WISKUNNEND WISKE

DE VOLHARDENDE VOETBALFAN

FINALE 2018 - OPDRACHT 3

Leo is een hevige fan van het Belgisch voetbal. Behalve een vurige fan van Blauw Zwart, is hij ook geïnteresseerd in de voetbaltempels van de eersteklassevoetbalclubs. Daarom wil hij, samen met zijn kameraad Lambik, een tocht ondernemen langs alle speelsteden van de clubs uit de Jupiler Pro League om de stadia te fotograferen. Leo denkt echter ook economisch. Graag maakt hij vanuit zijn woonplaats Brugge een tocht langs alle steden en legt daarbij een zo kort mogelijke afstand af. In deze opdracht moeten jullie Leo helpen door een route te vinden die alle 16 speelsteden precies eenmaal aandoet, vertrekkend in Brugge, en uiteindelijk aankomend in Brugge.

Dit probleem is in de wiskunde gekend als het Traveling Salesman Problem. Het is een zogenaamd NP hard probleem. Dat betekent dat er geen efficiënt algoritme bestaat dat je kan toepassen op een willekeurige configuratie van een willekeurig aantal steden en dat altijd de best mogelijke oplossing vindt. Er bestaan wel efficiënte algoritmes om een oplossing te vinden waarvan men kan aantonen dat die niet veel afwijkt van de best mogelijke oplossing, en zelfs kan afschatten hoe groot de afwijking is. Om Leo te helpen, zullen we het algoritme beschrijven dat jullie moeten uitvoeren.

Het algoritme om een oplossing te vinden bestaat uit vier stappen. We vatten de vier stappen samen, en leggen dan elk van de stappen uitgebreid uit.

(1) Bepaal een **opspannende boom van minimaal gewicht**.
(2) Bepaal de steden in deze opspannende boom met *oneven graad.*
(3) Bepaal een **perfecte koppeling van minimaal gewicht**.
(4) Bepaal een optimale route.

Elke gelijkenis met bestaande gebeurtenissen en/of personen berust op louter toeval.

Opdracht (40 minuten)

1. Bepaal een opspannende boom van minimaal gewicht

Een *graaf* is een structuur die bestaat uit *bogen*, dit zijn verbindingen tussen de *toppen*, punten in een vlak, zoals de 16 speelsteden. In deze concrete context is een boog steeds een rechte lijn tussen twee steden. Alle mogelijke bogen tussen de 16 speelsteden vormen dus de bogen van een graaf. Elk van deze bogen heeft een gewicht, namelijk de afstand tussen de twee steden in vogelvlucht. Een *opspannende boom* is een verzameling van bogen met als eigenschap dat je vanuit elk punt, dus vanuit elke stad, op precies één wijze, via de bogen, naar elke andere stad kan. Het volgende algoritme bepaalt een opspannende boom van minimaal gewicht, dat is een verzameling bogen met als eigenschap dat als je de bijhorende afstanden optelt, je geen groter getal uitkomt dan de afstanden horende bij de bogen van elke andere mogelijke opspannende boom. Omdat Leo in Brugge woont, gaan we een opspannende boom bepalen met als beginpunt Brugge.

Noem T de verzameling van steden die je al kan bereiken, initieel is dit dus Brugge. Noem B de verzameling bogen die je reeds gekozen hebt, initieel is dit dus de ledige verzameling. Voer nu de volgende stappen uit tot dat T alle 16 steden bevat.

1. Schrijf voor elke stad $x \in T$ de dichtstbijzijnde stad $y \notin T$ op en de bijhorende afstand tot x.

2. Kies de stad y waarvoor de genoteerde afstand de kleinste is.

3. Voeg deze stad y toe aan T en voeg de boog van x naar y toe aan B. Begin nu opnieuw bij stap 1 tot je alle steden in T hebt.

Gebruik de bijgevoegde afstandstabel en teken de verbindingen op de kaart van België.

2. Bepaal de steden van oneven graad

Beschouw nu de opspannende boom uit 1. Noem de *graad* van een stad het aantal verbindingen met andere steden. Je zal bijvoorbeeld zien dat Oostende graad 1 heeft (enkel met Brugge verbonden), Brugge graad 2 (verbonden met Gent en Oostende) en Gent graad 3. Als je het algoritme uit 1 correct uitvoerde, zijn er precies 8 steden met **oneven** graad. (Dus Gent en Oostende zijn twee van die acht steden).

