

tentamen datastructuren en algoritmen 27 oktober 2008

Opgave 1.

- (a) Implementeer een stack (met operaties push en pop) met twee queues.
(5 punten)
- (b) We implementeren een priority queue met een ongeordend rijtje. Wat is dan de complexiteit van priority-queue-sort, in termen van grote- O ? Beargumenteer je antwoord.
(5 punten)
- (c) Maak een hash-tabel ter grootte 9. Gebruik de hash-functie $h(k) = k \bmod 5$. Voeg aan de (initieel lege) hash-tabel de volgende getallen toe (in deze volgorde):

15, 28, 29, 25, 0, 33, 12, 17

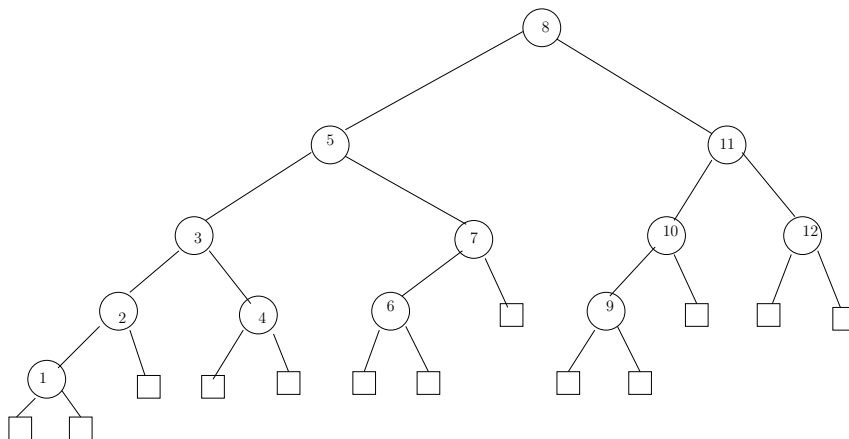
en wel op twee manieren:

- door collision met separate chaining op te lossen,
- door collision met linear probing op te lossen.

(6 punten)

Opgave 2.

- (a) Verwijder uit de volgende AVL-boom de knoop met label 4:



(5 punten)

- (b) Bewering: een preorder traversal van een heap geeft de elementen in niet-dalende volgorde.

Toon met een voorbeeld aan dat deze bewering onwaar is.

(5 punten)

- (c) Gegeven is een array ter lengte n met daarin de gesorteerde lijst van alle getallen van 0 tot en met n op één na. Een voorbeeld van zo'n array met $n = 5$ is $[0, 1, 3, 4, 5]$ waarbij 2 het ontbrekende getal is.

Geef een algoritme in $O(\log n)$ dat als input zo'n array neemt en als output het ontbrekende getal teruggeeft.

(6 punten)

Opgave 3.

- (a) Geef de merge-sort-boom voor het sorteren van de rij 8, 5, 2, 7, 1, 3, 4, 6, 9.

(5 punten)

- (b) De recurrente betrekking die bij merge-sort hoort is:

$$T(n) = \begin{cases} 1 & \text{als } n = 1 \\ 2T(\frac{n}{2}) + n & \text{als } n > 1 \end{cases}$$

Los deze recurrente betrekking op, en geef aan wat dus de tijdscomplexiteit van merge-sort is in termen van grote- O .

(6 punten)

Opgave 4.

- (a) Geef een voorbeeld waaruit blijkt dat steeds een item met een grootste waarde kiezen niet altijd leidt tot een optimale oplossing van het fractional knapsack probleem.

(5 punten)

- (b) Gegeven zijn 5 items met benefit-gewicht waarden als volgt:

$$(50, 1) \quad (12, 4) \quad (30, 5) \quad (1, 1) \quad (10, 2)$$

en gegeven is een maximum totaalgewicht $W = 10$.

Pas het algoritme voor fractional knapsack toe; geef je volledige berekening.

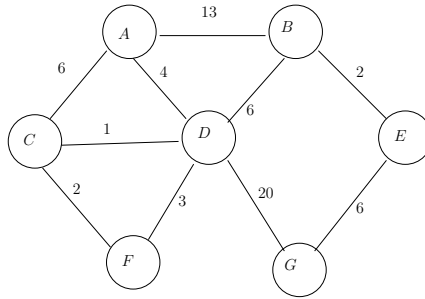
(5 punten)

- (c) Leg uit waarom het algoritme voor fractional knapsack in $O(n \log n)$ is, met n het aantal elementen in de verzameling S waaruit we (delen van) items kiezen.

(5 punten)

Opgave 5.

- (a) Pas Dijkstra's kortste pad algoritme toe op de volgende graaf, met A als startknoop. Geef stap voor stap aan wat er gebeurt, door in elke stap de graaf met de relevante data te tekenen.



(9 punten)

- (b) Geef (beargumenteerd) de complexiteit van Dijkstra's kortste pad algoritme in termen van grote- O .

(6 punten)

Opgave 6. Gegeven zijn de tekst T en het patroon P als volgt:

$$\begin{aligned} T &= abababdcabababcabbadab \\ P &= ababcab \end{aligned}$$

- (a) Beschrijf stap voor stap het toepassen van het brute-force pattern matching algoritme; nummer de stappen zodat je kunt zien hoeveel stappen gedaan worden.

(6 punten)

- (b) Geef de failure functie van het Knuth–Morris–Pratt pattern matching algoritme voor het patroon P .

(5 punten)

- (c) Beschrijf stap voor stap het toepassen van het Knuth–Morris–Pratt pattern matching algoritme; nummer de stappen zodat je kunt zien hoeveel stappen gedaan worden.

(6 punten)

Het cijfer is (het totaal aantal punten plus 10) gedeeld door 10.