Technische Universiteit Delft

module 2

Het algoritme van Dijkstra

Dit studiemateriaal is ontwikkeld door de kerngroep wiskunde D Delft en mag gratis gebruikt worden in het wiskundeonderwijs in het vo.

Kerngroep wiskunde D Delft

Liesbeth Bos	Scala College	
Wim Caspers	TU Delft / Adelbert College	
Wim van Dijk	Montessori Lyceum	
David Lans	Emmaus College	
Jan Moen	Int. College Edith Stein	
Rob van Oord	**Coenecoop College**	**Module 2**
Sanne Schaap	Marecollege	
Jan Schrik	Christelijk Lyceum Delft	
Jeroen Spandaw	TU Delft	
Agnes Verweij	TU Delft	

website: www.wiskundedsteun.nl
contact: w.t.m.caspers@tudelft.nl

Module 2	**Het algoritme van Dijkstra**	
Theorie	**p 5-12**	
Antwoorden	**p 13-17**	
Bijlage 1 **(figuren en tabel bij oefening 2 tot en met 5)**	**p 18-19**	
Bijlage 2 **(figuur Seervada Park)**	**p 20**	
Bijlage 3 **(antwoordenblad oefening 7)**	**p 21-22**	
Programmeren	**p 23-25**	
Toetsopgaven	**p 26x-34**	
Literatuur	**H. Tijms, 2002, Operationele analyse, Epsilon Uitgaven**	**(p. 143, 144,** **en 146 t/m 149)**
	F. S. Hillier & G. J. Lieberman, 2005, Introduction to operations research, 8th ed., McGraw-Hill Int. Editions	**H9 p. 380, 381**
	H. Broersma, 2006, Grafen in de praktijk, Epsilon Uitgaven; Zebra-reeks nr. 14	**H3 p. 39-45**
	K. Roos, Kaleidoscoop 1e jaars TUD, F **Optimaliseren in Netwerken, september 2006**	

Geachte docent,

Ter inleiding op een of meer onderwerpen van Optimaliseren in netwerken kunt u op diverse wijzen gebruik maken van bovengenoemd materiaal.

U kunt de opdrachten voorleggen aan de leerlingen en de antwoorden voor uzelf houden. U kunt er ook voor kiezen de antwoorden ook aan de leerlingen uit te delen.

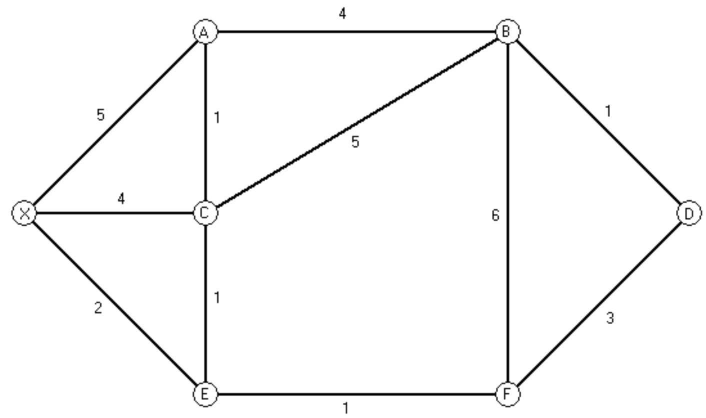
Om leerlingen van het begin af aan vertrouwd te maken met Engelstalig studiemateriaal op dit gebied kunt u ze ook direct de genoemde paragrafen uit Introduction to Operations Research (Hillier / Lieberman) voorleggen. Dit boek wordt aan universiteiten gebruikt. Wellicht kan de woordenlijst het lezen van dit universitair studiemateriaal bij volgende onderwerpen van Optimaliseren in netwerken vergemakkelijken.

Als u er voor kiest de leerlingen te toetsen aan de hand van een in te leveren verslag of werkstuk, is het stuk over programmeren uit module 0 misschien interessant. Met dit gratis programma van het Freudenthal Instituut kunnen de leerlingen een graaf tekenen en bijbehorende afstandentabel invullen.

Module 2

Kortste pad

In figuur 1 zie je een netwerk getekend. Bij de lijntjes die de knoopunten met elkaar verbinden staan getallen. Deze getallen worden vaak "gewichten" genoemd, maar kunnen in principe van alles betekenen, zoals afstand tussen de knopen die erdoor verbonden worden, of reiskosten, reistijd, enz. De getallen bij de zogenaamde *takken* (*links*) geven dus alleen schematisch de "afstanden" aan tussen de zogenaamde *knopen* (*nodes*).
We zijn benieuwd naar de kortste route langs de takken van ***X*** naar ***B***.



figuur 1

oefening 1
Zoek het kortste pad van **X** naar ***B***. Hoe groot is de lengte ervan?
Vermeld daarbij ook de route met behulp van de letters van de knopen waarlangs dit pad loopt.
Vertel met eigen woorden hoe je het hebt gevonden.

In de praktijk bestaan netwerken meestal uit honderden knopen met nog grotere aantallen takken. Je kunt je voorstellen dat het handig is als je een recept hebt waarmee je de kortste route kunt vinden. Dit vraagt meestal veel rekenwerk. De computer kan dan helpen om de kortste paden te vinden.
In deze module leer je manieren (algoritmen) om de kortste route tussen knopen te berekenen. We zoeken naar paden tussen knopen waarbij het totale gewicht van de takken op dat pad minimaal is. Zo'n pad heet dan het *kortste pad*. Je zult begrijpen dat het kan voorkomen dat er in een netwerk verschillende routes kunnen zijn met hetzelfde (minimale) gewicht.

In de volgende paragrafen kom je verschillende algoritmes tegen waarmee je systematisch een kortste pad kunt vinden. Het eerst komt het algoritme van Dijkstra aan bod. We nemen daarbij figuur 1 als voorbeeld. Daarna word je gevraagd het algoritme uit het boek van Hillier & Lieberman te bestuderen. De tekst is in het Engels. Als je dit niet meteen kunt volgen, kun je verderop in die paragraaf lezen hoe zij het doen. Er volgen er enkele oefeningen met dit algoritme. Daarna word je gevraagd beide algoritmes te vergelijken.
Tot slot moet je proberen ook de tekst in het boek van H. Tijms te begrijpen. Als je wilt weten hoe je een programma voor de GR moet schrijven dat het kortste pad berekent, dan moet je de syllabus van K. Roos er op naslaan. In de bijlage

staan programmaatjes waarmee je op de TI-83 of TI-84 PLUS een kortste pad kunt laten berekenen.

Het algoritme van Dijkstra

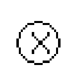
(gebruik bij deze paragraaf Bijlage 1 en 2)

Een manier om systematisch een kortste pad in een netwerk te vinden is volgens het algoritme van Dijkstra. Dit algoritme bepaalt op een slimme manier de kortste weg van het beginpunt X naar elke knoop van het netwerk. Stap voor stap wordt elke knoop afgehandeld, de dichtstbij gelegen knoop het eerst. In een tabel houden we de vorderingen bij. Elke stap levert een nieuwe regel in de tabel. Van elke knoop wordt vermeld hoe lang (***L***) het kortste pad naar deze knoop is en ook via (***V***) welke knoop dat kortste pad die knoop bereikt. Dat wil zeggen dat in een tabel bij elke knoop wordt bijgehouden hoe lang (***L***) het kortste pad naar deze knoop tot dan toe is, en ook wat het ***V***oorlaatste punt is voor deze knoop: het zogenaamde *V*-label.
Een voorbeeld: onder knoop A staat in de derde regel van de tabel "5 ; *X*". Dit betekent dat het kortste pad van X naar A dat tot dan toe gevonden is, een lengte heeft van 5 en dat je over dit pad rechtstreeks vanuit knoop X komt.
Om de kortste paden van X naar elke knoop uit het netwerk te vinden, bouwen we het netwerk als het ware stap voor stap opnieuw op.
Je kunt zelf het labelen bij de stappen bijhouden in de tabel van bijlage 1. Je vindt daar ook het netwerk uit figuur 1.

Begin met knoop X.
Vanaf dit moment gaan we stuk voor stuk alle knopen "afhandelen" totdat we ze allemaal gehad hebben. Hoe dat moet dat lees je hieronder. Houd het netwerk van de bijlage erbij.

Stap 1.
De eerste stap van het algoritme legt in feite vast dat we bij X het beginpunt is. Dat gaat als volgt [kijk ook bij stap 1 in de tabel hieronder]:
Zet onder de startknoop X: "0; X", waarmee je aangeeft dat het kortste pad van X naar X gewicht 0 heeft en dat je X bereikt via X.
Wanneer twee punten (nog) niet rechtstreeks door een tak verbonden zijn stellen we de afstand op ∞. Omdat X nog met geen enkel andere knoop verbonden is, is de afstand tot alle andere knopen dus in eerste instantie oneindig groot (∞).
Onder alle andere knopen zet je dus de afstand ∞, gevolgd door ; - . Dit streepje wordt gezet omdat er nog geen korste pad naar die andere knopen gevonden is.
Knoop X is nu afgehandeld. In de kolom onder X (de tweede kolom) komt geen nieuwe (kortere) afstand meer.

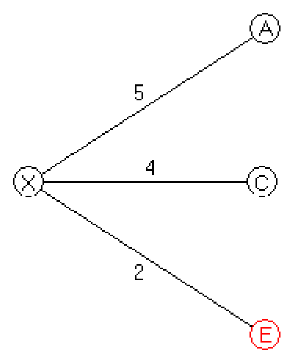
Stap 1:
Netwerk opbouwen te beginnen met knoop X;
de afstanden tot alle andere knopen zijn ∞;
het tweede label van elke knoop is een streepje (-),
want er is nog geen enkele verbinding met X.



Stap 2.
In de tweede stap wordt de knoop gezocht die het dichtst bij X ligt.Dit gaat als volgt [gebruik nu verder de tweede figuur van de bijlage, zet bij elke volgende stap de afstanden die in aanmerking komen in de figuur, in deze stap zijn dat 5 (tot A), 4 (tot C) en 2 (tot E)]. Onder alle knopen die rechtstreeks met X verbonden zijn zet je hun afstand tot X, gevolgd door ; X. Knoop X heeft tot knoop A een afstand 5, tot C een afstand 4, tot E een afstand 2. We zeggen dus dat L(A) = 5, L(C) = 4 en L(E) = 2, en V(A) = V(C) = V(E) = X voor alle drie de punten. Omdat L(E) het kleinste is, zeggen we dat knoop **E** nu is afgehandeld. Knoop E wordt vervolgens in de laatste kolom achter de X als afgehandeld opgeschreven. Kleur E in de tweede figuur van de bijlage. In de kolom onder E komt bij volgende stappen geen nieuwe (kortere) afstand meer.

	X	A	B	C	D	E	F	afgehandelde knopen
stap	L(X); V	L(A); V	L(B); V	L(C); V	L(D); V	L(E); V	L(F); V	
1	0 ; X	∞ ; -	∞ ; -	∞ ; -	∞ ; -	∞ ; -	∞ ; -	X
2		5 ; X	∞ ; -	4 ; X	∞ ; -	2 ; X	∞ ; -	X,E

Stap 2:
Alle verbindingen met X worden bekeken;
de kortste afstand daarvan is naar E;
E is afgehandeld;
het is handig om afgehandelde punten te kleuren; kleur E



Stap 3.
In deze stap komen de takken erbij van E naar direct met E verbonden knopen. De afstanden 1 (van E tot C) en 1 (van E tot F) komen erbij. We kijken naar knopen die via E een kortere route naar X zouden kunnen krijgen dan die er al was en naar nieuwe routes. Het gaat nu om de knopen C en F.
Voor de knopen K die al met X verbonden zijn (alleen C in dit geval) berekenen we welke van de afstanden korter is: de afstand *L*(K) die de knoop al had (tot X) of *L*(E) + *W*(EK) = de afstand van X tot E + de afstand van E tot K. Voor K = C krijg je: *L*(C) = 4 en *L*(E) + *W*(EC) = 2 + **1 = 3**, dus *L*(C) wordt 3 i.pv. 4 in stap 3, en het *V*-label van C wordt E. F is nu door E ook verbonden met X met afstand 3, dus L(F) verandert van ∞ in 3. Het *V*-label van F wordt ook E. Verder verandert er niets.

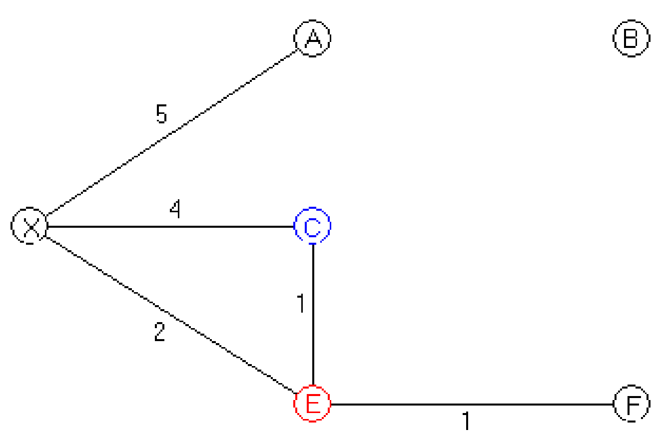
Wanneer we nu stap 3 invullen krijg je de volgende tabel:

	X	**A**	**B**	**C**	**D**	**E**	**F**	**afgehandelde knopen**
stap	***L(X)*; V**	***L*(A); *V***	***L*(B); *V***	***L*(C); *V***	***L*(D); *V***	***L*(E); *V***	***L*(F); *V***	
1	**0 ; X**	**∞ ; -**	**∞ ; -**	**∞ ; -**	**∞ ; -**	**∞ ; -**	**∞ ; -**	**X**
2		**5 ; X**	**∞ ; -**	**4 ; X**	**∞ ; -**	**2 ; X**	**∞ ; -**	**X,E**
3		**5 ; X**	**∞ ; -**	**3 ; E**	**∞ ; -**		**3 ; E**	**X,E,?**
4								

Je ziet dat er nu twee (nog) niet afgehandelde knopen zijn met (kleinste) afstand 3. Het maakt niet uit welke van de twee als afgehandeld beschouwd wordt. Dit heeft geen invloed op de uiteindelijke oplossing. We kiezen C. Dus bij stap 3 in de rechter kolom wordt ? = C.

Stap 3:
Alle directe verbindingen met E komen erbij; (kortere) afstanden naar X (via E) worden in de tabel genoteerd; kies een van de knopen met de kortste afstand (dus C of F); we kiezen C;
C is nu afgehandeld (kun je kleuren).

[F is het andere punt met afstand 3 tot X]



Stap 4.
In deze stap moeten alle directe verbindingen vanuit C erbij betrokken worden.
Dit zijn CA en CB.

oefening 2
Zoek nu hoe de lengtes van de routes (via C) naar A en B moeten worden aangepast .
Vul de labels achter stap 4 in de tabel hierboven in.
Welk punt kun je nu als afgehandeld beschouwen?

Stap 5 t/m 7.
We vervolgen dit algoritme tot en met stap 7, dan zijn alle punten afgehandeld.
Er zijn immers zeven knopen. In de onderstaande tabel zie je de oplossing.

