

Materiaal voor de les

- Papier/Pen voor oefeningen
- Tijdens oefeningen verwacht ik dat je nadenkt en probeert de flowcharts te tekenen en oefeningen te programmeren

T4.5 Dijkstra (kortste pad)

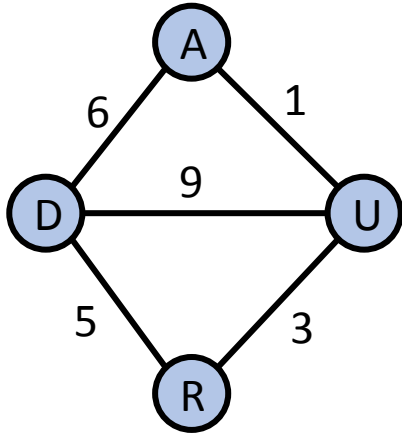
Mnr. Segers

Hoe vindt Google Maps de kortste route?



08-06-2023

Opslaan van een routekaart in computergeheugen

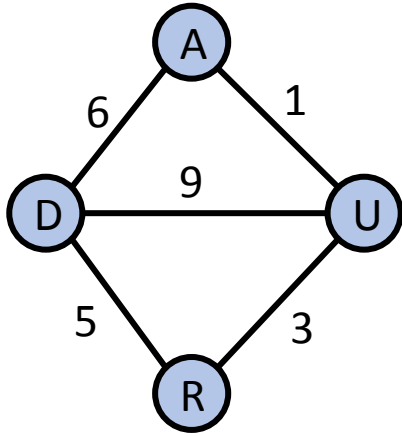


Startpunt	Eindpunt	Reistijd
Amsterdam	Den Haag	6
Amsterdam	Utrecht	1
Den Haag	Rotterdam	5
Den Haag	Utrecht	9
Rotterdam	Utrecht	3

Een routekaart kun je modelleren als een **graaf**. Een graaf bevat

- **Knopen** (denk aan steden of kruispunten)
- **Takken** (Denk aan wegen tussen steden of kruispunten)
- **Gewicht** (Denk aan reistijd of afstand per weg)
- **Pad** (Denk aan route over 1 of meerder takken van een startpunt naar een eindpunt)

Opslaan van een routekaart in computergeheugen

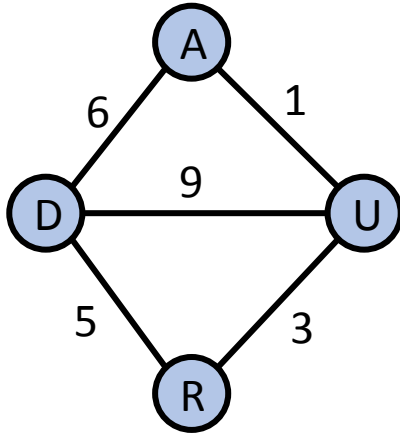


Startpunt	Eindpunt	Reistijd
Amsterdam	Den Haag	6
Amsterdam	Utrecht	1
Den Haag	Rotterdam	5
Den Haag	Utrecht	9
Rotterdam	Utrecht	3

In ons eenvoudige voorbeeld geldt:

- Maximaal 1 directe weg tussen twee steden
- Heenweg duurt net zolang als terugweg
- Geen informatie over soort weg, locatie van stad enzovoort

Algoritme 1: Kies steeds meest dichtbijge buur



Algoritme:

- Ga steeds over een weg naar een naast-liggende stad waar je nog niet geweest bent, kies daarbij de stad die achter de kortste weg ligt.
- Als je in alle naastliggende steden geweest bent en je hebt het eindpunt nog niet gevonden, ga dan steeds een stap terug, totdat je weer naar een naastliggende stad kunt waar je nog nooit bent geweest.
- Als je in het eindpunt bent, dan ben je klaar

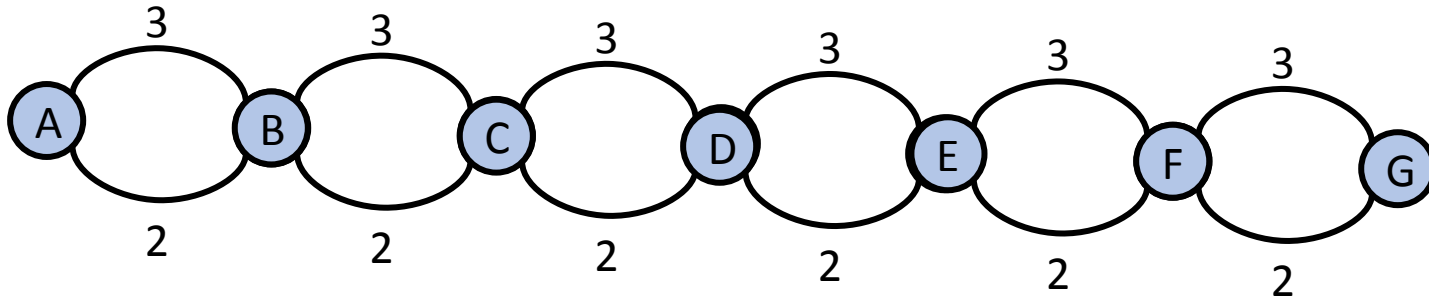
Dit algoritme is niet goed, want het vindt niet altijd het kortste pad.

