Tentamen 'Datastructuren'

1 juni 2006, 9.30–12.30 uur

`Het tentamen bestaat uit 5 opgaven. Lees de opgaven zorgvuldig, schrijf duidelijk en beargumenteer je antwoorden. Schrijf op elk ingeleverd blad je naam en je collegekaartnummer. Je mag GEEN gebruik maken van een boek, aantekeningen of ander schriftelijk materiaal. Succes!`

1. Gelden de volgende beweringen? Zo ja, leg uit waarom; zo nee, geef een tegenvoorbeeld.

   (a) $9n(n+1)^7 + n^2$ is $O(n^7)$.

   (b) Als we veronderstellen dat de pivot altijd het laatste element is, dan is de worst-case running time van een quick-sort van een rij met n elementen $O(n^2)$.

   (c) De worst-case running time van een preorder traversal van een binaire zoekboom met $3n$ elementen is $O(\log n)$.

2. Beschouw het volgende algoritme in pseudo-code dat gebruik maakt van een interne, aanvankelijk lege stapel S.

```
Input: een array A van n positieve getallen
Output: ?
t ← 0
for i ← 0 to n−1 do
  if A[i] is oneven then
    S.push(A[i])
  else
    while ! S.isEmpty() do
      t ← t + S.pop()
while ! S.isEmpty() do
  t ← t + S.pop()
return t
```

(a) Wat is de output van dit algoritme op de input

$$A = \{1, 9, 11, 14, 5, 3, 7, 13, 3, 12, 5\}$$

?

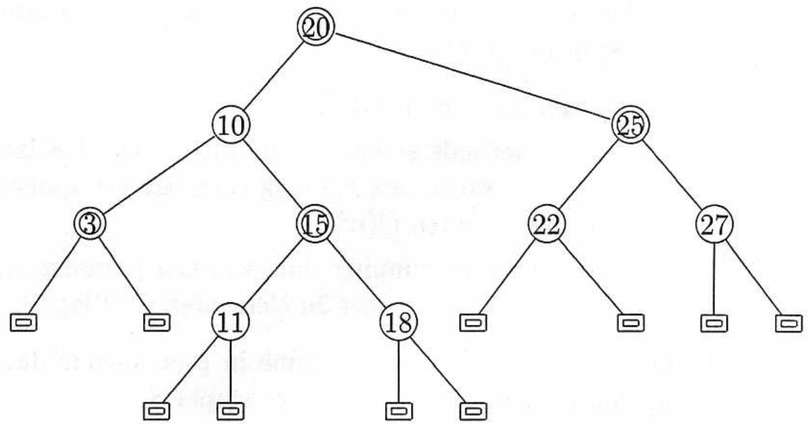
(b) Beschrijf in EEN zin wat dit algoritme doet.

(c) Bepaal de worst-case tijdscomplexiteit van het algoritme in termen van n. Leg informeel uit!

3.

(a) Geef de definitie van een rood/zwart-boom.

(b) Gegeven is de volgende rood/zwart-boom (zwarte knopen hebben een dubbele rand, de rand van de rode knopen is enkelvoudig).



Hoe ziet de boom er uit na de rood/zwart-operatie $insert(12)$?

(c) "In een rood/zwart-boom verschilt de diepte van 2 bladeren hoogstens 1". Geldt deze bewering? Zo ja, bewijs dat dan; zo nee, geef een tegenvoorbeeld.

4.

(a) Geef de definitie van een heap.

(b) Teken de heap die door het onderstaande array wordt gerepresenteerd.

	1	2	5	8	25	19	20	23	18

(c) Een heap heet *echt* als alle bladeren dezelfde diepte hebben. Laat zien dat voor een echte heap geldt dat

$$B = 2^d$$

waar B het aantal bladeren is en d hun diepte.

5. Stel we hebben een hash-tabel A van lengte 11. Met behulp van de hash-functie $h(i) = (2i + 5) \bmod 11$ plaatsen we achtereenvolgens de keys

$$23, 55, 24, 99, 34, 105, 22, 50, 31, 27, 16$$

in een aanvankelijk lege tabel.

(a) Hoe ziet A er uit als collision afgehandeld wordt met

i. *separate chaining*?

ii. *linear probing*?

iii. *quadratic probing*?

(b) De *quadratic probing* strategie heeft het nadeel dat het kan voorkomen dat geen lege positie gevonden wordt, terwijl die er wel is. Een betere strategie is om de volgende serie posities te bekijken op zoek naar een lege:

$$A[h(k)], A[(h(k) + 1) \bmod N], A[(h(k) - 1) \bmod N],$$

$$A[(h(k) + 4) \bmod N], A[(h(k) - 4) \bmod N],$$

$$A[(h(k) + 9) \bmod N], A[(h(k) - 9) \bmod N], \ldots$$

etc. oftewel $A[(h(k) \pm j^2) \bmod N]$ met $j = 0, 1, 2, 3, \ldots, N - 1$. Hoe ziet A er uit als collision op deze manier afgehandeld wordt?