	X	**A**	**B**	**C**	**D**	**E**	**F**	**afgehandelde knopen**
stap	**L(*X*); *V***	***L*(A); *V***	***L*(B); *V***	***L*(C); *V***	***L*(D); *V***	***L*(E); *V***	***L*(F); *V***	
1	**0 ; X**	**∞ ; -**	**∞ ; -**	**∞ ; -**	**∞ ; -**	**∞ ; -**	**∞ ; -**	**X**
2		**5 ; X**	**∞ ; -**	**4 ; X**	**∞ ; -**	**2 ; X**	**∞ ; -**	**X,E**
3		**5 ; X**	**∞ ; -**	**3 ; E**	**∞ ; -**		**3 ; E**	**X,E,C**
4		**4 ; C**	**8 ; C**		**∞ ; -**		**3 ; E**	**X,E,C,F**
5		**4 ; C**	**8 ; C**		**6 ; F**			**X,E,C,F,A**
6			**8 ; C***		**6 ; F**			**X,E,C,F,A,D**
7			**7 ; D**					**X,E,C,F,A,D,B**

* 8 ; A kan ook

oefening 3
Controleer voor jezelf hoe de stappen 5, 6 en 7 zijn berekend. Gebruik daarbij de tabel van de bijlage waarop je zelf de stappen afmaakt. Kleur per stap het daarin afgehandelde punt.

Naast de lengte van de kortste route van X naar B geeft de tabel nog meer informatie:

- Van X naar B is het pad met de meeste stappen, en de minimale afstand van X naar B is 7.
- In feite heb je in de tabel meteen het kortste pad vanaf X naar elke knoop in de graaf gevonden.

Bijvoorbeeld de totale (kortste) afstand van X naar A is **4**, het pad loopt via C (V(A) = C), en E, (want V (C)=E) naar X (want V(E) =X).

oefening 4
Hoe haal je uit de tabel de lengte van het kortste pad van X naar D?

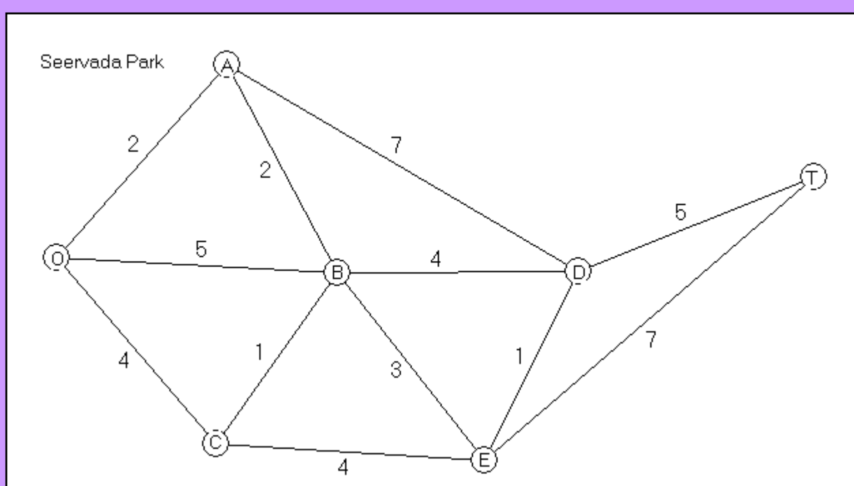
oefening 5

- **a** Probeer uit de tabel te achterhalen hoe het kortste pad van X naar B loopt..
- **b** Teken met kleur het op deze manier gevonden kortste pad in de figuur op de bijlage.
  Vergelijk dit pad met je antwoord bij vraag **1**.
- **c** Als je in stap 3 voor F gekozen had, hoe gaat de tabel er dan uitzien?
- **d** Verklaar waarom je in beide gevallen tot hetzelfde kortste pad komt.

Bekijk het netwerk dat hieronder getekend is.
Het komt uit Hillier & Lieberman [2]. Het staat ook op bijlage 2.

oefening 6

- **a** Kleur het kortste pad van O naar T door je gezonde verstand te gebruiken.
  Dat moet nog lukken.
  Hoe lang is het?
- **b** Maak een tabel en pas het algoritme van Dijkstra toe op het netwerk dat hieronder staat.
  Vind je dezelfde oplossing?



- **c** Maak een afstanden netwerk met 7 tot 9 knopen voor een medeleerling.
  Zelf krijg je er een van een ander.
- **d** Zoek het kortste pad in dat netwerk met het algoritme van Dijkstra.

De methode uit Hillier en Lieberman

(gebruik bij deze paragraaf bijlage 3)

Neem de tekst uit Hillier en Lieberman er bij. Lees paragraaf 9.3 op p. 380 en p. 381 eerst eens door. Je merkt dat zij een iteratief proces beschrijven waarmee het kortste pad gevonden kan worden.

Vul de antwoorden van opdrachten 7 t/m 11 in op het antwoordenblad (bijlage 3).

oefening 7
Hoe noemen H&L de afgehandelde knopen?

Neem het netwerk van Seervada Park er weer bij (zie bijlage 2).
Gezocht wordt het kortste pad van O naar T.
Om het proces goed te begrijpen moet je op een leeg blad beginnen met punt O. Daarna teken je de verbindingslijntjes die naar het kortste pad leiden stukje bij beetje erbij.
Gebruik de **TABLE 9.2** van p. 381 om te snappen hoe ze het doen.
Maak zelf een lege tabel met 7 kolommen en vul die gaandeweg mee in.
Probeer het eerst aan de hand van de Engelse tekst;als dat niet lukt kun je hieronder lezen hoe het werkt.

Het gaat als volgt:
Start in O en zoek de knoop die de kortste verbinding heeft met O.
Dit is knoop A met afstand 2 tot O.
In de tabel is dit de regel achter 1onder ***n***. Je netwerk van "afgehandelde" knopen bestaat nu uit O en A met hun verbindingslijntje.
Nu moet het volgende telkens herhaald worden, en de tabel worden bijgewerkt:
Zoek vanuit het deelnetwerk van de al "afgehandelde" knopen, bij elke "afgehandelde" knoop die in het totale netwerk nog met een of meer niet "afgehandelde" knopen verbonden is, de niet "afgehandelde" knoop met de kortste verbinding.
Als je de hele tabel hebt ingevuld (en ook hebt begrepen hoe het is gedaan) dan kun je uit de laatste kolom het kortste pad terugvinden. Als je goed kijkt kun je zelfs twee verschillende kortste paden vinden (wel beide met lengte 13).

oefening 8
- **a** Probeer de tabel zelf te maken aan de hand van de figuur. Schrijf de kortste routes op met behulp van de juiste verbindingslijntjes. Heb je er twee?
- **b** Op welke manier zie je in deze tabel de kortste afstand van O tot elke andere knoop?
  Bedenk dat knoop C niet in het kortste pad zit.

oefening 9
Vergelijk het algoritme van H&L met dat van Dijkstra.
Kun je het verschil aangeven tussen beide methoden?

oefening 10
Maak een lege tabel met 7 kolommen waarmee je volgens de methode van H&L het kortste pad van X naar B gaat berekenen in figuur 1 van blz 1.

oefening 11
Maak nu de volgende opgaven uit H&L uit de paragraaf Problems van p. 428.

- **a** Maak 9.3-1. (a) en (b)
- **b** Maak 9.3-2. (a) en (b)
- **c** Maak 9.3-3. (a) en (b)
- **d** Maak 9.3-5. (a) en (b)
- **e** Maak 9.3-6. (a) en (b)

De methode uit Tijms

Lees van H. Tijms p. 144 en 146 t/m p. 151 door.

oefening 12

- **a** Zoek het kortste pad van N_1 naar N_7 in figuur 3.1.
- **b** Op welke manier worden afgehandelde punten bijgehouden?
- **c** Wat wordt bedoeld met iteraties (p. 147)?
- **d** Wat wordt bedoeld met initialisatie (p. 149)?
- **e** Wat wordt er beweerd over het meest betrouwbare pad?
- **f** Hoe moeten de formules van p. 150 worden toegepast op tabel 3.2?
- **g** Voer ook zelf de berekeningen uit.
- **h** Op welke manier is tabel 3.3 berekend uit tabel 3.2?
- **i** Teken het netwerk bij tabel 3.3 en bereken zelf met het algoritme van Dijkstra het kortste pad in dat netwerk.
- **j** Waarom is het kortste pad ook het meest veilige pad?

Varianten

Er zijn ook netwerken waarin over de takken tussen de knopen éénrichtingsverkeer geldt.
Er ook nog negatieve gewichten bij de takken staan.
Zoek in de literatuur op welke manier kortste paden in dergelijke gevallen gevonden worden.

