

Gulzige handelsreiziger

Programmeeropdracht 2 Algoritmen en Complexiteit 2016

1 Probleembeschrijving

Een handelsreiziger wil graag bij alle verschillende steden in het land langsgaan om zijn producten te verkopen. Het is echter voor hem niet zonder kosten om deze producten te kunnen verkopen. De verschillende steden zijn namelijk alleen verbonden door tolwegen. Om een tolweg te kunnen gebruiken moet je eenmalig het desbetreffende bedrag betalen. Hierna kun je daarentegen wel gratis de tolweg nog meerdere keren gebruiken.

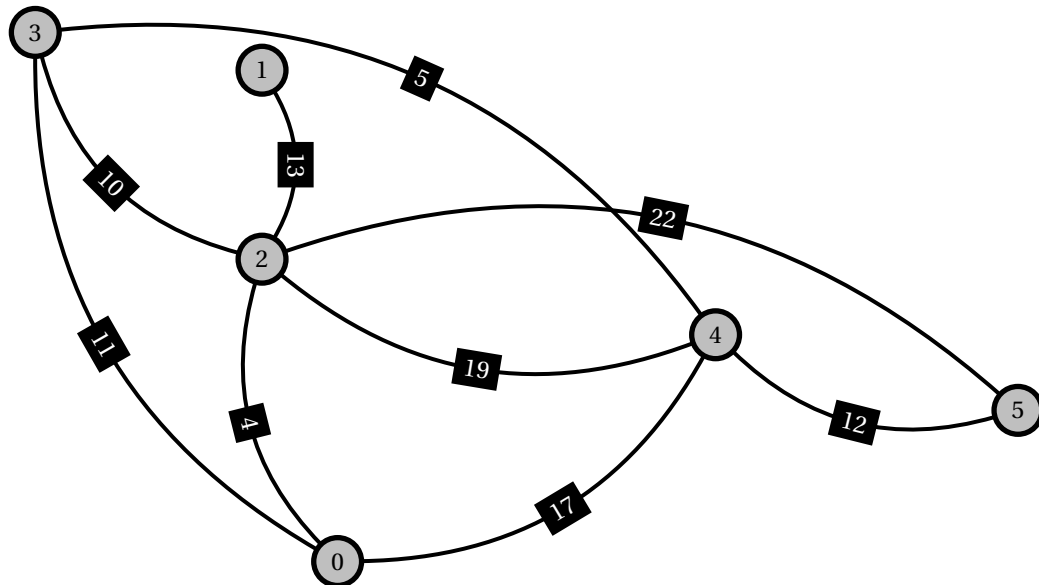
Gelukkig heeft de handelsreiziger een kaart met daarop alle kosten van alle mogelijke tolwegen. Het is jouw taak om een route te vinden die langs alle steden komt en waarvoor de handelsreiziger het minste aan tolgeld hoeft te betalen.

2 Invoer/uitvoer

Op de eerste regel staan twee getallen. Het eerste getal geeft aan hoeveel steden s er zijn op de kaart. Het tweede getal geeft aan hoeveel tolwegen t er zijn. Vervolgens wordt op de volgende t regels op elke regel een tolweg aangegeven met drie getallen. Het eerste getal op deze regel geeft aan vanuit welke stad de weg vertrekt, het tweede getal in welke stad de weg aankomt en het derde getal geeft aan wat de kosten zijn om deze weg te gebruiken (NB: wegen zijn bidirectioneel, dus de volgorde van de steden in een weg maakt niet en de kosten hoeven dus maar één keer betaald te worden in het geval je meerdere keren heen en weer gaat).

Als output worden de minimale totale kosten verwacht die nodig zijn om alle steden te bezoeken. De handelsreiziger begint altijd bij stad 0 en kan bij elke andere stad eindigen, zolang hij maar elke stad uiteindelijk minimaal één keer heeft bezocht. Mocht het niet mogelijk zijn alle steden te bezoeken, bijvoorbeeld omdat niet alle steden met elkaar verbonden zijn door wegen, print dan als output "Geen route mogelijk".

De voorbeeldkaart die ook in onderstaande voorbeeldinput staat weergegeven, is hieronder weergegeven. Een voorbeeldroute die minimale kosten heeft, is de route $0 \rightarrow 2 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ met kosten $4 + 13 + 0 + 10 + 5 + 12 = 44$.



2.1 Voorbeeldinvoer

```
6 9
0 2 4
0 3 11
0 4 17
3 4 5
1 2 13
2 3 10
2 4 19
2 5 22
4 5 12
```

2.2 Voorbeelduitvoer

```
44
```

3 Inleveren

Let op: Deze programmeeropdracht wordt voor een groot deel automatisch nagekeken. Zorg dus dat je je precies aan onderstaande richtlijnen houdt, anders kan je code niet worden nagekeken.

Controleer of je code goed werkt voor het inleveren

Door `check.sh` uit te voeren in dezelfde map als je python- en input/output- files kun je controleren of je code de juiste uitvoer geeft op de kleine input (`small.input`) en het juiste inputformaat heeft.

Nakijkeisen

- De code dient geschreven te zijn in Python 3.
- De code leest de input van *stdin* en schrijft naar *stdout*. Om eenvoudig invoer te testen kun je gebruikmaken van input redirection:
`python ex2.py < small.input`

Hint: de volgende voorbeeldcode leest op de juiste manier van stdin en schrijft hetzelfde naar stdout.

```
import fileinput
for line in fileinput.input():
    # line bevat al een newline van zichzelf
    print(line, end="")
```

- Het codebestand zelf moet de naam `ex2.py` hebben.
- Lever **alleen** het python bestand met je code in en geef het de naam `ex2.py`. Lever dus geen zip/tar in of iets dergelijks.

Hoe wordt de code nagekeken?

Er zijn twee inputfiles aan de hand waarvan je code wordt nagekeken:

- Bijgeleverd is `small.input` met de bijbehorende output `small.output`. Als je code dezelfde output geeft met `small.input` als input, wordt in principe het cijfer 6 toegekend.
NB: Om deze score te behalen is het wel noodzakelijk dat je daadwerkelijk re- delijkerwijs het algoritme hebt geïmplementeerd. Simpelweg altijd dezelfde output retourneren wordt niet goedgekeurd.
- Daarnaast is ook voor elke student apart (in `large.input.zip`) een inputfile `large_{studentnummer}.input` bijgeleverd (zonder de bijbehorende out- put). Als je code ook de juiste output voor deze input geeft, wordt in principe het cijfer 10 toegekend.

De code wordt hiernaast gecontroleerd op plagiaat van anderen en code op het internet. Hierop is de algemene plagiaat- en frauderegeling van de UvA van toepassing.

Geen inputfile ontvangen / overige vragen

Mocht je studentnummer niet tussen de inputfiles staan of mocht je overige vragen/problemen hebben over het inputformaat, stuur dan een mail naar R.B.Kluiwing@uva.nl.