- Probeer het algoritme maar eens met de route van Den Haag naar Utrecht.
- Het algoritme kiest voor D-R-U, terwijl D-A-U korter is

Algoritme 2: Probeer alle mogelijkheden

Dit algoritme is niet efficiënt.

- Je moet heel veel mogelijkheden proberen.
- Het onderstaande voorbeeld heeft alleen voor een pad van 6 takken al 64 mogelijkheden.



Edsger Wybe Dijkstra (1930-2002)

- Nederlandse wiskundige en informaticus
- Geboren in Rotterdam
- Bedacht het kortste-pad-algoritme (1959)

What's the shortest way to travel from Rotterdam to Groningen? It is the algorithm for the shortest path, which I designed in about 20 minutes. One morning I was shopping in Amsterdam with my young fiancée, and tired, we sat down on the café terrace to drink a cup of coffee and I was just thinking about whether I could do this, and I then designed the algorithm for the shortest path. As I said, it was a 20-minute invention. In fact, it was published in 1959, three years later. The publication is still quite nice. One of the reasons that it is so nice was that I designed it without pencil and paper. Without pencil and paper you are almost forced to avoid all avoidable complexities. Eventually that algorithm became, to my great amazement, one of the cornerstones of my fame. I found it in the early 1960s in a German book on management science—"Das Dijkstra'sche Verfahren" ["Dijkstra's procedure"]. Suddenly, there was a method named after me. And it jumped again recently because it is extensively used in all travel planners. If, these days, you want to go from here to there and you have a car with a GPS and a screen, it can give you the shortest way.

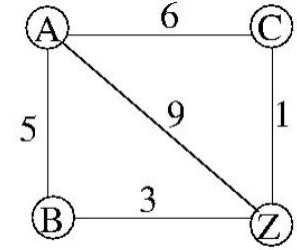
Bron: <https://cacm.acm.org/magazines/2010/8/96632-an-interview-with-edsger-w-dijkstra/fulltext>

Dijkstra's kortste-pad-algoritme (2)

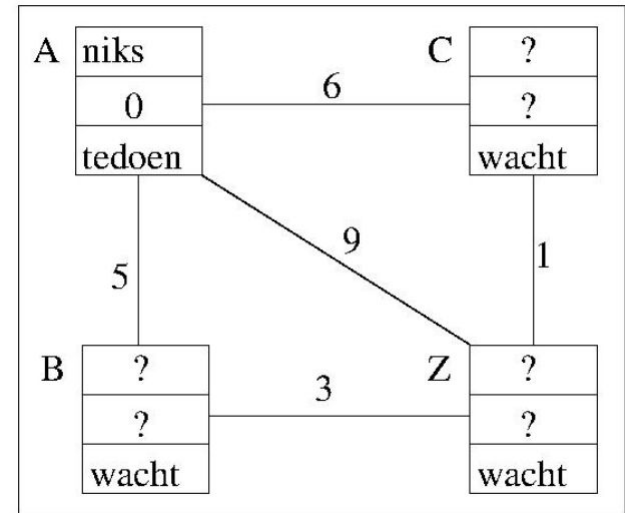
Kies de TeDoen knoop met de kortste afstand tot A □ kies A

Bekijk alle buurkopen

- Als status van buurknoop is Klaar: sla deze buurknoop over
- Als status van buurknoop is TeDoen: Werk info (vorige knoop en afstand) van buurknoop bij als de route vanaf de huidige knoop korter is
- Als status van buurknoop is Wacht, verander status van buurknoop in teDoen en vul info (vorige knoop en afstand) in



