Departement Informatica en Informatiekunde, Faculteit Bètawetenschappen, UU.
In elektronische vorm beschikbaar gemaakt door de TBC van A−Eskwadraat.
Het college INFODS werd in 2004/2005 gegeven door Frank van der Stappen.

Datastructuren (INFODS)
20 mei 2005

Opgave 1 *(1.5 punt)*

Bewijs of weerleg de volgende beweringen:

a) Als $f(n)$ is $O(n)$ en $g(n)$ is $O(n)$, dan $f(n) + g(n)$ is $O(n^2)$. *(0.5 punt)*

b) Als $f(n)$ is $O(n^2)$ en $g(n)$ is $O(n)$, dan $\frac{f(n)}{g(n)}$ is $O(n^2)$. *(0.5 punt)*

c) Als $f(n)$ is $O(d(n))$ en $g(n)$ is $O(e(n))$, dan $2f(n) + 3g(n)$ is $O(d(n) + e(n))$. *(0.5 punt)*

Opgave 2 *(2 punten)*

Orden de functies $f_1 \dots f_9$ van klein naar groot op basis van hun asymptotisch groeigedrag (of, in andere woorden, in grote-O notatie). Geef ook aan welke functies gelijk groeigedrag hebben (of, in andere woorden, grote-Θ van elkaar zijn):

$$f_1(n) = n \log n \quad f_2(n) = n^2\sqrt{n} + n^{\frac{7}{3}} \quad f_3(n) = n^2 \log n$$

$$f_4(n) = 550^{550} \quad f_5(n) = n^{\frac{7}{3}} \quad f_6(n) = n^2$$

$$f_7(n) = n^{549} \quad f_8(n) = (n \log n)^2 \quad f_9(n) = n^2 \log(n^2)$$

Opgave 3 *(1 punt)*

Bewijs: $n^3 \log n + n^2 + 3n$ is $\Theta(n^3 \log n)$.

Opgave 4 *(2 punten)*

Gegeven zijn een List A van m verschillende oneven positieve integers en een List B van n verschillende oneven positieve integers. De elementen van A zitten van klein naar groot in A en de elementen van B zitten van klein naar groot in B. Geef een zo efficiënt mogelijk algoritme *Tweevouden* dat tweevouden van de elementen in A aan B toevoegt, en tweevouden van elementen in B aan A toevoegt. De elementen van de resulterende List A moeten weer van klein naar groot in A zitten en de elementen van de resulterende List B moeten weer van klein naar groot in B zitten.
Tweevouden zal dus bijvoorbeeld de Lists $A = (7, 13, 15, 19, 21, 25)$ en $B = (1, 5, 33, 47, 51)$ veranderen in $A = (2, 7, 10, 13, 15, 19, 21, 25, 66, 94, 102)$ en $B = (1, 5, 14, 26, 30, 33, 38, 42, 47, 50, 51)$.
Maak voor je algoritme slechts gebruik van de standaardoperaties (ofwel methods):

- first(), last(), prev(p) en next(p), size(), isEmpty() voor toegang tot (de Positions p van) de Lists A en B,
- replace(p, e), insertFirst(e), insertLast(e), insertBefore(p, e), insertAfter(p, e) en remove(p) voor wijziging van (de Positions p en elementwaarden e) van de Lists A en B,
- element() voor toegang tot de (positieve integer) elementwaarde van een Position p.

Analyseer de looptijd van je algoritme als je weet dat alle standaardoperaties $O(1)$ tijd kosten.

Opgave 5 *(2 punten)*

Gegeven is een enkelvoudig gelinkte lijst (singly linked list) L met een onbekend maar even aantal knopen, en een sentinel *head* die wijst naar de eerste knoop van L. Geef een zo efficiënt mogelijk algoritme dat de volgorde van de knopen van de eerste helft van L omdraait. Analyseer de looptijd van je algoritme.

Opgave 6 *(1.5 punt)*

Geef een zo goed mogelijke worst-case tijdsanalyse voor het volgende algoritme:

ALGORITHM *DoeMaarWat(A)*:

input: een array A van n integers
output: een array F van n berekende integers

int $i, j, k, F[\,]$
$F[n-3] \leftarrow 0$
$F[n-2] \leftarrow 0$
$F[n-1] \leftarrow 0$
for $i \leftarrow 0$ **to** $n-4$ **do**
    $F[i] \leftarrow A[i]$
    $j \leftarrow i+1$
    while $j < n-4$ **do**
        $F[i] \leftarrow F[i] + A[i+j]$
        $j \leftarrow 3 * j$
for $i \leftarrow 0$ **to** $n-4$ **do**
    for $j \leftarrow 1$ **to** 3 **do**
        $F[i] \leftarrow F[i] + F[i+j]$
return F