

Workshop: Routeplannen met Wybe

Gerard Tel
Informatica, Univ. Utrecht

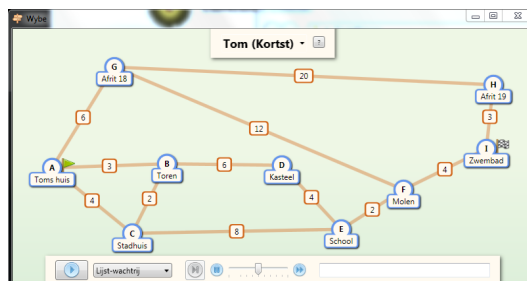
Voorjaar 2012

Wat gebeurt er in een routeplanner, en wat kun je daar allemaal mee doen? Bas den Heijer heeft een demo gemaakt: *Wybe*, en met deze opdrachtjes kun je met Wybe aan de slag.

1 Op pad met Wybe

We beginnen de workshop met zorgen dat alle bestanden die we nodig hebben er zijn.

1. **De Wybe-bestanden krijgen.** Het programma en voorbeeldkaarten staan in een map met naam *RoutePlanner*. Je kunt deze map naar je bureaublad downloaden van de webpagina <http://www.staff.science.uu.nl/~tel00101/Wybe/>, maar misschien staat hij al op je computer of op je memorstick.
2. **Welke bestanden zijn er?** Dubbelklik op de map *RoutePlanner* om te zien welke bestanden erin zitten. Er moet een *Wybe.exe* zijn (met icoon: wegwijzertje) en een paar bestanden waarvan de naam eindigt op *.graph.txt*; dat zijn de kaarten van Wybe.
3. **Heb je een tekst-editor?** Dubbelklik op het bestand *negatief.graph.txt*. Het moet geopend worden in een tekst-editor (zoals NotePad of Kladblok). Sluit het bestand weer.
4. **Welke knoppen heeft Wybe?** Als je het programma *Wybe.exe* opstart (dubbelklik), krijg je een schermje te zien zoals hier is afgebeeld.



Op het scherm zijn deze controls te gebruiken:

- *Kaartkeuze*: Staat in het begin op Tom (Kortst). Hiermee kies je tussen verschillende kaarten. Je kunt ook zelf kaarten maken.
- *Info*: Achtergrond van Wybe.
- *Startknop* en ook *Stop & Resetknop*: Linksonderin, hiermee start je de routeberekening. Na het starten zal deze knop alles weer terugzetten en de berekening resetten.
- *Wachtrij*: Kies (Lijst, Boom, of Mens) hoe de volgende definitieve knoop gekozen wordt. Bij keuze Mens moet je zelf de knoop kiezen door aanklikken.
- *Stapknop en snelheidsschuif*: Zet de schuif helemaal naar rechts voor supersnel rekenen, naar links om het stap voor stap te doen met de stapknop.
- *Uitleg*: Bij elke rekenstap zegt Wybe waar hij mee bezig is.
- *Startvlag*: Klik op de startvlag en dan op een knoop om het beginpunt te kiezen. (De start- en eindvlag kunnen nooit op dezelfde plaats staan.)
- *Eindvlag*: Klik op de eindvlag en dan op een knoop om de bestemming te kiezen.

2 Routes vinden met Wybe

In de volgende opdrachten laat je Wybe een paar routes uitrekenen. Je komt er achter, hoe de gekozen route afhangt van de gegevens in het kaartmateriaal.

5. **Een kortste route.** Als je Wybe hebt opgestart, is de geladen kaart “Tom (Kortst)”. Bij elk weggedeelte staat de lengte (in kilometers). Wat is volgens jou de kortste weg van Toms huis naar het Zwembad?

Je kunt de kortste route door Wybe laten uitrekenen door te drukken op de startknop (links in het onderste bedieningspaneel). Het gaat wel een beetje langzaam... maar door de snelheidschuif (onderin in het midden) naar rechts te schuiven, laat je het rekenen sneller gaan.

Uiteindelijk wordt de kortste route zwart. Is deze route ook wat je zelf als kortste had bedacht?

