

Heroes in Computer Science



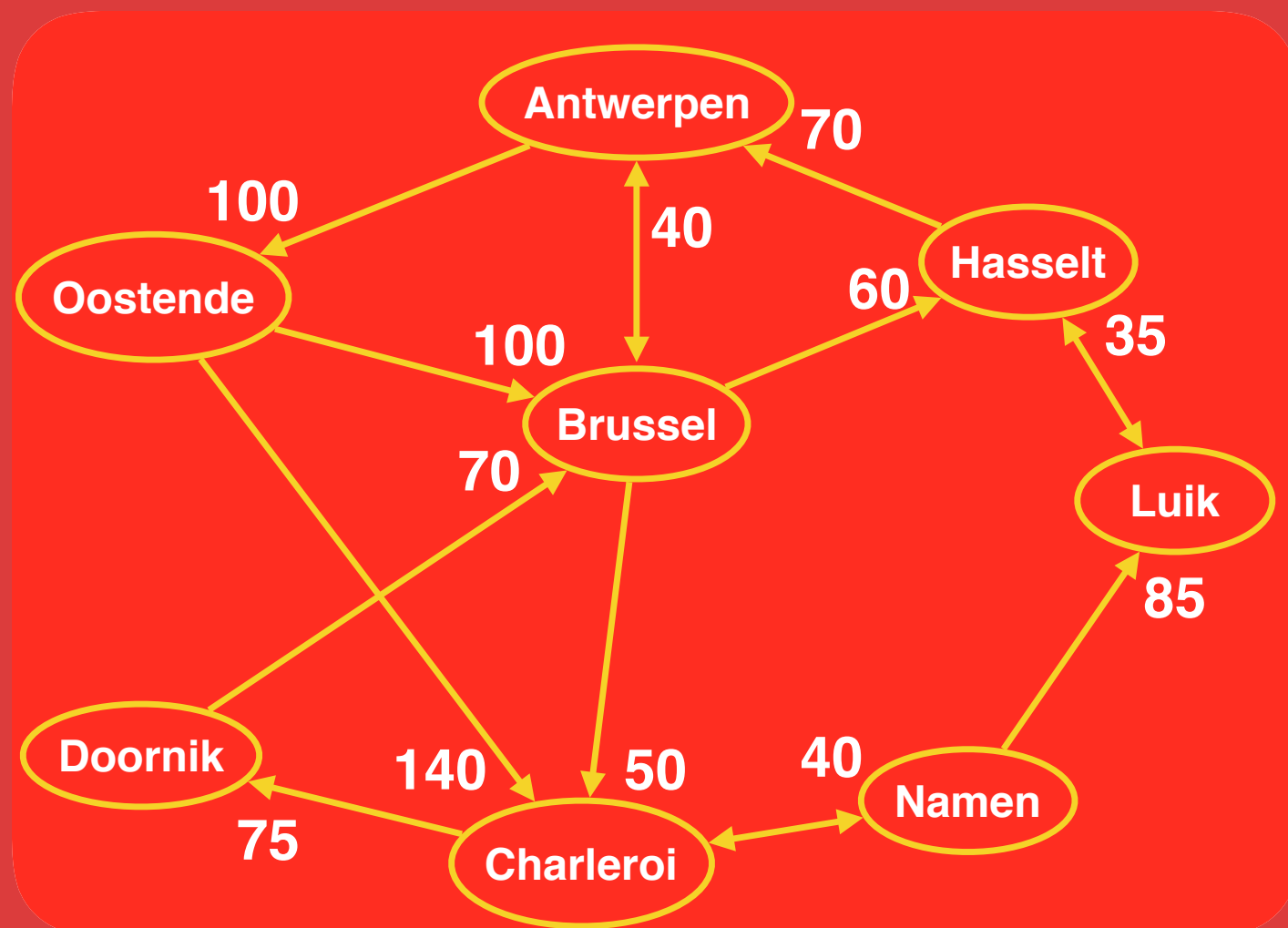
Edsger Dijkstra

1930 - 2002 — Nederland

Edsger Wybe Dijkstra was een Nederlandse informaticus die talloze belangrijke vorderingen heeft gemaakt binnen de informatica. Hij concentreerde zich vooral op de meer wiskundige problemen binnen de computerwetenschappen, maar hij hield zich ook bezig met het ontwerpen van programmeertalen.

GRAFEN

Dijkstra was een informaticus die zich onder andere bezighield met *grafen*. Een graaf is een structuur waar een aantal *knopen* met elkaar verbonden zijn door lijnen. Zulke grafen zijn nuttige structuren waarmee heel wat wiskundige problemen kunnen voorgesteld worden. Zo kunnen ze bijvoorbeeld gebruikt worden in een GPS om de snelste route te berekenen tussen twee locaties. Zo zie je op de graaf hieronder een kleine voorstelling van enkele Belgische steden.



Elke knoop stelt een stad voor. Als er een lijn van de ene stad naar de andere gaat, wil dat zeggen dat je van die eerste stad rechtstreeks naar de andere kan rijden. Let op! De meeste lijnen hebben slechts één richting: je kan in dit plan dus niet rechtstreeks van Antwerpen naar Hasselt rijden. De getallen naast elke lijn duiden aan hoeveel kilometer de twee steden van elkaar verwijderd zijn.

KORTSTE PAD

Dijkstra vroeg zich af of er een gemakkelijke en snelle manier was om in een graaf het kortste pad tussen twee steden te vinden. Hoe zou je bijv. het snelst van Oostende naar Luik kunnen rijden? Probeer het zelf eens uit! Gevonden? Stel je nu voor dat je dit zou moeten doen voor een veel grotere kaart, waar elke stad in België, of zelfs de hele wereld, getoond wordt.



De techniek die Dijkstra bedacht, verloopt ongeveer als volgt:

1. Maak een verzameling van alle steden en hun afstand, die rechtstreeks vanuit je beginstad (bijv. Oostende) te bereiken zijn.
2. Neem dan de stad uit die verzameling die het dichtste bij de beginstad ligt (Brussel). Neem alle steden die van hieruit rechtstreeks te bereiken zijn. (Antwerpen, Charleroi en Hasselt) Tel hun afstand telkens op bij de afstand die je nodig had om van Oostende naar Brussel te geraken en voeg die nieuwe steden, met hun totale afstand, toe aan de verzameling.
3. Nu heb je Brussel nagekeken. Herhaal stap 2, maar dan met de volgende dichtstbijzijnde stad in de verzameling die nog niet is nagekeken. Blijf zo steden nakijken, tot je de eindstad hebt nagekeken.



Scan de code hiernaast om een visualisatie te bekijken van Dijkstra's techniek: