte 1, B, C, D op hoogte 2 etc. Op het werkblad is dezelfde graaf 2 keer weergegeven zonder getallen bij de links.



Figuur 11



Figuur 13