

herkansing datastructuren en algoritmen 4 januari 2011

Dit tentamen bestaat uit 7 opgaven.

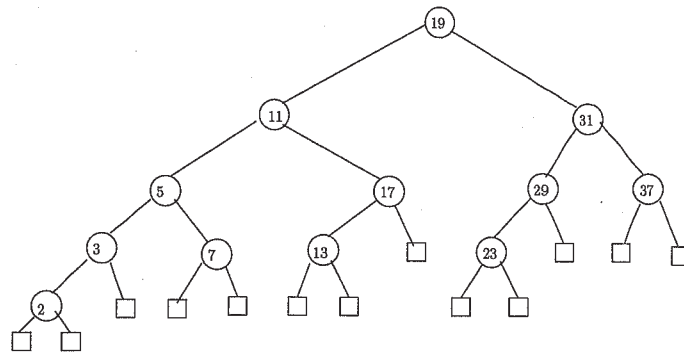
Het tentamencijfer is (het totaal aantal punten plus 10) gedeeld door 10.

Veel succes!

Opgave 1. (4+6 punten)

Deze opgave gaat over AVL-bomen.

- (a) Geef een klein voorbeeld van toevoeging aan een AVL-boom waardoor de boom uit balans raakt. Herbalanseer de boom.
- (b) Gegeven is de volgende AVL-boom:



Geef met plaatjes stap voor stap aan hoe de knoop met key 37 wordt verwijderd.

Opgave 2. (3+3 punten)

Deze opgave gaat over hashing.

Maak een hash-tabel ter grootte 7. Gebruik de hash-functie $h(k) = k \bmod 7$.

Voeg de volgende getallen (in deze volgorde) toe aan de initieel lege hash-tabel

17, 2, 23, 24, 5, 54

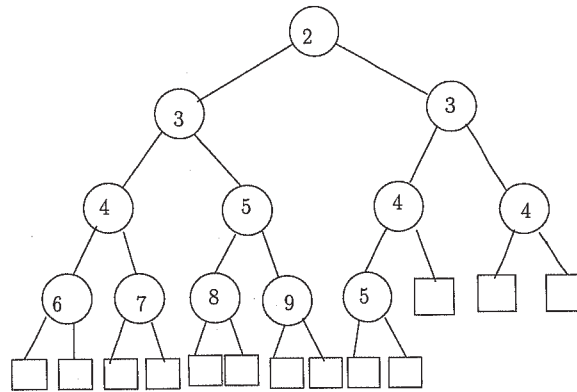
en wel op twee manieren:

- (a) door collision op te lossen met linear probing,
- (b) door collision op te lossen met chaining.

Opgave 3. (5+5+5 punten)

Deze opgave gaat over heaps en priority queues.

- (a) We implementeren een priority queue met een ongeordend rijtje. Wat is dan de complexiteit van priority-queue-sort, in termen van grote- $\mathcal{O}$? Beargumenteer je antwoord.
- (b) Geef het eindresultaat van het toevoegen van 1 aan de volgende heap:



- (c) Wat is de worst-case tijdscomplexiteit van de operatie 'toevoegen aan' een heap met n knopen, in termen van grote- $\mathcal{O}$? Ga uit van een vector-implementatie van een heap, en licht je antwoord kort toe.

Opgave 4. (4+5+5 punten)

Deze opgave gaat over binaire bomen en traversals.

- (a) Welke knoop wordt als allerlaatste bezocht in een preorder traversal?
- (b) Geef een algoritme (mag maar hoeft niet in pseudo-code) dat als input neemt een knoop v in een binaire boom T , en dat als output geeft de knoop die na v bezocht wordt in een preorder traversal.
- (c) Geef een voorbeeld waaruit blijkt dat de preorder-rij en de postorder-rij samen niet genoeg informatie leveren om uniek te bepalen om welke binaire boom het gaat.

Opgave 5. (5+5+5 punten)

Deze opgave gaat over sorteren.

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

- (c) Leg uit waarom de worst-case tijdscomplexiteit van bucket-sort $\mathcal{O}(n + N)$ is, met n het aantal elementen van de te sorteren rij, en $N - 1$ een bovengrens op de keys.

Opgave 6. (5+5+5 punten)

Deze opgave gaat over pattern matching en tekst-compressie.

- (a) Gegeven zijn de tekst T en het patroon P als volgt:

$$\begin{aligned} T &= \text{ACAAGATGCCATTGTCCCCAG} \\ P &= \text{CCCAG} \end{aligned}$$

Bepaal de *last*-functie, van het Boyer-Moore pattern-matching algoritme, voor het patroon P en het alfabet A, C, G, T.

- (b) Beschrijf stap voor stap het toepassen van het Boyer-Moore pattern-matching algoritme op patroon P en tekst T uit 6 (a); nummer de stappen.
- (c) Maak een frequentie-tabel en vervolgens een Huffman-coderingsboom voor de string parterretrappen.

Opgave 7. (5+5+5 punten)

Een gemengde opgave.

- (a) Is een algoritme in $\mathcal{O}(n)$ *altijd* sneller dan een algoritme in $\mathcal{O}(n^2)$?
- (b) Pas het algoritme voor knapsack01 toe op de volgende verzameling $S = \{s_1, s_2, s_3\}$, met items s_i met benefit b_i en gewicht w_i :

	b	w
s_1	4	2
s_2	10	6
s_3	8	4

met maximaal totaalgewicht $W = 8$.

Geef je antwoord in de vorm van een tabel:

$k \backslash w$	...
$\vdots$	

- (c) Gegeven is een verzameling taken, elk met begin- en eind-tijd. Geef een greedy algoritme (mag maar hoeft niet in pseudo-code) voor het kiezen van een zo groot mogelijke deelverzameling van taken die sequentieel op één machine uitgevoerd kunnen worden.