6. **Waarom stopt Wybe met uitrekenen?** Wybe heeft ook de afstand berekend van Toms huis naar Afrit 18, de Toren en het Kasteel. Bij Afrit 19 staat nu 26, terwijl het (via het Zwembad) maar 21km is. Waarom maakt de routeplanner de berekening niet af?
7. **Hoe ziet een Wybe-kaart er uit?** Dubbelklik in de map *RoutePlanner* op het bestand *atomafstand.graph.txt* om het te openen in een editor (bv. NotePad). Bovenaan staat hoe deze kaart heet: Tom (Kortst). Iets lager staat een lijst met locaties

(Nodes), elk met een naam en plaatscoördinaten. Nog lager staat een lijst met weggedeelten (Edges), elk met de twee eindpunten van het wegdeel en de lengte.

Kun je de edge 0-6 terugvinden op de kaart?

8. **Wegen toevoegen en veranderen.** Voeg een weg tussen de Toren en de School toe met lengte 12. (Om die te zien in Wybe moet je het bestand opslaan, en dan in Wybe een andere kaart kiezen en weer teruggaan naar Tom (Kortst) om die opnieuw te laden.)

Is de kortste route veranderd? Hoe kort zou je de weg van Toren naar School moeten maken om een korte route naar het Zwembad te krijgen? Probeer de weg zo kort te maken, dat de totale afstand 16 is.

9. **Route-voorkeuren.** In je routeplanner kun je meestal allerlei voorkeuren opgeven. Wil je wandelen, fietsen of autorijden? Wil je snel, kort, makkelijk doorrijden of weinig brandstof gebruiken? Wil je onverharde wegen, veerboten of flitspalen vermijden?

Vergelijk de kaarten Tom (Kortst), Tom (Snelst), Tom (Schoonst) en Tom (Wandel). Ze geven weer, hoe een routeplanner de kaart bekijkt bij verschillende voorkeuren. Als je de planner op Wandel zet, is de snelweg zelfs helemaal afwezig.

Een echte routeplanner gebruikt maar één kaart, waarin voor elk wegstukje allerlei informatie is opgenomen. Tijdens het zoeken wordt dan voor elk wegdeel op basis van de voorkeuren bepaald, of het gebruikt mag worden en wat de kosten dan zijn.

3 Hoe werkt Wybe?

De volgende opdrachten gaan iets verder in op de werking van Dijkstra's algoritme. Als je liever meer wilt weten over wat je er mee kunt doen, ga dan door naar opdracht 16.

Het zoeken naar routes gaat in ieder geval niet "logisch", op de manier waarop mensen naar een kaart kijken.

10. **Hoe wordt er gezocht?** Open de kaart Klungelig. Welke route moet je nemen van A naar AK? Als je op de startknop drukt, zie je dat Wybe al snel door heeft dat de afstand tot AK 10000 bedraagt. Toch wordt het hele rechterdeel van het netwerk bekeken voordat die route definitief wordt gekozen. Een computeralgoritme heeft geen "richtingsgevoel", en bekijkt toch wel alle punten rechts om zeker te zijn dat er niet een kortere route daarlangs is.
11. **Definitieve en voorlopige knopen: Schatting.** Tijdens het uitrekenen van de route zie je dat sommige knopen blauw worden: die noemen we *definitief* omdat van die knopen de afstand vanaf het startpunt volledig berekend is. Van de witte knopen is de afstand nog niet zeker; toch staat naast sommige daarvan ook een getal: de *schatting* van een voorlopige knoop. De schatting is: de lengte van een route via een blauwe buur.

12. **Verandering van schatting: Knoop N.** De schatting geeft soms wel een indicatie van de afstand, maar: de schatting van een knoop *kan nog veranderen* doordat burenen van knoop blauw worden.

Kies als kaart Rooster 1 en let op knoop N. Nadat knoop H blauw gemaakt is, krijgt N schatting 11, maar als later ook knoop M blauw wordt, verbetert de schatting van N tot 10.

Kun je ooit zeker weten dat je voor een knoop de uiteindelijke, echte afstand hebt gevonden?

13. **Dijkstra's vondst: de laagste schatting.** Edsger Wybe Dijkstra vond in 1959 het antwoord op dat vraagstuk. De *laagste* voorlopige schatting kan niet meer veranderen. En daarom is het steeds mogelijk, de laagste witte knoop definitief te maken (er verschijnt dan tijdelijk een rood wolkje achter).

En, als je nogmaals Rooster 1 kiest, en misschien de snelheidsschuif iets naar links zet, dan kun je ook zien dat dat is wat er gebeurt. Steeds wordt, uit de witte knopen, degene gekozen met laagste schatting; die knoop wordt definitief gemaakt (blauw), en van zijn burenen wordt de schatting aangepast.

