



(nog checken)

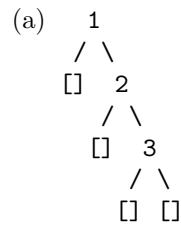
Opgave 1. (5 punten)

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

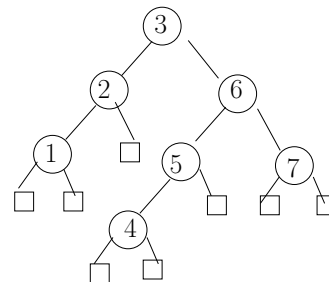
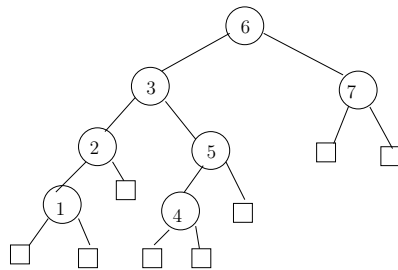
Opgave 2. (5 punten)

Insertion sort is in $\mathcal{O}(n^3)$. De tijdscomplexiteit wordt gegeven door $T(n) = \sum_{k=1}^{n-1} k$ dus (Thm 1.13) $T(n) = \frac{1}{2}(n-1)n$ dus de tijdscomplexiteit is in $\mathcal{O}(n^2)$.

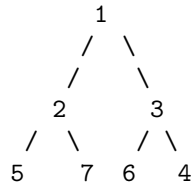
Opgave 3. (5+10 punten)



- (b) Eerst verwijderen we de knoop met key 8 en vervolgens moeten we één keer herbalanceren:



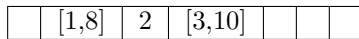
Opgave 4. (10 punten)



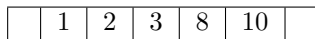
Er waren 4 bubbel stappen nodig.

Opgave 5. (5+5 punten)

(a) Met chaining:



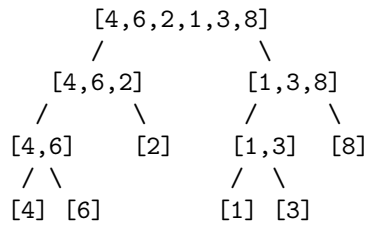
(b) Met linear probing:



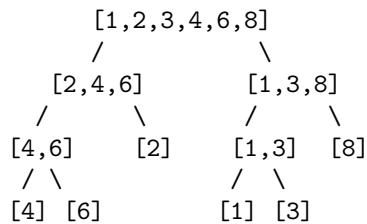
Opgave 6. (10+5 punten)

Deze opgave gaat over sorteren.

(a) De merge-sort boom:



En dan:

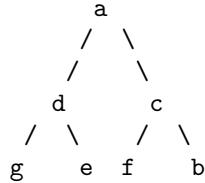


(b) Een best-case input-rijtje: [1, 3, 2, 5, 7, 6, 4].

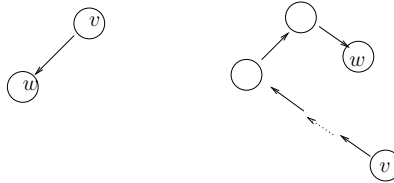
Opgave 7. (5+5+10 punten)

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

- (a) Een binaire boom waarvan de postorder traversal het rijtje $[g, e, d, f, b, c, a]$ geeft:



- (b) Plaatje met links v intern, rechts v extern:



- (c) Dit algoritme neemt als input een knoop v in een binaire boom en geeft als output de knoop die na v bezocht wordt in een preorder traversal:

Algorithm preorderNext(T, v):

```
if  $T.isInternal(v)$  then
  return  $T.leftChild(v)$ 
else
   $p := T.parent(v)$ 
  while  $T.leftChild(p) \neq v$  do
     $v := p$ 
     $p := T.parent(p)$ 
  return  $T.rightChild(p)$ 
```

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.

