

Datastructuren 2020

Programmeeropdracht 3: Priority Queue & Dijkstra

Implementeer een priority queue en gebruik deze om afstanden tussen knopen te bepalen in een graaf.

Deadline. Maandag 30 November 23:59

Binaire heap

Implementeer een priority queue als binaire heap, dus op basis van een sequentiele datastructuur zoals array of vector. De specificatie van de ADT 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 decreaseKey(Item, prio)` — Verlaag de prioriteit van het gegeven element naar de gegeven waarde

Gebruik intern de handige functies `bubbleUp` en `trickleDown`.

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 priority queue. Als Dijkstra tijdens het zoeken een kortere afstand vindt voor een knoop moet deze waarde ook in de heap aangepast worden, met de operatie `decreaseKey`. Dat betekent dat de knoop (via de sleutel `Item`) in de heap terug te vinden moet zijn. Bij een standaard heap kan dat niet: alleen het kleinste element is rechtstreeks te benaderen.

Ontwerp daarom een geschikte ‘index’ binnen uw heap, zodat op elegante en efficiënte wijze de sleutel gebonden is aan zijn positie in de heap.

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: `fibo.grf`. Het programma vraagt om een beginknoop, en rekt 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. Documenteer uw code. Lever behalve de code ook een kort verslag in, waarin in het bijzonder uitgelegd wordt hoe de ‘index’ tot de heap-elementen is geïmplementeerd.