

Tentamen Data Structures and Algorithms

19 Oktober 2009

Dit tentamen bestaat uit 7 opdrachten en 1 extra (niet verplichte) opdracht.

Opdracht 1.

(a) Gegeven is het volgende algoritme:

```
Loop(n):  
    s = 0  
    for i = 1 to n do  
        for j = 1 to 3n do  
            s = s + 1  
        done  
    done
```

Wat is de tijdscomplexiteit van dit algoritme in termen van O ?

(5 points)

(b) Geef pseudo-code voor het implementeren van de operaties enqueue en dequeue van het ADT voor de queue, gebruikmakend van twee stacks S_1 and S_2 .

(5 points)

(c) Geef de worst-case tijdscomplexiteit van de methoden enqueue en dequeue uit je antwoord op vraag (a) in termen van O . (We gaan uit van stack operaties in $O(1)$.) Het antwoord volstaat.

(4 points)

(d) Vervang de vraagtekens door een zo simpel mogelijke expressie:

- (i) $n^2 + n^3 + n \in O(?)$
- (ii) $2 \cdot n + \log_2 n \in O(?)$
- (iii) $4^n + 7 \cdot n \cdot \log_2 n \in O(?)$

(6 points)

Opdracht 2.

- (a) We implementeren een priority queue met een **ongesorteerde** array. Wat is de tijdscomplexiteit van enqueue en dequeue in termen van O ? Motiveer je antwoord.
(5 points)
- (b) Beschouw weer een priority queue geïmplementeerd met een ongesorteerde array. Wat is de tijdscomplexiteit voor het sorteren van een (ongesorteerde) lijst gebruikmakend van deze priority queue? Motiveer je antwoord.
(5 points)
- (c) Geef de quick-sort boom voor het sorteren van de volgende lijst:

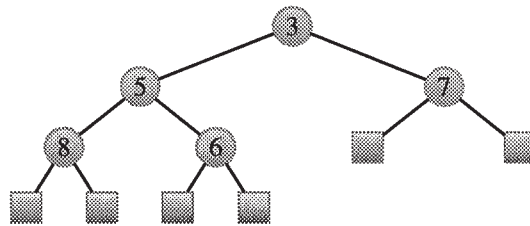
7, 5, 3, 8, 1, 9, 2, 6, 4

Neem als pivot steeds het meeste rechter element van de lijst.

(5 points)

Opdracht 3.

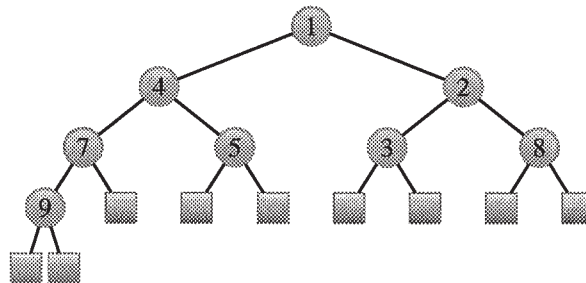
- (a) Voeg 1 toe aan de volgende heap (geef alleen het eindresultaat):



Voeg vervolgens 4 toe (geef weer alleen het eindresultaat).

(5 points)

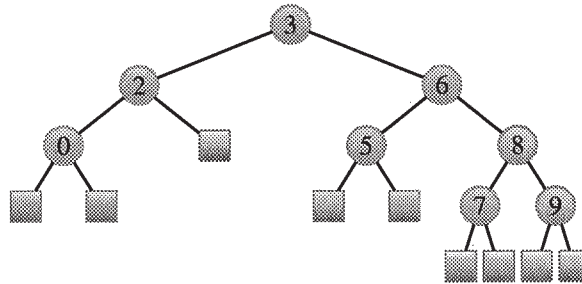
- (b) Verwijder de key 1 van de volgende heap (geef alleen het eindresultaat):



(5 points)

Opdracht 4.

Gegeven is de volgende AVL boom:



- (a) Geef stap voor stap aan hoe een knoop met key 10 wordt toegevoegd. (5 points)
- (b) Geef stap voor stap aan hoe de knoop met key 3 wordt verwijderd. (Werk met de hierboven gegeven boom, niet met je antwoord op vraag (a).) (5 points)

Opdracht 5.

- (a) Maak een hash tabel van grootte 9. Gebruik de hash functie $h(k) = k \bmod 9$. Pas linear probing toe op de initieel lege hash tabel om de volgende getallen (in deze volgorde) toe te voegen:

20, 15, 2, 11, 3

- (5 points)
- (b) Wat is de :
- (i) average (expected), en
 - (ii) worst-case
- tijdscomplexiteit (in termen van O) voor het zoeken van een key k in een hash tabel (gebruikmakend van chaining)? (5 points)
- (c) Wat is de :
- (i) average (expected), en
 - (ii) worst-case
- tijdscomplexiteit (in termen van O) voor het zoeken van een key k in een AVL boom? (5 points)

Opdracht 6.

Gegeven is de volgende text/string T:

a	b	a	c	a	b	a	b	a	c	a	b	c	a	b
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

samen met het volgende patroon P:

a	b	a	c	a	b	c
0	1	2	3	4	5	6

Pas de volgende pattern matching algoritmen toe om het patroon P in T te zoeken:

(a) Boyer-Moore (5 points)

(b) Knuth-Morris-Pratt (5 points)

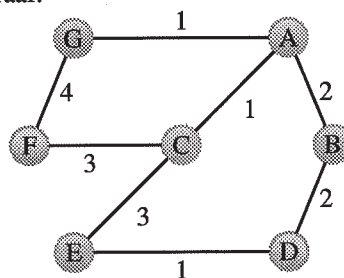
Geef per stap aan welke letter uit P vergeleken wordt met welke letter van T, en geef ook aan hoe P opgeschoven wordt. Een voorbeeld van deze notatie bij toepassen van het brute-force algoritme:

1. $P[0] = T[0]$
2. $P[1] = T[1]$
3. $P[2] = T[2]$
4. $P[3] = T[3]$
5. $P[4] = T[4]$
6. $P[5] = T[5]$
7. $P[6] \neq T[6]$
8. patroon wordt met 1 positie opgeschoven (nieuwe positie is 1)
9. $P[0] = T[1]$

...

Opdracht 7.

Gegeven is de volgende graaf:



- (a) Geef stap voor stap aan hoe de minimal spanning tree (MST) wordt berekend gebruikmakend van het Prim-Jarnik algoritme. Begin met knoop G en geef de volgorde aan waarin de knopen worden bekeken.

(5 points)

- (b) Wat is de tijdscomplexiteit van het Prim-Jarnik algoritme (in termen van O)? Motiveer kort je antwoord.

(5 points)

Opdracht 8. Niet verplicht; alleen voor extra punten

- (a) Geef een datastructuur waarmee de methoden toevoegen en zoeken van een key in een woordenboek (dictionary) tijdscomplexiteit $O(1)$ voor expected, en $O(\log n)$ voor worst-case hebben. Geef kort aan wat het idee is. Hint: denk aan een combinatie van twee bekende datastructuren.

(5 points)

- (b) Bewijs dat de worst-case hoogte van een AVL-boom $O(\log_2 n)$ is (met n het aantal knopen van de boom).

(5 points)

The grade is the (total amount of points plus 10) divided by 10.