Informatica Instituut, Faculteit Wiskunde en Informatica, UU.
In elektronische vorm beschikbaar gemaakt door de TEX van A−Eskwadraat.
Het college INFODS werd in 2003/2004 gegeven door Frank van der Stappen.

Datastructuren (INFODS)
14 mei 2004

Opgave 1 *(0.5 punt)*

Wat is de inhoud van de aanvankelijke lege stack na de operaties push(4), push(9), push(3), pop(), push(7), push(1), push(1), push(1), pop(), push(8), pop()?

Opgave 2 *(2 punten)*

Orden de functies $f_1 \ldots f_{12}$ 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):

$$\begin{array}{lll} f_1(n) = n \log n & f_2(n) = n\sqrt{n} + 27n & f_3(n) = \log(\sqrt{n}) \\ f_4(n) = 3^{500} & f_5(n) = 2^{\log(\log n)} & f_6(n) = n^2 \\ f_7(n) = 2^n & f_8(n) = (\log n)^2 & f_9(n) = \frac{n^2}{\log n} \\ f_{10}(n) = \log n^{2n} & f_{11}(n) = 2n^{\frac{3}{2}} - 3n & f_{12}(n) = 3^n \end{array}$$

Opgave 3 *(0.5 + 0.5 + 1 punt)*

Bewijs

a) $2n^3 + 9n^2 \log n$ is $O(n^3)$.

b) $\frac{(n \log n)}{8}$ is $\Omega(n \log n)$.

c) $2^{n+2} - n$ is $\Theta(2^n)$.

Opgave 4 *(2 punten)*

Gegeven is een List A van n positieve integers. De elementen van A zitten van klein naar groot in A. Geef een zo efficiënt mogelijk algoritme *Ontrafel* dat de volgorde van A zodanig verandert dat na afloop eerst de oneven elementen van klein naar groot en daarna de even elementen van klein naar groot staan. *Ontrafel* zal dus bijvoorbeeld de List $A = (2, 6, 7, 10, 13, 15, 16, 19, 21, 22, 25)$ veranderen in $A = (7, 13, 15, 19, 21, 25, 2, 6, 10, 16, 22)$.
Maak voor je algoritme slechts gebruik van de standaardoperaties (oftewel methods):

- first(), last(), prev(p) en next(p), size(), isEmpty() voor toegang tot (de Positions p) van de List A,
- replace(p, e), insertFirst(e), insertLast(e), insertBefore(p, e), insertAfter(p, e) en remove(p) voor wijziging van (de Positions p en de elementwaarden e) van de List A,
- element() voor de 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 zijn een Queue B met m positieve integers en een Queue C met n positieve integers. De elementen van B zitten van klein naar groot in B. De elementen van C zitten van klein naar groot in C. Geef een zo efficiënt mogelijk algoritme *Dertien* dat telt hoeveel paren bestaand uit een element van B en een element van C er zijn waartussen het verschil 13 is. *Dertien* zal dus bijvoorbeeld voor de Queues $B = (2, 19, 21, 24, 28)$ en $C = (4, 8, 15, 19)$ het getal 3 opleveren.
Maak voor je algoritme slechts gebruik van de standaardoperaties (oftewel methods):

- enqueue(e), dequeue(e), front(), size(), isEmpty(), waarbij e een nieuw (positief integer) element is

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

Opgave 6 *(1.5 punten)*

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

ALGORITHM *Raar(A)*

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

```
int i, j, k, F[ ]
for i ← 0 to n − 1 do
    F[i] ← 0
    j ← i
    while j < n do
        for k ← 0 to 4 do
            F[i] ← F[i] + A[j + k]
        j ← 2 ∗ j
return F
```