Antwoorden oefeningen

Oefening 1
pad XEFDB, met lengte 2+1+3+1 = 7

Oefening 2
L(A)wordt 4; met tweede label C, *L*(B) wordt 8; met tweede label C, *L*(F) = 3 is het kleinste, dus F is afgehandeld

Oefening 3
zie de tabel onder vraag 3

Oefening 4
in stap 6 zie je *L*(D) = 6

Oefening 5
Van achter naar voor redeneren: laatst afgehandelde punt is B, dus pad eindigt in B
kijken in kolom onder B geeft 7;D , dus kortste pad komt van D
kijken in kolom onder D geeft 6;F , dus kortste pad komt van F (met DB = 7 – 6 = 1)
kijken in kolom onder F geeft 3;E , dus kortste pad komt van E (met FD = 6 – 3 = 3)
kijken in kolom onder E geeft 2;X , dus kortste pad is X – E – F – D – B (met XE = 3) ; totale lengte 3+3+1=7
Is je gekleurde pad hetzelfde?

Oefening 6
kortste pad van O naar T in Seervada Park met algoritme van Dijkstra

	O	A	B	C	D	E	T	afgehandelde punten
stap	*L*(O) ; *V*	*L*(A) ; *V*	*L*(B) ; *V*	*L*(C) ; *V*	*L*(D) ; *V*	*L*(E) ; *V*	*L*(T) ; *V*	
1	**0** ; O	∞ ; -	∞ ; -	∞ ; -	∞ ; -	∞ ; -	∞ ; -	O
2		**2** ; O	5 ; O	4 ; O	∞ ; -	∞ ; -	∞ ; -	O, A
3			**4** ; A	4 ; O	9 ; A	∞	∞ ; -	O, A, B
4				**4** ; O	8 ; B	7 ; B	∞ ; -	O, A, B, C
5					8 ; B	**7** ; B	∞ ; -	O, A, B, C, E
6					**8** ; E *		14 ; E	O, A, B, C, E, D
7							**13** ; D	O, A, B, C, E, D, T

* 8 ; B kan ook

Oefening 7
solved nodes

Oefening 8

a in de kolom met Last Connection kun je van beneden naar boven de lijnstukken aan elkaar "knopen";
v.r.n.l. wordt dit OA ←– AB ←– BE ←–ED ←– DT of OA ←– AB ←– BD ←– DT

b OC = 4,
[verder kun je in het kortste pad van O naar T alle kortste afstanden vanaf O tot de overige knopen vinden]

Oefening 9

Bij Dijkstra is elke stap een volgend afgehandeld punt, met de kortste afstand tot dusver naar het startpunt; bij H&L is elke volgende stap een lijntje, met de kortste verbinding van een "afgehandeld" punt (dwz al met lijntjes verbonden naar het startpunt) naar een nog niet afgehandeld punt.

Bij het algoritme van Dijkstra is het laatst afgehandelde punt (P) het punt met de tot dusver kortste afstand (tot S =startpunt).
In de volgend stap wordt van alle punten (X) die nog niet "afgehandeld" zijn, en direct met P verbonden zijn, de kortste afstand (tot S) berekend: dit is dan ofwel de afstand die al eerder naar dat punt (X) berekend is, of een zo mogelijk kortere afstand via P (= de in de vorige stap gevonden (kortste) afstand van S tot P plus PX).
Van de punten X waarvan in deze stap de (nieuwe) afstanden zijn berekend, kies je het punt met de kortste afstand (Q); dit punt is daarmee het volgende afgehandelde punt.

H&L werken als volgt:
Teken (kleur) achtereenvolgens het lijntje in het netwerk met een kortste afstand vanuit alle "afgehandelde" punten naar punten nog niet zijn afgehandeld en die rechtstreeks verbonden zijn met een van de afgehandelde punten. Anders gezegd:
In elke volgende stap worden vanuit alle punten die al met lijntjes aan het startpunt verbonden zitten, en waarvandaan een rechtstreekse verbinding gemaakt kan worden met een punt dat er nog niet aan vast zit, de totale afstanden naar de "nieuwe" punten berekend. Het lijntje waarmee de totale afstand het kortst is wordt dan gekozen om aan het eerder gevormde netwerk vast te knopen. De punten die in het verbonden (gekleurde) netwerk zitten zijn "afgehandeld".

Oefening 10

n	Solved Nodes Directly Conn to Uns Nodes	Closest Connected Unsolved Node	Total Distance Involved	*n* th Nearest Node	Minimum Distance	Last Connection
1	X	E	2	E	2	XE
2	X E	C C	4 2 + 1 = 3	C	3	EC
3	X C E	A A F	5 3 + 1 = 4 2 + 1 = 3	F	3	EF
4	A C F	B B D	4 + 4 = 8 3 + 5 = 8 3 + 3 = 6	D	6	FD
5	F D	B B	3 + 6 = 9 6 + 1 = 7	B	7	DB

Oefening 11

a

9.3-1. (a) Maak een netwerk met O (Origin), A, B, C, D, E en X (Destination); trek verbindingslijntjes tussen de punten waarvan een afstand bekend is; zet bij die lijntjes de gegeven afstanden

(b) het (enige) kortste pad : O – A – B – E – D – X ; lengte 40+10+40+10+60 = 160

b

9.3-2. (a) Maak een netwerk met 0, 1, 2, en 3; trek de zes verbindingslijntjes en zet de getallen erbij

(b) Het (enige) kortste pad is 0 – 1 – 3 ; lengte 8 + 21 = 29.000

c

9.3-3. (a) O – A – B – D – T ; lengte 4+1+5+6 = 16; andere route O – A – B – E – D – T ; lengte 4+1+4+1+6 = 16

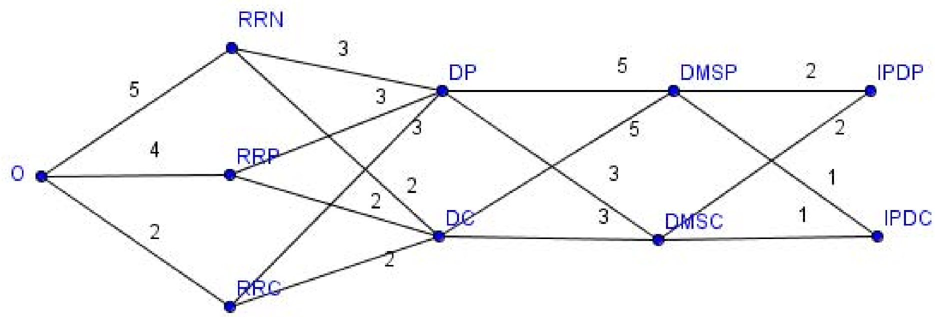
(b) O – C – F – G – T ; lengte 6+2+2+7 = 17

d

9.3-5. (a) de totale vliegtijd moet zo klein mogelijk zijn

(b) SE – C – E – LN ; lengte 4.2 + 3.5 + 3.6 = 11.3

e
9.3-6. (a)



(b)) O – RRC – DP – DMSC – IPDP ; 2+3+3+2 = 10 komt op 9+6+12+3 = 30 miljoen
(O – RRC – DP – DMSC – IPDC is korter maar komt boven de 30 miljoen)

Opgave 12
Zoek het kortste pad van N_1 naar N_7 in figuur 3.1.
N1 – N4 – N5 –N7 ; lengte 3+5+4 = 12

Op welke manier worden afgehandelde punten bijgehouden?
Met een *, het label in de eerste kolom is dan afgehandeld; in het voorbeeld is dus N4 als eerste afgehandeld

Wat wordt bedoeld met iteraties (p. 147)?
herhaald stapsgewijs

Wat wordt bedoeld met initialisatie (p. 149)?
De beginsituatie vaststellen; index *s* is het eerste knooppunt (meestal neem je daarvoor index "$_1$" , dus startpunt N_s is N_1);
De labels worden in eerste instantie tussen haakjes (directe afstand startpunt tot knooppunt 1 ; knooppunt 1) gelabeld,
waarbij punten die niet direct verbonden zijn met het startpunt afstand "oneindig" [T: een heel groot getal] genomen wordt.

