INFORMATICA INSTITUUT, FACULTEIT WISKUNDE EN INFORMATICA, UU.
IN ELEKTRONISCHE VORM BESCHIKBAAR GEMAAKT DOOR DE $\mathcal{TBC}$ VAN A–Eskwadraat.
HET COLLEGE INFODS WERD IN 2002/2003 GEGEVEN DOOR MARINUS VELDHORST.

Datastructuren (INFODS)
28 mei 2003

Opgave 1 *(3 punten)*

Gegeven twee queues A en B, beide van integers. Neem aan dat de elementen van A van klein naar groot in A zitten (het kleinste element vooraan, in de front-positie). Hetzelfde geldt voor B. Elke integer in A komt precies één keer voor in A. Elke integer in B komt precies één keer voor in B.

a) Geef een algoritme dat een nieuwe queue C genereert die als elementen precies die integers x bevat die voldoen aan een van de volgende twee voorwaarden:

- x zit wel in A, maar niet in B;
- x zit wel in B, maar niet in A.

Na afloop van je algoritme moet A hetzelfde zijn als bij de start van het algoritme. Hetzelfde geldt voor B. Na afloop van je algoritme moeten de integers in C van klein naar groot in C zitten, en mag C geen dubbele bevatten.

Geef je algoritme in pseudocode, en gebruik voor het access van A, B en C alleen de operaties behorend bij het abstract datatype **queue**. Verder is er nog de eis dat je algoritme werkt in $O(n+m)$ waarbij n en m het aantal elementen zijn in, respectievelijk, de oorspronkelijke A en B. Je mag ervan uit gaan dat de standaard queue-operaties in constante tijd plaatsvinden.

b) Beargumenteer dat je algoritme correct is en voldoet aan de gestelde tijdgrens, onder de aannames genoemd in onderdeel a).

Opgave 2 *(2.5 punten)*

Gegeven het volgende algoritme:

Algorithm weird(A):
Input: een array A van n^2 integers, A heeft grenzen 0 en $n^2 - 1$
Output: een array S van n^2 integers berekend volgens onderstaande methode.

```
for i ← 1 to n × n
do xx ← 0;
    integer sqrti ← ⌊√i⌋;          ⌊ ⌋ rondt naar beneden af
    for j ← 0 to n × sqrti − 1
    do xx ← xx + A[(3 × j)  mod (n × n)] + sqrti   enddo
    S[i − 1] ← ⌊xx/((i + 1) × (sqrti + 1))⌋
enddo
return S
```

Maak een zo goed mogelijke tijdsanalyse (worst-case) van algoritme *weird*, in termen van grote O van een functie in n.
Je mag ervan uit gaan dat $\lfloor\sqrt{i}\rfloor$ in tijd $O(\log i)$ berekend wordt uit i, dat **return** S in constante tijd plaatsvindt, en dat de operaties mod en $\lfloor\ \rfloor$ elk ook in constante tijd werken.

Opgave 3 ***(2.5 punten)***

Orden de volgende lijst van functies van klein naar groot in grote O notatie (oftewel, hoe staan ze geordend op de functiethermometer). Geef daarbij ook aan welke functies grote Θ van elkaar zijn. Ga ervan uit dat de log-functie grondtal 2 heeft. Verder zijn ter herinnering nog vermeld dat $\log\log n$ een verkorte schrijfwijze is van $\log(\log n)$.

$$\begin{array}{lllll}
n\log n & 2^{\frac{n}{\log n}} & 4^n & n^3 & (\log(\sqrt{n}))^2 \\
2^{(2^n)} & \lceil\sqrt{n}\rceil & n^{0.01} & 2^n & 4n^{\frac{3}{2}} \\
4^{\log n} & 2^{100} & \log\log n & (\log n)^2 & 2^{\log\log n} \\
\frac{n^2}{\log n} & \sqrt{\log n} & 5n & \lfloor 2n(\log n)^2\rfloor & \frac{n^2\log n}{\sqrt[3]{n^2}}
\end{array}$$

Er worden geen bewijzen gevraagd.

Opgave 4 ***(2 punten)***

Stel we hebben een enkelvoudig gelinkte lijst L zonder header en trailer sentinels (schildwachten). L bevat een oneven aantal knopen. Geef in pseudocode een algoritme dat, gegeven een pointer naar de eerste knoop van L, de middelste knoop van L vindt, waarbij je alleen met pointer hopping door de lijst mag gaan. Met name mag je geen teller gebruiken. Je algoritme moet werken in een worst-case die lineair is in het aantal knopen in L.
Beargumenteer dat je algoritme correct is en voldoet aan de tijdgrens.