



*Dit voortentamen bestaat uit 8 opgaven.
Het cijfer is (het totaal aantal punten plus 10) gedeeld door 10.
Veel succes!*

Opgave 1. (5 punten)

Vervang de vraagtekens door een zo scherp en eenvoudig mogelijke expressie:

- (a) $7n + 5 \in \mathcal{O}(?)$
- (b) $67n + n \log(n) \in \mathcal{O}(?)$
- (c) $5n + 11n^2 + 7n \log(n) \in \mathcal{O}(?)$

Opgave 2. (5 punten)

Gegeven is de pseudo-code van insertion sort. Wat is de tijdscomplexiteit van insertion sort in termen van grote- \mathcal{O} ? Leg (kort) uit waarom.

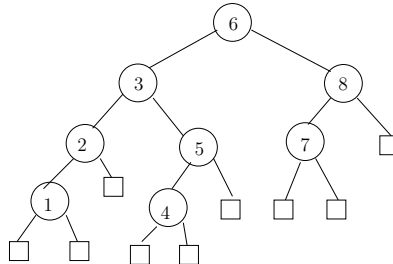
```
Algorithm insertionSort( $A, n$ ):  
  for  $j := 1$  to  $n - 1$  do  
     $key := A[j]$   
     $i := j - 1$   
    while  $i \geq 0$  and  $A[i] > key$  do  
       $A[i + 1] := A[i]$   
       $i := i - 1$   
     $A[i + 1] := key$ 
```

Opgave 3. (5+10 punten)

Deze opgave gaat over binaire zoekbomen en AVL-bomen; de externe knopen zijn leeg en de keys staan alleen op de interne knopen.

- (a) Geef een binaire zoekboom die geen AVL-boom is. (Antwoord volstaat.)

(b) Gegeven is de volgende AVL-boom:



Verwijder de knoop met key 8; geef alle stappen.

Opgave 4. (10 punten)

Deze opgave gaat over min-heaps; de externe knopen zijn leeg en de keys staan alleen op de interne knopen.

Bouw een min-heap van de getallen 1, 2, 3, 4, 5, 6, 7 (in deze volgorde) met de bottom-up heap constructie.

Geef in een plaatje alleen het eindresultaat, en vermeld het totaal aantal bubbel-stappen dat nodig was.

Opgave 5. (5+5 punten)

We bekijken een hash tabel ter lengte 7 en de hash functie h gedefinieerd door: $h(n) = n \bmod 7$. Voeg (in deze volgorde) de volgende getallen toe

1, 3, 2, 8, 10

aan een initieel lege hash tabel, waarbij collision wordt opgelost

- (a) met chaining (geef alleen het eindresultaat),
- (b) met linear probing (geef alleen het eindresultaat).

Opgave 6. (10+5 punten)

Deze opgave gaat over sorteren.

- (a) Geef de merge-sort boom voor het uitvoeren van merge-sort op de volgende input-rij: [4, 6, 2, 1, 3, 8].
- (b) Geef een best-case input-rijtje ter lengte 7 voor quick-sort waarbij als pivot steeds het laatste element wordt gekozen. (Antwoord volstaat.)

Opgave 7. (5+5+10 punten)

Deze opgave gaat over binaire bomen, met labels op de interne en externe knopen.

- (a) Geef een binaire boom waarvan de postorder traversal het volgende rijtje geeft: $[g, e, d, f, b, c, a]$.
- (b) Geef met een plaatje aan wat de knoop is die bezocht wordt na een knoop v bij een preorder traversal; onderscheid de gevallen v intern en v extern.
- (c) Geef pseudo-code voor een algoritme dat als input neemt een knoop v en een binaire boom T , met v een knoop in T , en dat als output geeft de knoop die na v wordt bezocht bij een preorder traversal ($\text{preorderNext}(T, v)$). Je mag de methoden uit het ADT voor binaire bomen gebruiken, zoals bijvoorbeeld isInternal , leftChild , parent , etc.

Opgave 8. (5+5 punten)

We bekijken een variant van quick-sort, waarbij we in plaats van één pivot twee pivots kiezen, om de input-rij in drie delen te splitsen. De input is een array A (waarvan we desgewenst de lengte kunnen bepalen). Neem voor de pivots de laatste twee elementen. We noemen deze variant 3-quick-sort.

- (a) Beschrijf (hoeft niet in pseudo-code, maar wel duidelijk) het algoritme 3-quick-sort.
- (b) Geef de 3-quick-sort-boom (analoog aan de merge-sort-boom en de gewone quick-sort-boom) voor het toepassen van 3-quick-sort op de input-rij $[14, 6, 1, 31, 7, 3, 27, 2, 21, 0, 22, 9, 4, 15]$.