

Tentamen Datastructuren

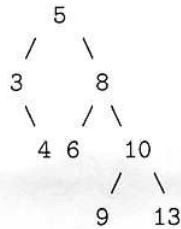
24 maart 2010, 14.00-16.00 uur

- Wat betekent "functie $f(n)$ is **niet** van orde $O(g(n))$ "?
 - Geef een big-Oh karakterisatie van het volgende algoritme:

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

- Algoritme A gebruikt $10n \log n$ operaties en algoritme B gebruikt n^2 operaties.
Bepaal de waarde van n_0 zó dat A beter is dan B voor $n \geq n_0$.
 - Geldt de volgende bewering? Leg uit waarom of waarom niet!
Er bestaat een binaire zoekboom met de getallen $1, \dots, 1000$ waarvoor geldt dat het *findElement*-algoritme op zoek naar het getal 363 de getallen 925, 202, 911, 240, 912, 245, 363 tegenkomt.
- Hoe zou een stack eruit zien na de volgende operaties:
push(14), push(42), pop(), top(), push(19), push(2),
pop(), pop(), push(5), top()
 - Hoe zou een queue eruit zien na de volgende operaties:
enqueue(4), enqueue(8), dequeue(), enqueue(98), enqueue(15),
dequeue(), dequeue(), enqueue(5), enqueue(2), front()
 - Geef de pseudocode voor een **push** op een stack waarbij gebruik wordt gemaakt van 2 queues met de bijbehorende queue-operaties. Je mag hierbij niet meer dan 3 regels gebruiken en je mag ook geen indices gebruiken.

3. (a) Voer op de volgende binaire zoekboom achtereenvolgens de volgende operaties uit en teken de boom na elke operatie:



(1) *insert(11)* (2) *delete(5)* (3) *insert(5)* (4) *insert(12)*

- (b) Geef aan of de binaire zoekboom gegeven in (a) een AVL-boom is. (Teken de hoogtebalans van de knopen.)
- (c) Voer nog een keer de *insert/delete* operaties van (a) uit, geef de hoogtebalans van de knopen en herstel zonnodig de hoogtebalans na elke stap. Teken de boom na elke operatie.
- (d) Wat is het voordeel van een AVL-boom t.o.v. een gewone binaire zoekboom?
4. (a) Geef de definitie van een heap.
- (b) Teken de heap die door de onderstaande array wordt gerepresenteerd. Afgebeeld zie je de sleutelwaarden van de opgeslagen entries.

1	2	5	8	25	19	20	23	18
---	---	---	---	----	----	----	----	----

- (c) Wat is de resulterende heap na het toevoegen (*insert*) van een entry met sleutelwaarde 6 in de heap gegeven bij (b)?
- (d) Vervolgens voeren we een *removeMin* uit. Wat is de nu de resulterende heap? Leg uit.
5. Stel we hebben een hash-tabel A van lengte 11. Met behulp van de hash-functie $h(i) = (i + 1) \bmod 11$ plaatsen we achtereenvolgens de sleutels

33, 10, 12, 22, 55, 20, 42

in een aanvankelijk lege tabel.

- (a) Hoe ziet A er uit als collisions afgehandeld worden met *separate chaining*?
- (b) Hoe ziet A er uit als collisions afgehandeld worden met *linear probing*?