Wat wordt er beweerd over het meest betrouwbare pad?
Het pad waarbij de kans dat het bericht niet verloren gaat het grootst is

Hoe moeten de formules van p. 150 worden toegepast op tabel 3.2? Voer ook zelf de berekeningen uit.
Omdat $\alpha(P)$ een product (vermenigvuldiging) is van alle kansen $1 - p_{i,j}$, is de logaritme van $\alpha(P)$ een optelling van de
logaritmes van die kansen. Denk aan $\log(a \cdot b) = \log a + \log b$. Let op! Met logaritme wordt hier ln bedoeld.

Verlieskans van N_1 naar N_2 is 0,03 dus 1 – 0,03 = 0,97 ; ln0,97 = 0,0346, dus neem je c_{12} = −ln0,97 = 0,0305.
Zo ook voor de verlieskans van N_1 naar N_3 is 0,05 geeft −ln0,95 = 0,0513 .
Enzovoorts.

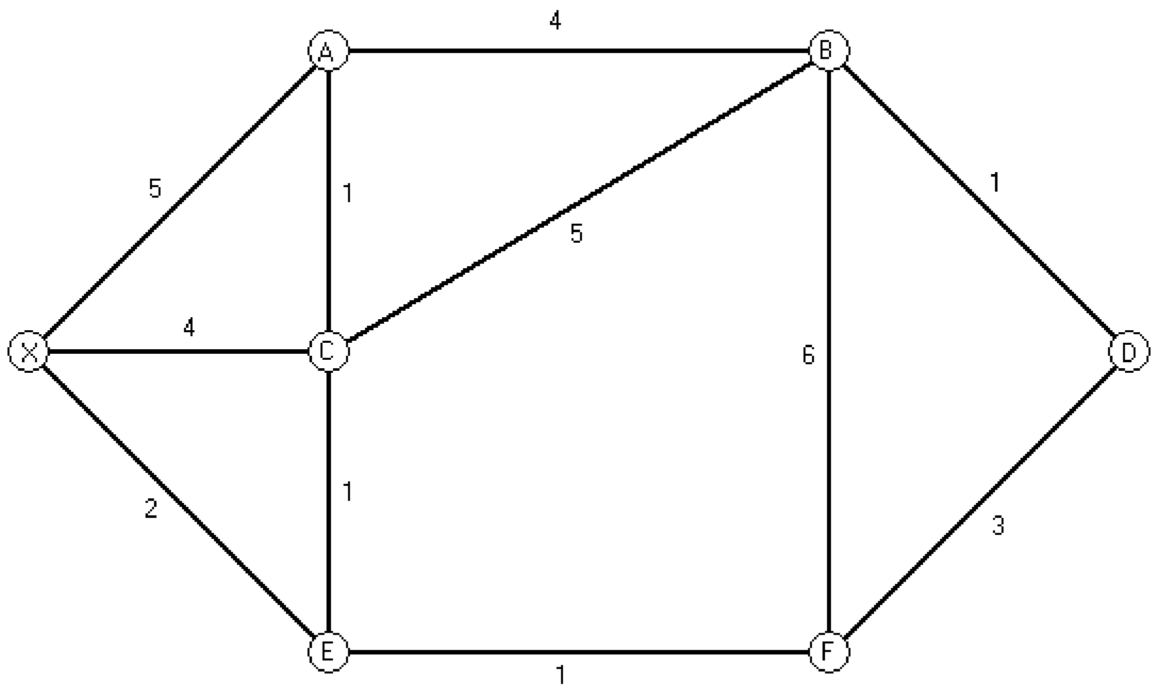
Voor pad P N_1 - N_2 – N_4 – N_3 – N_5 geldt: de kans dat de boodschap niet verloren gaat is $\alpha(P) = 0{,}97 \cdot 0{,}98 \cdot 0{,}98 \cdot 0{,}95$.
Zo is log $\alpha(P)$ = ln0,97 + ln 0,98 + ln0,98 + ln0,95

Op welke manier is tabel 3.3 berekend uit tabel 3.2?
Zie antwoord op de vorige vraag.

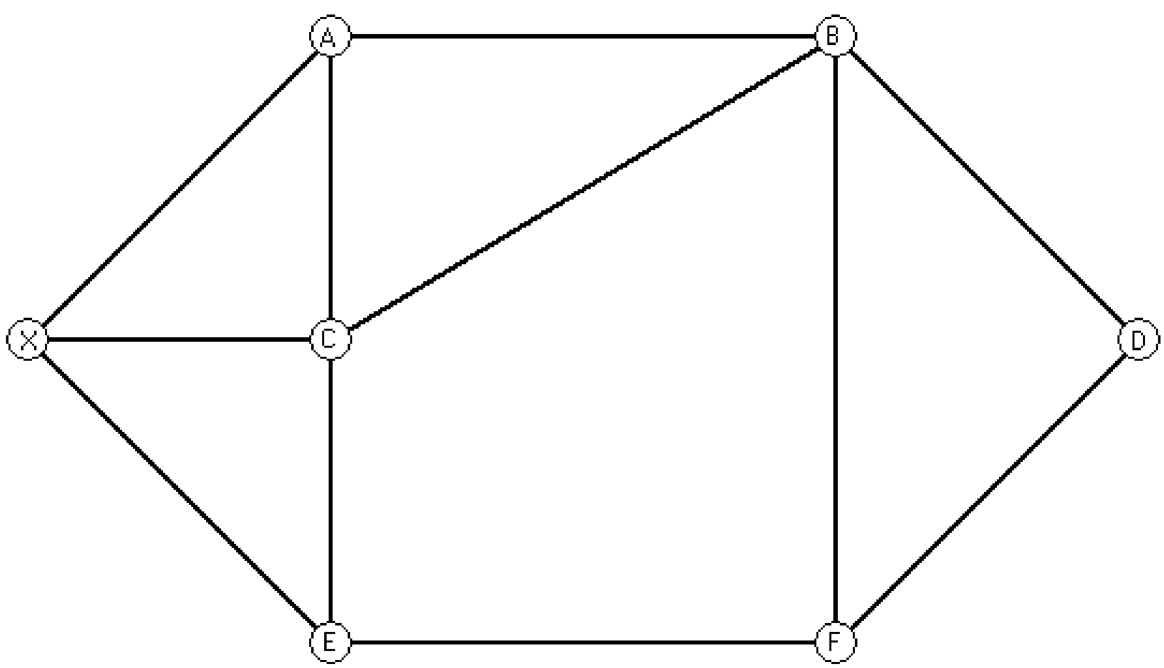
Teken het netwerk bij tabel 3.3 en bereken zelf met het algoritme van Dijkstra het kortste pad in dat netwerk.
N_1 – N_2 – N_5 ; lengte 0,0305 + 0,0356 = 0,0661

Waarom is het kortste pad ook het meest veilige pad?
Je hebt hiermee berekend welk pad de grootste kans heeft dat de boodschap niet verloren gaat, dus het meest betrouwbaar.