Plaatje: Nadat A klaar is



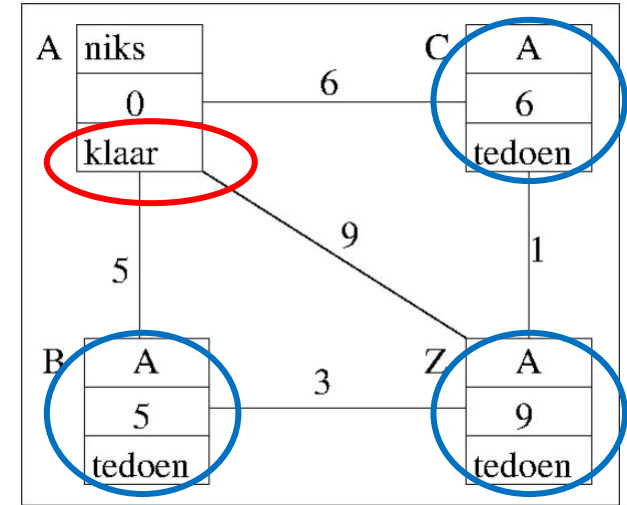
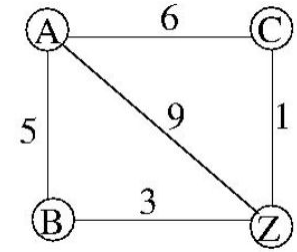
Dijkstra's kortste-pad-algoritme (2)

Kies de TeDoen knoop met de kortste afstand tot A □ kies A

Bekijk alle buurkopen

- Als status van buurknoop is Klaar: sla deze buurknoop over
- Als status van buurknoop is TeDoen: Werk info (vorige knoop en afstand) van buurknoop bij als de route vanaf de huidige knoop korter is
- Als status van buurknoop is Wacht, verander status van buurknoop in teDoen en vul info (vorige knoop en afstand) in

Plaatje: Nadat A klaar is



Dijkstra's kortste-pad-algoritme (3)

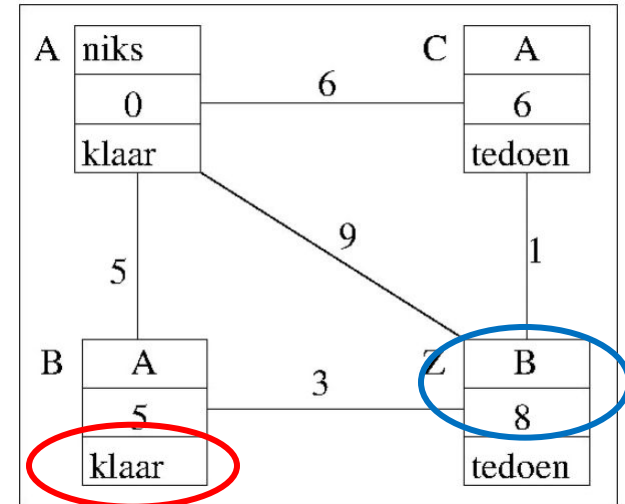
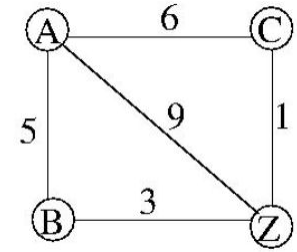
Kies de TeDoen knoop met de kortste afstand tot A □ kies B

Bekijk alle buurkopen

- Als status van buurknoop is Klaar: sla deze buurknoop over
- Als status van buurknoop is TeDoen: Werk info (vorige knoop en afstand) van buurknoop bij als de route vanaf de huidige knoop korter is
- Als status van buurknoop is Wacht, verander status van buurknoop in teDoen en vul info (vorige knoop en afstand) in

Plaatje: Nadat B klaar is,

- Er is een afstand voor Z gevonden: 8 maar die kan nog worden verbeterd



Dijkstra's kortste-pad-algoritme (4)

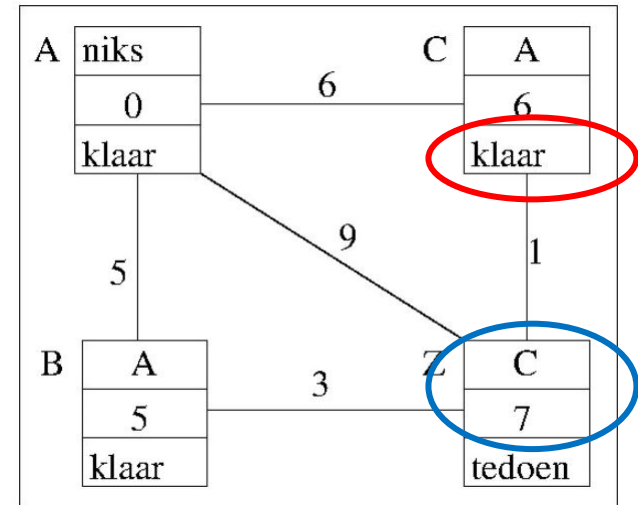
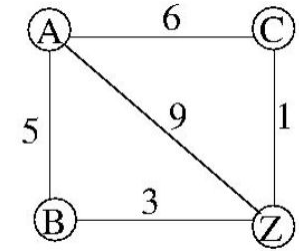
Kies de TeDoen knoop met de kortste afstand tot A □ kies C

