

Tentamen Algoritmen en Datastructuren

donderdag 30 augustus 2007, 9 - 12 uur

Het tentamencijfer T is $(p/10) + 1$, waarbij p het totaal aantal behaalde punten is.

Met de zinsnede ‘geef een algoritme’ in een opgave wordt bedoeld:

**beschrijf een algoritme in pseudocode (dus niet in Java),
licht de werking ervan toe,
beargumenteer de correctheid.**

1. (30 punt) We gebruiken een hash-tabel $T[0..n-1]$ voor het opslaan van niet-negatieve gehele getallen (de keys). De tabelgrootte n is een priemgetal. De hash-methode is open addressing, waarbij *collisions* worden afgehandeld m.b.v. linear probing. De hashfunctie is gedefinieerd door $h(i) = i \bmod n$. Je mag aannemen dat de *load factor* α kleiner is dan 1. Twee verschillende locaties in de tabel mogen dezelfde key bevatten.

(a) Geef een algoritme voor het zoeken naar een key in de hash-tabel.

(b) Geef een algoritme voor het verwijderen van een key uit de hash-tabel. Geef ook aan waarom en hoe het zoekalgoritme moet worden aangepast indien verwijderen en zoeken beide mogelijk moeten zijn.

2. (30 punt) Karel Knapzak gaat met vakantie. Voor zijn bagage moet hij een keuze maken uit N artikelen genummerd 0 tot en met $N-1$. Artikel i heeft gewicht g_i en waarde w_i voor Karel: hierbij zijn g_i en w_i positieve gehele getallen. Karel kan ten hoogste gewicht G meenemen, en ook G is een positief geheel getal.

Geef een algoritme dat in $O(GN)$ tijd berekent voor welke waarde Karel maximaal aan bagage-artikelen kan meenemen. Geef ook aan hoe het algoritme uitgebreid kan worden indien ook de geselecteerde artikelen moeten worden opgeleverd.

(Aanwijzing: gebruik dynamisch programmeren.)

3. (30 punt) Het algoritme van Dijkstra vindt, in een gewogen ongerichte graaf G met daarin knoop u , voor elke knoop v in G de lengte van een kortste pad van u naar v . De gewichten zijn niet negatief. Neem aan dat G samenhangend en enkelvoudig is (geen self-loops, geen parallelle kanten), en verder dat G n knopen en m kanten bevat.

Het globale idee van Dijkstra's algoritme is om het probleem stapsgewijs op te lossen, en daarbij in elke knoop bij te houden wat de lengte van het tot dan toe gevonden kortste pad is. Bij elke stap wordt de verzameling van ‘goede’ knopen uitgebreid met de knoop buiten de verzameling met de laagste padwaarde; vervolgens wordt de padwaarde van een aantal knopen aangepast (*edge relaxation*).

Geef het algoritme van Dijkstra, en analyseer de tijdscomplexiteit. Je mag gebruik maken van een efficiënte priority queue. Geef ook aan welke datastructuur je kiest voor het representeren van de graaf, en waarom.