Bijlage 1: figuren en tabellen bij het algoritme van Dijkstra; oefening 2 tot en met 5



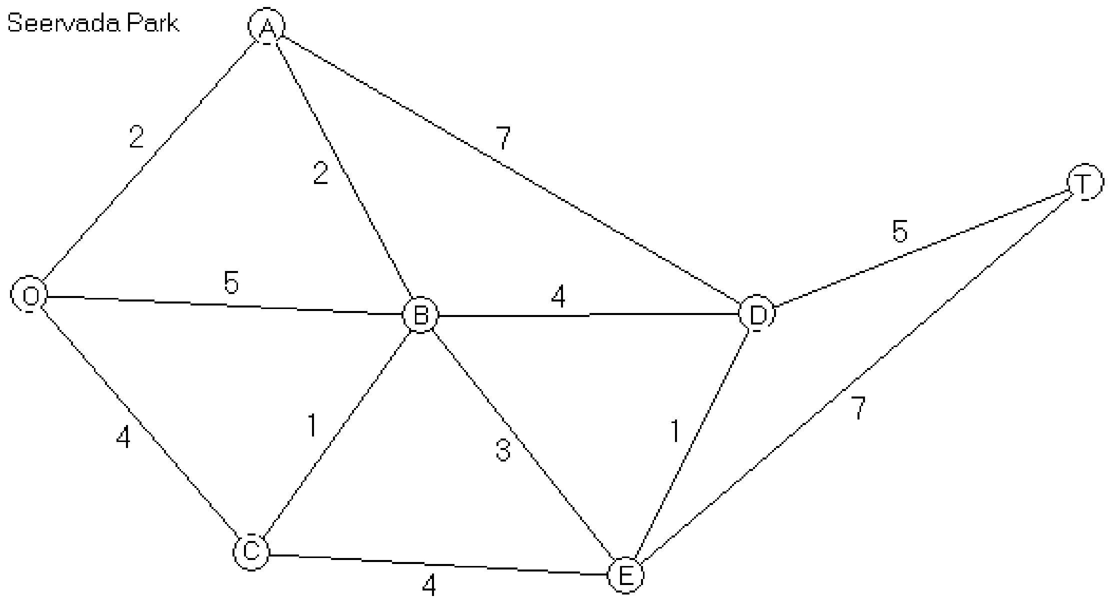
figuur 1



stap	startpunt L; V	X *L*; *V*	A *L*; *V*	B *L*; *V*	C *L*; *V*	D *L*; *V*	E *L*; *V*	F *L*; *V*	afgehandelde knopen
1									
2									
3									
4									
5									
6									
7									
8									
9									

Tabel 1: Kortste pad volgens het algoritme van Dijkstra bij oefening 2 tot en met 5

Bijlage 2: figuur Seervada Park; oefening 6



Bijlage 3: antwoordenblad bij oefening 7 tot en met 11

….. ….. ….. ….. ….. ….. …..

opdracht 7: ….. ….. ….. ….. ….. …. ….. ….. ….. ….. …..
….. ….. ….. …..

opdracht 8 a De twee kortste routes (met afstanden) zijn:
….. …. ….. ….. ….. ….. ….. ….. ….. ….. ….. ….. ….
….. ….. ….. ….. ….. ….. ….. ….. ….. ….. …. ….. …..
….. ….. ….. ….. ….. ….. ….. ….. …. ….. ….. ….. …..
….. ….. ….. ….. ….. ….. …. ….. ….. ….. ….. ….. …..
….. ….. ….. ….. …. ….. ….. ….. ….. ….. ….. ….. …..
….. ….. …. ….. ….. ….. …..

opdracht 8 b
….. …. ….. ….. ….. ….. ….. ….. ….. ….. ….. ….. ….
….. ….. ….. ….. ….. ….. ….. ….. ….. ….. …. ….. …..
….. ….. ….. ….. ….. ….. ….. ….. …. ….. ….. ….. …..
….. ….. ….. ….. ….. ….. …. ….. ….. ….. ….. ….. …..
….. ….. ….. ….. …. ….. ….. ….. ….. ….. ….. ….. …..
….. ….. …. ….. ….. ….. …..

opdracht 9 Schrijf het belangrijkste verschil op tussen beide methoden:
….. …. ….. ….. ….. ….. ….. ….. ….. ….. ….. ….. ….
….. ….. ….. ….. ….. ….. ….. ….. ….. ….. …. ….. …..
….. ….. ….. ….. ….. ….. ….. ….. …. ….. ….. ….. …..
….. ….. ….. ….. ….. ….. …. ….. ….. ….. ….. ….. …..
….. ….. ….. ….. …. ….. ….. ….. ….. ….. ….. ….. …..
….. ….. …. ….. ….. ….. …..

opdracht 10 Tabel op een apart vel inleveren.

opdracht 11 PROBLEMS van H&L Geef op dit blad alleen de kortste route(s) (met afstanden).
Lever de tekeningen en tabellen in op een apart blad. Denk aan namen + opg.nr.!

a 9.3-1 (b) Het (een) kortste pad is : ….. ….. ….. ….. ….. …..
….. …. ….. ….. ….. …..
Zijn er nog meer? ….. ….. ….. ….. ….. …..
….. …. ….. ….. ….. …..
Zo ja, welke dan? ….. ….. ….. ….. ….. …..
….. …. ….. ….. ….. …..
….. ….. ….. ….. ….. …..
….. …. ….. ….. ….. …..

b 9.3-2 (b) Het (een) kortste pad is : ….. ….. ….. ….. ….. …..
….. …. ….. ….. ….. …..
Zijn er nog meer? ….. ….. ….. ….. ….. …..
….. …. ….. ….. ….. …..

Zo ja, welke dan? ….. ….. ….. ….. ….. …..
….. …. ….. ….. ….. …..

c 9.3-3 (a) Het (een) kortste pad is : ….. ….. ….. ….. ….. …..
….. …. ….. ….. ….. …..
Zijn er nog meer? ….. ….. ….. ….. ….. …..
….. …. ….. ….. ….. …..
Zo ja, welke dan? ….. ….. ….. ….. ….. …..
….. …. ….. ….. ….. …..

9.3-3 (b) Het (een) kortste pad is : ….. ….. ….. ….. ….. …..
….. …. ….. ….. ….. …..
Zijn er nog meer? ….. ….. ….. ….. ….. …..
….. …. ….. ….. ….. …..
Zo ja, welke dan? ….. ….. ….. ….. ….. …..
….. …. ….. ….. ….. …..

d 9.3-5 (b) Het (een) kortste pad is : ….. ….. ….. ….. ….. …..
….. …. ….. ….. ….. …..
Zijn er nog meer? ….. ….. ….. ….. ….. …..
….. …. ….. ….. ….. …..
Zo ja, welke dan? ….. ….. ….. ….. ….. …..
….. …. ….. ….. ….. …..

e 9.3-1 (b) Het (een) kortste pad is : ….. ….. ….. ….. ….. …..
….. …. ….. ….. ….. …..
Zijn er nog meer? ….. ….. ….. ….. ….. …..
….. …. ….. ….. ….. …..
Zo ja, welke dan? ….. ….. ….. ….. ….. …..
….. …. ….. ….. ….. …..

Programma voor de TI83 (Plus): het algoritme van Dijkstra

gebaseerd op het globale programma dat prof. Roos in zijn syllabus voor de cursus Caleidoscoop heeft beschreven, zie hoofdstuk F Optimaliseren in netwerken, pagina 5.

De programma's kunnen gemaakt, verbeterd/aangevuld en uitgevoerd worden via resp. PRGM NEW, PRGM EDIT en PRGM EXEC. Bij het editen geldt:
De dubbele punt in het begin van de regel verschijnt bij het editen na ENTER.
Er kan een regel tussengevoegd worden via 2nd INS ENTER.
De dik gedrukte woorden en tekens kunnen uit de CATALOG gehaald worden.
Namen van matrices kunnen ook opgehaald worden via MATRX NAMES.
→ wordt verkregen via de toets STO→.

