

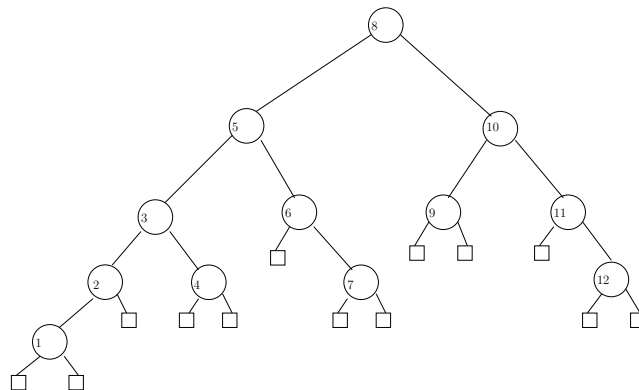
herkansing datastructuren 4 februari 2008

Opgave 1.

- (a) Geef in een plaatje een (echte) binaire boom met hoogte 4 en een minimum aantal externe knopen.
(3 punten)
- (b) Beschrijf hoe je de operaties **enqueue** en **dequeue** van het ADT voor queues kunt implementeren met twee stacks S_1 en S_2 .
(5 punten)
- (c) Geef een niet-recursief algoritme voor inorder traversal.
(8 punten)

Opgave 2.

- (a) Geef in een plaatje een voorbeeld van een binaire zoekboom die geen AVL-boom is. Leg uit waarom je voorbeeld geen AVL-boom is.
(3 punten)
- (b) Gegeven is de volgende AVL-boom:



Geef met plaatjes stap voor stap aan hoe de knoop met label 9 verwijderd wordt. Geef in je plaatjes steeds aan welke knoop niet in balans is, en welke knopen een rol spelen in de rotatie.

(8 punten)

Opgave 3.

- (a) Gegeven is het volgende algoritme:

```
Algorithm Loop(n):  
  s ← 0  
  for i ← 1 to 2n do  
    for j ← 1 to i do  
      s ← s+i
```

Geef een precieze karakterisering (Θ) van de tijdscomplexiteit van Loop. Motiveer je antwoord.

(5 punten)

- (b) Geef de merge-boom voor het uitvoeren van merge-sort op de input

1 9 8 5 2 7 4 3 6

(5 punten)

- (c) Beschrijf een methode om te bepalen of er in een gegeven rij van n elementen twee opeenvolgende elementen voorkomen (dus zowel i als $i + 1$ zitten in de rij, voor zekere $i \in \mathbb{N}$). Geef de tijdscomplexiteit van je methode met motivatie.

(6 punten)

Opgave 4.

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

(12, 4) (32, 8) (40, 2) (30, 6) (50, 1)

en gegeven is een totaal gewicht $W = 10$.

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

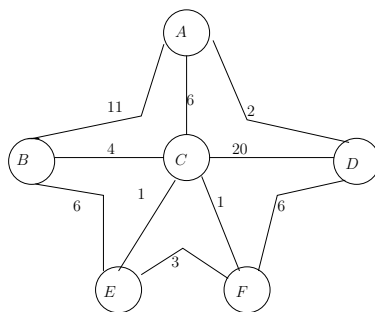
(8 punten)

- (b) Geef een voorbeeld waaruit blijkt dat het greedy fractional knapsack algoritme niet altijd een optimale oplossing levert voor het 0/1 knapsack probleem. Leg je voorbeeld duidelijk uit.

(7 punten)

Opgave 5.

- (a) Beschrijf stap voor stap (geef ook expliciet de D -waarden) het toepassen van Dijkstra's kortste pad algoritme met als input de volgende ongerichte graaf en startpunt A :



(9 punten)

- (b) Leg informeel maar in detail uit waarom het algoritme van Prim-Jarník (dat gegeven een graaf een minimale opspannende boom vindt) correct is.
(7 punten)

Opgave 6.

- (a) Leg uit waarom de tijdscomplexiteit van het brute-force pattern-matching algoritme in $O(nm)$ is, met n de lengte van de tekst en m de lengte van het patroon.
(5 punten)
- (b) Gegeven zijn het patroon P en de tekst T :

$$\begin{aligned} P &= abc \\ T &= acbabaabacabc \end{aligned}$$

Geef de failure functie van P .

(3 punten)

- (c) Pas het Knuth-Morris-Pratt pattern-matching algoritme toe op P en T zoals gegeven in 6(b). Geef stap voor stap aan wat er gebeurt.
(8 punten)

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