14. **Het kiezen van de volgende knoop.** Voor de kaart Rooster 1 is dit proces goed te volgen. De kaart Rooster 2 is al minder overzichtelijk. Kies die kaart, en zet onderaan Wybe de wachtrij-keuze op *Mens-wachtrij*. Dit betekent, dat je nu zelf steeds de laagste knoop moet aanklikken.

Als je op de startknop drukt, verschijnt er naast A de schatting 0, maar verder gebeurt er niets. A is de enige knoop met schatting, en dus ook de laagste. Klik op A om A blauw te maken en schattingen te krijgen voor B en V. Welke is de laagste? Klik op die knoop. Nu zijn er twee knopen met laagste schatting (namelijk 3) en het maakt niet uit of je nu eerst M of eerst V aanklikt.

Gelukkig hoef je dit klikken bij een echte routeplanner niet te doen: bij het berekenen van een route kunnen miljoenen knopen bekeken worden! Als je een keer een verkeerde knoop aanklikt (een witte waarvan de schatting niet het kleinst is op dat moment) kan de berekende route een omweg zijn!

15. **De lijst- en boom-wachtrij.** Als je zo'n 10 à 20 knopen blauw hebt gemeekt, wordt het al behoorlijk onoverzichtelijk. In elke stap moet je het "randgebied" van de blauwe zone in de gaten houden. De knopen die een voorlopige schatting hebben, vormen samen het *front*. In een routeberekening op een kaart van Europa kan het front uit meerdere duizenden knopen bestaan.

De lijst-wachtrij bewaart wel al die knopen, maar let niet op de volgorde. In elke stap wordt de hele lijst met duizenden knopen bekeken om daar de kleinste uit te vissen. Gelukkig werkt een computer erg snel, toch voldoet dat idee van Dijkstra tegenwoordig niet meer. De boom-wachtrij bewaart ook het front, maar houdt ze met hun volgorde (van schatting) bij. Het werkt iets ingewikkelder, wat als een

schatting verandert moet je die volgorde soms veranderen. Maar het is ook sneller: in elke ronde hoef je maar enkele tientallen knopen te bekijken of te verplaatsen.

4 Modelleren met Wybe

Soms is je routeberekening iets ingewikkelder en lijkt het, of je het algoritme anders wilt laten werken. Gelukkig is het vaak mogelijk, de eisen van een toepassing te verwerken in de kaart, in plaats van het algoritme te veranderen.

16. **De Metro.** Open de kaart Metro (Naief). Je ziet het metronet van een grote Centraaleuropese stad, met op elk traject de rijtijd in minuten. Je wilt van Zicin naar Haje. Druk op Start (kies eers de lijst- of boom-wachtrij en schuif de snelheidsschuif voldoende naar rechts) en zie dat je gaat overstappen in Mustek en Muzeum.

Stop het algoritme (Stop&Reset), verplaats het startpunt naar Cerny Most (klik op groene vlag, dan op F) en het eindpunt naar Skalka. De route met overstap in Florenc en Muzeum is niet goed! Weliswaar zit je wat korter in de metro, maar omdat overstappen ook tijd kost, is het sneller om door te rijden tot Mustek en alleen daar over te stappen naar Skalka.

Hoe kun je de routeplanner zo aanpassen, dat er met overstaptijd rekening wordt gehouden? We willen natuurlijk, dat je altijd de snelste route krijgt, en soms betekent dat *wel* tweemaal overstappen.



17. **De Metro: Gesplitste stations.** Open de kaart Metro (Gesplitst). Je ziet, dat daarop de overstapstations tweemaal zijn weergegeven. Door Florenc rijden de rode en de gele lijn, daarom heeft Metro (Gesplitst) een knoop FlorencRo en een FlorencGe.

Als je het bestand `metrosplits.graph.txt` in de teksteditor opent en vergelijkt met `metronaief.graph.txt`, kun je de verschillen zien. De naieve kaart heeft één Florenc (knoop nr. 3), maar de gesplitste kaart heeft als knoop 3 FlorencGe en een

extra knoop 17, FlorencGe. Bij de Edges kun je zien, dat er een verbinding 3-17 is met kosten 4 minuten.