Hieronder hebben we de regels met (al dan niet bij de uitvoering zichtbaar) commentaar, die zonder bezwaar weggelaten kunnen worden, laten inspringen.

```
PROGRAM: DIJKSTRA
                    : Disp "VOORAF MATRIX C"
                    : Disp "INVOEREN ZO DAT"
                    : Disp "C(I, J)=AFSTAND"
                    : Disp "LANGS TAK VAN"
                    : Disp "KNOOP I NAAR J"
                    : Disp "EN C(I, J)=10^10"
                    : Disp "ALS GEEN TAK."
: dim ([C])→L3
: L3(1)→N
                    : "DE ENEN EN NULLEN IN L1 GEVEN AAN WELKE KNOPEN Q
                      WEL RESP NIET BEVAT"
: N→dim (L1)
: 1→L1(1)
                    : "L2 BEVAT DE п-WAARDEN VAN ALLE KNOPEN"
: N→dim (L2)
: Fill (10^10, L2)
: 0→L2(1)
                    : "IN MATRICES A EN B WORDEN ITERATIES MBT RESP Q EN
                    DE
                      п-WAARDEN BIJGEHOUDEN"
: List►matr (L1, [A])
: List►matr (L2, [B])
                    : "EINDE INITIALISATIE"
: While max(L1)>0
: prgmKIESU
                    (Z.O.Z.!)
: prgmVERKENU
                    (Z.O.Z.!)
: List►matr (L1, [D])
: augment ([A], [D])→[A]
: List►matr (L2, [E])
: augment ([B], [E])→[B]
```

```
: End
                    : Disp "LENGTEN KORTSTE"
                    : Disp "PADEN NAAR DE"
                    : Disp "KNOPEN ZIJN RESP"
: L2
```

de programma's KIESU en VERKENU die gebruikt worden in het programma DIJKSTRA.

```
PROGRAM: KIESU
                    : "ZO WORDT U IN Q MET KLEINSTE π-WAARDE GEKOZEN
                      (DIJKSTRA)"
: 10^10→P
: For (V, 1, N, 1)
: If  L1(V)=1
: Then
: If  L2(V)<P
: Then
: V→U
: L2(V)→P
: End
: End
: End
```

```
PROGRAM: VERKENU
                    : "ZO WORDT U VERKEND EN UIT Q VERWIJDERD (DIJKSTRA)"
: For (V, 1, N, 1)
: If  L2(V)>L2(U)+[C](U,V)
: Then
: L2(U)+[C](U,V)→L2(V)
: 1→L1(V)
: End
: End
: 0→L1(U)
```

Voordat het programma DIJKSTRA uitgevoerd gaat worden, moeten de gegevens van het netwerk ingevuld worden in de matrix C:
geef eerst de knopen opvolgende nummers 1, 2, 3, ..., N (waarin N een numerieke waarde heeft) zo dat de startknoop nummer 1 heeft en de eindknoop nummer N,
geef de matrix C de afmetingen N bij N (waarin N een numerieke waarde heeft), bijvoorbeeld via {N, N}→ **dim ([**C**])**,
vul de matrix voorlopig met uitsluitend getallen 10^10, bijvoorbeeld via **Fill (**10^10, **[**C**]**),
verander daarna voor iedere tak van het netwerk die van een knoop i naar een knoop j gaat, het getal 10^10 dat in de i-de rij en de j-de kolom van de matrix C staat mbv MATRX EDIT [C] in de afstand van knoop i naar knoop j via die tak.

Aanbevolen opties onder Mode: Normal en, als de afstanden in het netwerk geheel zijn, Float 0.

Na uitvoer van het programma kunnen de matrices A en B opgevraagd worden, via **[**A**]** resp. **[**B**]**, om aan de hand daarvan te bepalen via welke knopen het gevonden kortste pad loopt.

Toetsopgaven

Hieronder staan twee **netwerken**.

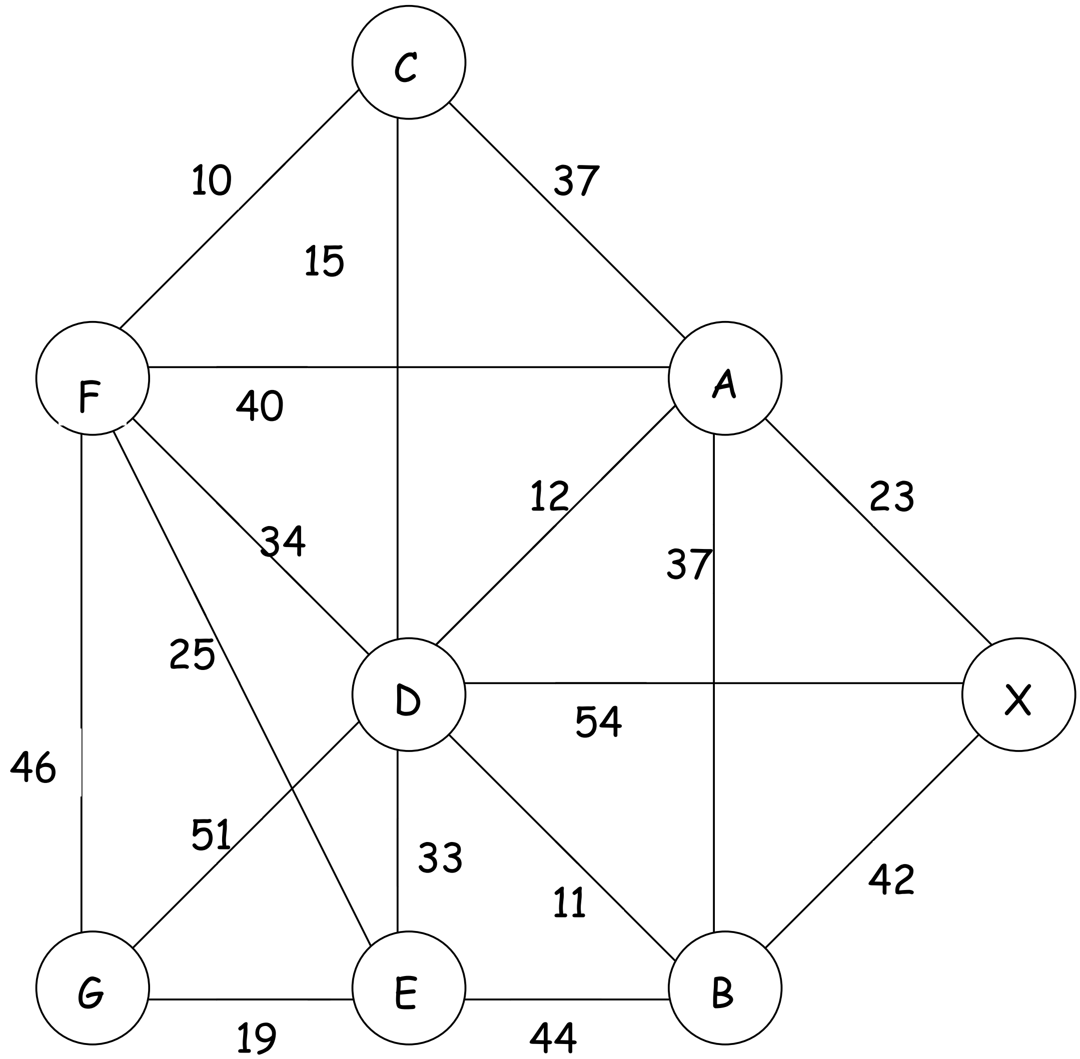
Netwerk 1 en netwerk 2 zijn gebruikt voor een toets.
Voor beide vragen staan antwoordbladen verderop en ook uitwerkingen.

In de netwerken kun je zelf veranderingen aanbrengen in de letters en de afstanden bij de verbindingen.

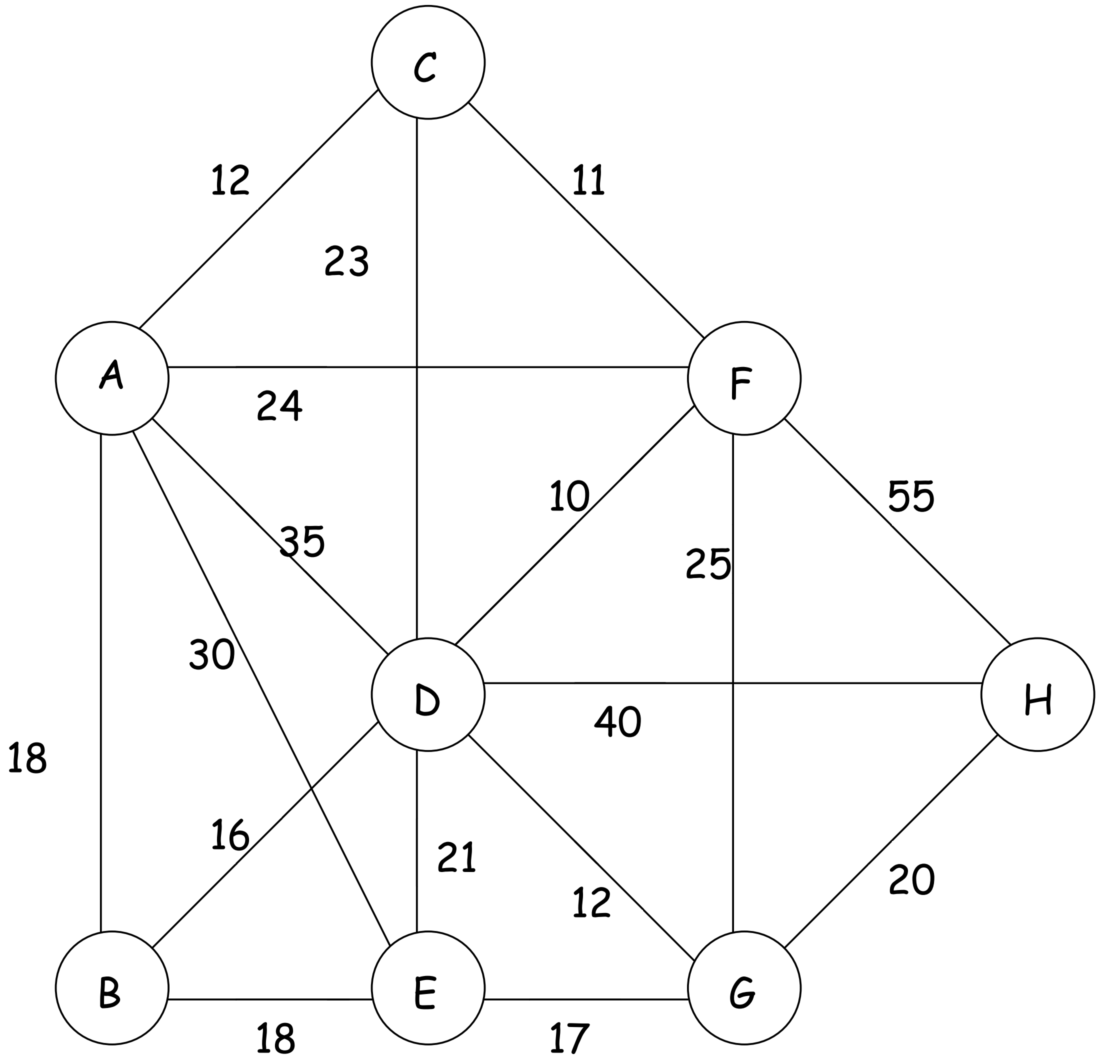
Dit netwerk is een WORD document, als volgt gemaakt:

- eerst zijn met AutoVormen in Basivormen twee vierkanten op hun kant getekend,
- vervolgens alle lijnen met uit de Tekenbalk
- daarna met AutoVormen in Basisvormen op alle hoekpunten een cirkel en daaroverheen een Tekstvak
- in het tekstvak is met Comic Sans MS en grootte 16 een letter gezet, daarna via klik met de rechtermuisknop op de rand van het tekstvak (er verschijnt een menu) gekozen voor Tekstvak opmaken
- daar in Kleur: gekozen voor Geen lijn
- vervolgens in tekstvakken alle afstanden bij de verbindingslijntjes gezet
- omdat er nu lijnen overschreven worden heb ik met kleine lijntjes de gaten boven op de tekstvakken gedicht

Netwerk 1



Netwerk 2



Toets
Optimaliseren in netwerken module 2
Het algoritme van Dijkstra Naam: ....

Opgave 1

Leg het blaadje met netwerk 1 voor je neer. Schrijf je naam er op.
Neem een leeg invulschema van het algoritme van Dijkstra erbij. Schrijf ook daar je naam op.

a Vul het schema in volgens het algoritme van Dijkstra vanaf startpunt ***X***; omcirkel telkens in elke stap welk punt (met de tot dan toe kortste afstand) als afgehandeld wordt beschouwd.

b Geef met letters het "langste" kortste pad in netwerk 1 an ***X*** af; schrijf de letters, de deelafstanden en de totale afstand op:

. . . .

. . . .

. . . .

c Hoe vind je het kortste pad van ***X*** naar ***F*** in het invulschema?
Hoe lang is dat en via welke punten loopt het?

. . . .

. . . .

. . . .

Opgave 2

Leg het blaadje met netwerk 2 voor je neer. Schrijf je naam er op.
Neem een leeg invulschema van de methode van Hillier & Lieberman erbij. Schrijf ook daar je naam op.

a Vul het schema volgens de methode van H & L in vanaf punt ***A***; kleur bij elke stap de Last Connection in het netwerk.

b Geef met letters het "langste" kortste pad in netwerk 2 van ***A*** af; schrijf de letters, de deelafstanden en de totale afstand op:

. . . .

. . . .

. . . .

Invultabel Kortste pad volgens het algoritme van Dijkstra

Naam: . . . .

stap	startpunt								afgehandelde knopen
	L; V	*L*; *V*	*L*; *V*	*L*; *V*	*L*; *V*	*L*; *V*	*L*; *V*	*L*; *V*	
1									
2									
3									
4									
5									
6									
7									
8									
9									

Invultabel opgave 1 netwerk 1 *Kortste pad volgens het algoritme van Dijkstra*

Naam: ANTWOORD

stap	X (startpunt)	A	B	C	D	E	F	G	afgehandelde knopen
	L; *V*	*L*; *V*	*L*; *V*	*L*; *V*	*L*; *V*	*L*; *V*	*L*; *V*	*L*; *V*	
1	0 ; X	∞ ; -	∞ ; -	∞ ; -	∞ ; -	∞ ; -	∞ ; -	∞ ; -	X
2		23 ; X	42 ; X	∞ ; -	54 ; X	∞ ; -	∞ ; -	∞ ; -	X, A
3			~~60 ; A~~ 42 ; X	60 ; A	35 ; A	∞ ; -	63 ; A	∞ ; -	X, A, D
4			~~46 ; D~~ 42 ; X	50 ; D		68 ; D	~~69 ; D~~ 63 ; A	86 ; D	X, A, D, B
5				50 ; D		~~86 ; B~~ 68 ; D	63 ; A	86 ; D	X, A, D, B, C
6						68 ; D	60 ; C	86 ; D	X, A, D, B, C, F
7						~~85 ; F~~ 68 ; D		~~106 ; F~~ 86 ; D	X, A, D, B, C, F, E
8								86 ; D	X, A, D, B, C, F, E, G
9									kortste pad: XADG lengte: 23+12+51=86

Invultabel Kortste pad volgens het algoritme uit Hillier en Lieberman

Naam: . . . .

n	Solved Nodes Directly Conn to Uns Nodes	Closest Connected Unsolved Node	Total Distance Involved	***n*** th Nearest Node	Minimum Distance	Last Connection
1						
2						
3						
4						
5						
6						
7						

Invultabel opgave 1 netwerk 1

Kortste pad volgens het algoritme uit Hillier en Lieberman Naam: ANTWOORD

n	Solved Nodes Directly Conn to Uns Nodes	Closest Connected Unsolved Node	Total Distance Involved	*n* th Nearest Node	Minimum Distance	Last Connection
1	A	C	12	C	12	AC
2	A C	B F	18 12+11=23	B	18	AB
3	A B C	F D F	24 18+16=34 12+11=23	F	23	CF
4	A B C F	E D D D	30 18+16=34 12+23=35 23+10=33	E	30	AE
5	A B C E F	D D D G D	35 34 35 30+17=47 23+10=33	D	33	FD
6	D E F	G G G	33+12=45 30+17=47 23+25=48	G	45	DG
7	F G	H H	23+55=78 45+20=65	H	65	GH
			kortste pad : AC+CF+FD+DG+GH= 12 +11 +10 + 12 +20 = 65			