

Datastructuren 2018

Programmeeropdracht 3: Priority Queue & Dijkstra

Implementeer een binaire heap en gebruik deze om afstanden tussen knopen te bepalen in een graaf.

Deadline. Woensdag 2 januari 2019 23:59.

Binaire heap

Implementeer een binaire heap als priority queue. We gebruiken een array als representatie van de binaire heap. De specificatie van de priority queue ziet er als volgt uit:

- `bool isEmpty()` — bepaal of de heap leeg is
- `void insert(newItem, prio)` — voeg een element met een prioriteit toe
- `getMinimum()` — Geef het element met de laagste prioriteit
- `void deleteMinimum()` — Verwijder het element met de laagste prioriteit
- `void decrease(Item, prio)` — Verlaag de prioriteit van het gegeven element naar de gegeven waarde

Gebruik ook handige functies zoals `trickleUp` en `trickleDown`. Zie voor meer informatie ook het vak Algoritmiek.

Dijkstra

Implementeer het algoritme van Dijkstra voor het vinden van afstanden in een gerichte graaf met gewichten. Ga uit van een graaf in *adjacency list* representatie, en maak ook op een verstandige wijze gebruik van deze representatie. Gebruik de door U ontwikkelde binaire heap en in het bijzonder ook de operatie `decrease`.

Invoer. Het programma vraagt om een naam `xxx`, en leest dan de graaf uit het bestand `xxx.grf` in. Dit bestand bevat als eerste getal het aantal knopen $N \geq 1$ dat de graaf heeft. Dan op aparte regels de takken van de graaf beschreven door drie integers: beginknoop, eindknoop en gewicht. De knopen zijn getallen liggend tussen 1 en N . Het gewicht is positief.

Het bestand mag commentaar bevatten: dit zijn regels waarvan het eerste karakter ‘%’ is. Een voorbeeldbestand staat op de website: `fibonacci.grf`. Het programma vraagt om een beginknoop, en rekent alle afstanden uit vanuit de gegeven beginknoop naar de overige knopen in de graaf.

Uitvoer. Het programma creëert het bestand `xxx.dis` met daarin de afstanden: elke regel bevat telkens een knoopnummer, en de bijbehorende afstand, volgens oplopend knoopnummer.

Instructies. Werk in tweetallen. Raadpleeg de *Werkcollege Intro*. Documenteer je code en vermeld de auteurs in elke file. Stuur je bijdrage in een zip-file naar `j.k.vis@lumc.nl` met in het onderwerp [PRIORITY] gevolgd door de achternamen van de inzenders. Stuur een cc naar je partner zodat een antwoord aan iedereen gegeven kan worden.