18. **VIA-punt instellen?** Open de kaart Toms Quest. Tom wil zo snel mogelijk zwemmen, maar heeft eerst geld nodig voor de entree. Bepaal de snelste route; die gaat helaas niet langs de PIN-machine.

Hoe kun je de snelste route vinden die wel langs de PIN machine gaat?

19. **De combinatorische route.** Tom heeft niet alleen geen geld, hij is ook zijn zwembroek kwijt! Geld kan hij halen bij de PIN-machine (kost 1 minuut), of lenen bij zijn vriend (dit kost 13 minuten) of bij Oma (die hem ook een zwembroek kan geven, maar hem 33 minuten lang thee en koekjes gaat geven). Hij kan zijn eigen zwembroek thuis zoeken (22 min.), er een bij Oma lenen (waar hij dan, na 33 min., ook geld krijgt), of in 2 minuten een zwembroek bij Zeeman kopen, maar dan moet hij eerst geld hebben!

Omdat er verschillende manieren zijn om aan een zwembroek en geld te komen, wordt het aantal te berekenen routes wel erg groot!

Hoeveel verschillende mogelijkheden zijn er voor Tom om aan geld en een zwembroek te komen? Als je bedenkt dat er bij al die routes ook tussenpunten zijn, hoeveel routeberekeningen zou je nodig hebben om de beste strategie te bepalen?

20. **Toms Quest: Locaties en Rugzakje.** Gelukkig is het mogelijk, een speciale “kaart” te maken van alle locaties en de spullen die Tom nodig heeft, waarmee je Toms probleem met één enkele routeberekening kunt oplossen. Open de kaart Toms Quest (Rugzak). Het idee achter deze kaart is, dat Tom niet alleen op elk van de 10 locaties kan zijn (Huis, Oma, Toren, etc). maar op elk van die tien locaties ook vier verschillende dingen kan hebben: (0) niets, (1) alleen geld, (2) alleen een zwembroek, (3) geld en zwembroek. De kaart heeft daarom 40 locaties (maar sommige zijn over elkaar heen getekend).

Open `ztomcombezit.graph.txt` in de tekst editor. Je ziet de startlocatie 0 (Tom is thuis en heeft nog niets) en de eindlocatie 38 (Tom is in het Zwembad met geld en zwembroek).

21. **Verbindingen in Toms Quest.** Niet alleen zijn er viermaal zoveel knopen, er zijn ook viermaal zoveel verbindingen. De oorspronkelijke kaart had een verbinding tussen Pin-Machine en School met een reistijd van 3 minuten, te zien als een regel 3-4 6 in het bestand `ztomcom.graph.txt`.

De kaart `ztomcombezit.graph.txt` bevat diezelfde regel, maar die betekent hier alleen maar dat je *zonder geld en zwembroek* van PIN-machine naar School kunt. Als je wel geld hebt, kun je ook van PIN-machine naar School, het duurt ook 6 minuten en bij de School heb je nog steeds geld; dit is uitgedrukt in de verbinding 13-14 6. Verder zijn er verbindingen 23-24 6 en 33-34 6.

Als je Wybe start met de kaart Toms Quest (Rugzak), zie je dat er wel lijnen naar het Zwembad worden getekend, toch krijg je de mededeling (rechtsonderin): Doel is onbereikbaar. Waarom?

22. **Spullen verkrijgen.** Met de kaart Toms Quest (Rugzak) kan Tom wel bij het Zwembad komen, maar niet met de gewenste zwembroek en het geld, omdat op de kaart nog niet is aangegeven hoe Tom daar aan kan komen.

Tom kan in 22 minuten zijn zwembroek thuis zoeken: dwz., hij kan van “Thuis met niets” (knoop 00) naar “Thuis met zwembroek” (knoop 20) in 22 minuten. Voeg in `ztomcombezit.graph.txt` onder [Edges] een regel toe met 0-20 22. Bij de PIN-machine kan Tom geld halen in 1 minuut: voeg de regel 3-13 1 toe.

Is zwembroek zoeken en pinnen genoeg om het doel te kunnen bereiken? Save het bestand `ztomcombezit.graph.txt`, wissel Wybe van kaart en ga terug naar Toms Quest (Rugzak) om de aangepaste kaart te laden. Druk op start.

23. **Verkrijg-kanten dupliceren.** Tom kan thuis zijn zwembroek niet alleen vinden als hij blut is, maar ook als hij geld heeft; daarom is nog een edge nodig: 10-30 22. Tom kan ook pinnen als hij zijn zwembroek al heeft; voeg de betreffende kant toe.