Bekijk alle buurkopen

- Als status van buurknoop is Klaar:
sla deze buurknoop over
- Als status van buurknoop is TeDoen
Werk info (vorige knoop en afstand)
van buurknoop bij als de route
vanaf de huidige knoop korter is
- Als status van buurknoop is Wacht,
verander status van buurknoop in teDoen
en vul info (vorige knoop en afstand) in

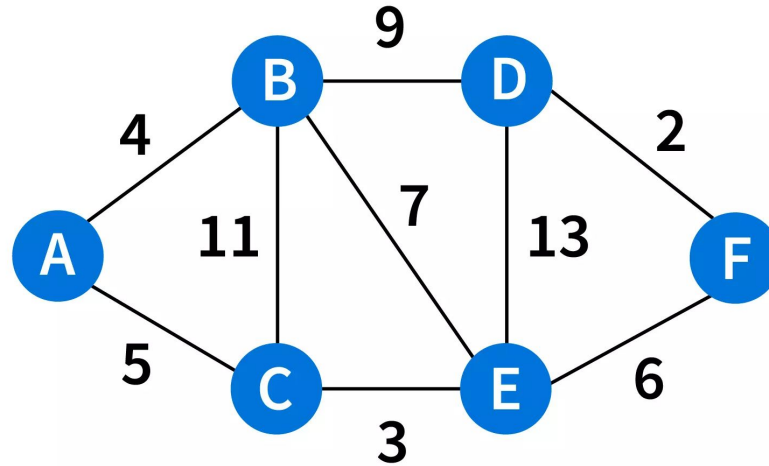
Plaatje: Nadat C klaar is

- Kortste afstand gevonden: 7
- Het kortste (vanaf Z teruglopen: Z □ C □ A



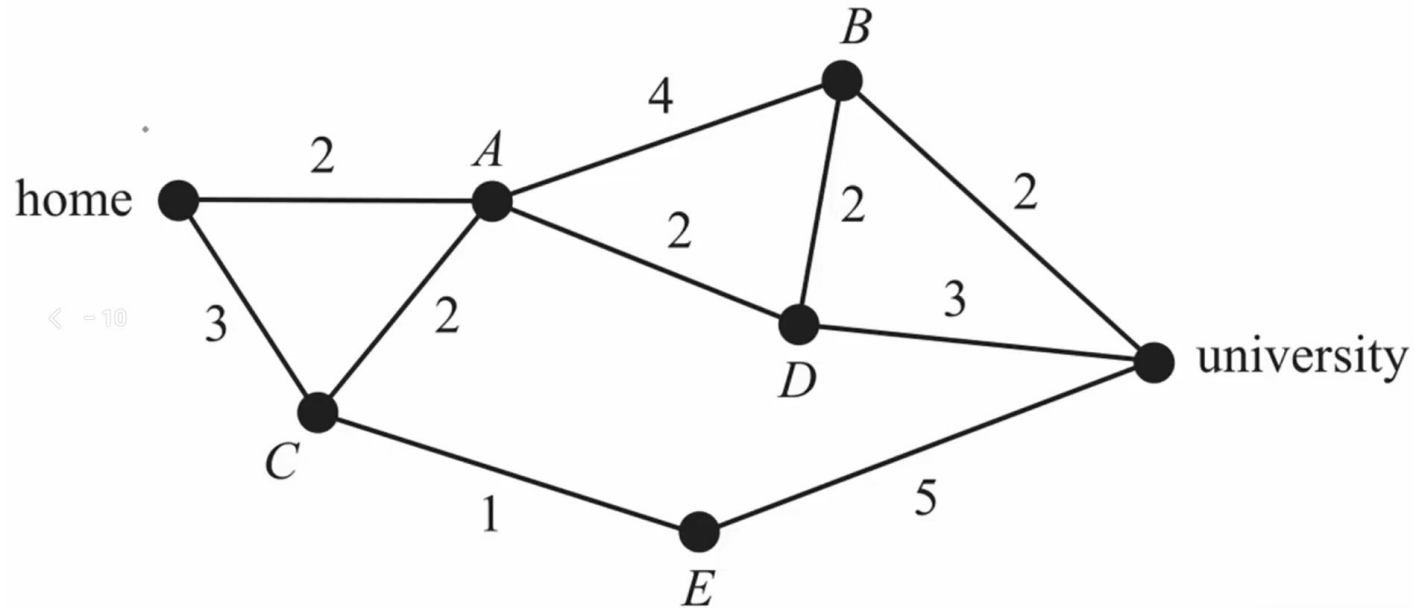
Dijkstra voorbeeld

Schrijf dit voorbeeld mee op!



Dijkstra uitgelegd

<https://www.youtube.com/watch?v=g5OV12u3uHY>



Zelf doen

Zoek de kortste routes vanaf X

