# Notities bij Ontwerp

Bijgevoegd een document met code voor het Globescope Assesment

In het assesment wordt gevraagd om een programma te maken, waarmee minimaal 10 specifieke vragen beantwoord kunnen worden. Dit programma is er op gericht om deze 10 vragen te beantwoorden, maar er kunnen ook gelijkaardige vragen mee beantwoord worden met iets andere input.

Ik ben begonnen om een gemeenschappelijke deler tussen de 10 vragen vast te stellen. Binnen de vragen kunnen een aantal eenheden gedefinieerd worden:

* Node = plaats die geidentificeerd wordt met een letter van het alfabet
* Edge = verbinding tussen twee nodes
* Weight = afstand van de verbinding tussen twee nodes
* Route = georderde lijst van afwisselend nodes en edges, die de verbinding tussen een start node en een eind node vormen.

Dit vormde een uitgangspunt bij de opzet van de progamma code.

**Programma opzet**

De progamma code heb ik opgesteld in Python, omdat ik veel ervaring heb met deze taal.

De code bestaat uit 3 bestanden:

* main3.py: rechtstreeks aanroepbare functies
* railroads.py: definities van classes
* unittest\_gsa.py: (unit)test voor het programma.

In principe worden het programma gestart met functies uit het main3.py bestand. Twee functies die zijn bedoeld om rechtstreeks aan te roepen.

* findroute: voor het beantwoorden van vragen voor rechtstreekse routes tussen een begin- en een eindpunt.
* findroutes: voor het beantwoorden van routes, waarbij de route kan doorlopen na het bereiken van het eindpunt.

Door het aanroepen van deze twee functies met de juiste input kunnen de 10 vragen van het assesment beantwoord worden.

Naast deze twee functies zijn er een aantal hulpfuncties. De belangrijkste hulpfunctie is “routing”. Met deze functie worden de mogelijke rechtstreekse routes bepaald op basis van een beginpunt, eindpunt en de graph informatie. Een route wordt opgesteld als een list met een afwisseling van nodes en edges. De route is afgerond als de eind node wordt bereikt.

De hulpfuncties kunnen ook los van de functies findroute en findroutes gebruikt worden, maar het toepassen ervan is niet uitgewerkt. Naast deze functies zijn er ook classes.

**Classes**

Classes staan in het bestand railroads.py. In main3.py worden de classes geïmporteerd. Binnen de classes-objecten worden uiteraard methodes en attributen meegegeven.

Ik ben begonnen met classes voor een node- en een edge-objecten te maken, “Node” en “Edge” class object. De edges en nodes worden samengesteld uit de graph. informatie, graph informatie wordt omgezet naar node en edge-objecten via een aparte functie.

In een later stadium is een aparte klasse gemaakt voor een route, “Route”. Een route die de eind node bereikt had krijgt een True-waarde mee, routes die het eindpunt niet bereiken krijgen een waarde False mee voor het attribuut end.

**(Unit)test**

De unittest wordt gestart door het aanroepen van het bestand unittest\_gsa.py

Met de test worden de uitkomsten van de 10 vragen van de assesment getest.

**Bekende beperkingen/bugs**

* In functie “findroute”: alleen shortest kan opgegeven worden voor args.
* Voor correcte werking van functie findroutes moeten Nodes uitwaartse verbinding met een node hebben.
* Antwoorden op andere vragen dan de 10 uit het assesment zijn niet geverifieerd.
* Find\_node method bij Edge class: niet uitgewerkt voor inout =’out’
* Find\_node method bij Edge class: mogelijkheid bestaat dat noderes niet gedefinieerd wordt.