Start Wybe met de weer aangepaste kaart; Tom kan na 51,5 minuten zwemmen.

24. **Verdere verkrijg-mogelijkheden.** We gaan nu de verdere mogelijkheden van Tom modelleren om aan geld en een broek te komen. Uiteindelijk kunnen we dan laten uitrekenen, hoe hij het snelst kan zwemmen. In de nummering van knopen zit enige logica: knopen 0 t/m 9 betekenen locaties zonder geld of zwembroek, 10 t/m 19 herhalen die locaties met geld, 20 t/m 29 herhalen die met een zwembroek, en knopen 30 t/m 39 hebben geld en zwembroek.

Bij Oma kan Tom geld en een zwembroek krijgen in 33 minuten. Voeg de regel 6-36 33 toe.

Bij zijn vriend kan Tom geld lenen in 13 minuten (let op: *met* of *zonder* zwembroek!).

Bij Zeeman kan hij een zwembroek kopen in 2 minuten (let op: alleen als hij geld heeft!).

Als je deze edges hebt toegevoegd, laad dan de kaart opnieuw in Wybe. Je kunt nu zien, dat Tom in 49,5 minuten kan zwemmen.

(Alle verbindingen die je in de laatste opdrachten hebt kunnen toevoegen, zijn ook aanwezig in het bestand `ztomcomruilen.graph.txt`.)

5 Wat kan niet met Wybe?

Je hebt hopelijk een idee gekregen van wat er allemaal mogelijk is met routeplan-programma's. Veel van die mogelijkheden bestaan al, maar sommige zitten misschien pas in de routeplanners die er over een aantal jaren zijn.

25. **Wat moet een routeplanner kunnen?** Misschien heb je zelf wel ideeën en plannen voor wat je wilt kunnen uitrekenen met een routeplanner. En wie weet, kun je ook nog bedenken, hoe de routeplanner het kan berekenen.
26. **Eénrichtingsverkeer.** Het algoritme van Dijkstra kan ook rekenen met kanten die maar in één richting te doorlopen zijn. Maar vanwege de simpelheid kent Wybe alleen maar tweerichtings-verbindingen.
27. **Langste Route.** Misschien wil je een keer niet zo weinig mogelijk, maar juist zo veel mogelijk van iets. Een toeristische, of natuur-route, is dat niet een route langs *zoveel mogelijk natuurgebieden*? Kun je per wegdeel opslaan hoeveel kastelen er staan, en dan met Wybe een route met maximaal aantal kastelen vragen?

Dit is over het algemeen niet mogelijk. Eigenlijk is het ook niet wat je wilt: je wilt best een stukje omrijden voor een paar kastelen extra. Maar een route van Utrecht naar Amersfoort, die langs alle duizenden kastelen in Noorwegen, Spanje en Transsylvanië voert, schiet zijn doel voorbij.

Het idee van “zoveel mogelijk kastelen” kun je wel in een routeplanner verwerken. Natuurlijk moet de kaart informatie over de kastelen bevatten. Dan wordt een route berekend op de gewone, dus “minimale” manier, maar wegdelen met een kasteel tellen met een zekere korting mee. Hierdoor worden wegdelen met kastelen aantrekkelijker, en zul je er meer in je route tegenkomen. Maar een echte garantie op het maximaal mogelijke aantal heb je niet.

28. **Handelsreiziger.** Bij het klassieke *Handelsreizigersprobleem* is sprake van een aantal steden (locaties) die bezocht moeten worden, waarna de handelsreiziger terugkeert naar zijn startpunt. De vraag is, in welke volgorde de steden bezocht moeten worden voor een minimale reistijd.

Je kunt deze vraag wel met een enkele Wybe-zoektocht beantwoorden. Je kunt er een variant van de Quest voor maken, waarbij er in elke stad een uniek voorwerp opgehaald moet worden.

Helaas, bij elke extra stad, verdubbelt het aantal knopen in het Wybe-netwerk. Een gedetailleerde kaart van Nederland bevat zo'n half miljoen knopen. Als er tien steden bezocht moeten worden, wordt het aantal knopen al duizend maal zo hoog: een half miljard. Met twintig te bezoeken steden zal het Wybe-netwerk uit 500 miljard knopen bestaan. Veel verder kun je zelfs met een snelle computer niet gaan.