3. Bepaal een perfecte koppeling van minimaal gewicht

Je moet nu een *perfecte koppeling* bepalen op de 8 steden van oneven graad: dit is een verzameling van vier bogen zodanig dat elke stad op precies één van die bogen ligt. Er zijn uiteraard heel veel mogelijkheden om dit te doen, maar **je moet een perfecte koppeling bepalen waarvoor de som van alle bijhorende afstanden minimaal is**.
Tip: het is tijdrovend om alle 105 mogelijkheden op te schrijven en na te rekenen. Je kan beter op de kaart van België kijken om te zien welke verbindingen de kortste zijn. Als extra tip geven we mee dat de verbinding Oostende – Moeskroen tot de perfecte koppeling moet behoren.
Teken de 4 extra verbindingen op de kaart van België.

4. Bepaal een optimale route

Als je 1, 2, en 3 correct uitvoerde, dan heb je precies 19 bogen getekend. Deze 19 bogen doen alle steden aan, sommige steden zelfs meerdere malen. Dit geeft precies de mogelijkheid om een optimale route te bepalen, je hoeft elke stad maar één keer aan te doen. Bepaal nu een optimale route als volgt. Vertrekkend in Brugge, kies je b.v. als eerstvolgende stad Gent. Daar heb je (als je 1,2 en 3 correct uitvoerde,) drie mogelijkheden. Om rekenwerk te besparen beschik je over de kaart van België, met vertrekkend vanuit Brugge een stukje touw met als lengte de lengte van de optimale toer. Met het touw volg je de bogen die je kiest, waarbij je een stad die je al eens

aandeed, mag overslaan. Je zal zien dat je daardoor verbindingen gebruikt die je nog niet tekende, dat is de bedoeling.
Het finaal antwoord bestaat uit de lijst van steden in de volgorde zoals je ze tegenkomt op de door jullie gevonden route.

Wist je dat ...

Er zijn $15! = 1307674368000$ mogelijke routes die Leo en Lambik zouden kunnen afleggen. De precieze betekenis van *het feit dat er geen efficiënt algoritme* bestaat om de route met minimaal afgelegde afstand te vinden, is dat je alle mogelijkheden moet uittesten om zeker te zijn dat een route de best mogelijke is.
Het algoritme dat jullie uitvoerden om een route te bepalen, is een algoritme met polynomiale complexiteit. Dat betekent dat het aantal uit te voeren stappen begrensd is door een macht van n, met n het aantal steden. Voor het algoritme dat jullie hier gebruikten is dat n^3. Als het aantal steden dus stijgt, dan stijgt het aantal uit te voeren stappen, of berekeningen, volgens een derdegraadsfunctie. Dit is uiteraard veel trager dan de functie $n!$.
Er zijn nog algoritmen met vergelijkbare complexiteit gekend. De uitkomst van dergelijke algoritmen is een route waarvan men niet zeker is dat ze de route met kleinst mogelijk afgelegde afstand is, maar men kan wel de afwijking inschatten ten opzichte van de best mogelijke route.

Afstandstabel tussen de 16 speelsteden

	Anderlecht	Antwerpen	Beveren	Brugge	Charleroi	Eupen	Genk	Gent	Kortrijk	Lokeren	Luik	Mechelen	Moeskroen	Oostende	Sint-Truiden	Waregem
Anderlecht	0	40	38	78	43	112	78	43	64	44	84	24	67	93	57	55
Antwerpen	40	0	11	76	81	117	73	47	83	28	94	19	88	93	64	71
Beveren	38	11	0	65	81	127	83	36	73	19	102	23	79	81	72	61
Brugge	78	76	65	0	112	188	146	37	40	51	161	83	47	17	130	38
Charleroi	43	81	81	112	0	104	87	79	84	74	76	61	84	124	63	81
Eupen	112	117	127	188	104	0	47	151	175	137	28	106	179	204	56	167
Genk	78	73	83	146	87	47	0	111	140	96	33	65	145	162	24	130
Gent	43	47	36	37	79	151	111	0	37	18	125	48	48	51	96	26
Kortrijk	64	83	73	40	84	175	140	37	0	54	147	79	8	44	120	12
Lokeren	44	28	19	51	74	137	96	18	54	0	111	31	60	67	80	42
Luik	84	94	102	161	76	28	33	125	147	111	0	80	150	176	30	138
Mechelen	24	19	23	83	61	106	65	48	79	31	80	0	84	98	49	69
Moeskroen	67	88	79	47	84	179	145	48	8	60	150	84	0	49	124	18
Oostende	93	93	81	17	124	204	162	51	44	67	176	98	49	0	146	43
Sint-Truiden	57	64	72	130	63	56	24	96	120	80	30	49	124	146	0	111
Waregem	55	71	61	38	81	167	130	26	12	42	138	69	18	43	111	0