INFORMATICA INSTITUUT, FACULTEIT WISKUNDE EN INFORMATICA, UU.
IN ELEKTRONISCHE VORM BESCHIKBAAR GEMAAKT DOOR DE $\mathcal{TBC}$ VAN A–Eskwadraat.
HET COLLEGE INFODS WERD IN 2003/2004 GEGEVEN DOOR FRANK VAN DER STAPPEN.

Datastructuren (aanvullend tentamen) (INFODS)
26 augustus 2004

Opgave 1 *(1 punt)*

Leg uit hoe *Bucket-sort* werkt op n gehele getallen uit het interval $[0,\ldots,N-1]$. Besteed aandacht aan de worst-case asymptotische looptijd. Je antwoord mag niet langer zijn dan 8 regels.

Opgave 2 *(1.5 punt)*

Orden de functies $f_1,\ldots,f_9$ van klein naar groot op basis van hun asymptotisch groeigedrag (of, in andere woorden, in grote-O notatie).

$$f_1(n) = n^2 \log n \qquad f_2(n) = 3n\sqrt{\tfrac{n}{8}} + 2n \qquad f_3(n) = \log n^{\pi}$$

$$f_4(n) = 44^{444} \qquad f_5(n) = n \qquad f_6(n) = n^2 + \sqrt[4]{n^3}$$

$$f_7(n) = 2^{\frac{n}{2}} \qquad f_5(n) = (\log n)^2 \qquad f_9(n) = \frac{n^2}{\sqrt[4]{n^3}}$$

Opgave 3 *(1 punt)*

Bewijs

a) $2^{n+2} + n^2$ is $O(2^n)$. *(0.5 punt)*

b) $(\log\sqrt{n})\cdot(\log \pi n)$ is $\Omega((\log n)^2)$. *(0.5 punt)*

Opgave 4 *(1.5 punt)*

We beschouwen een binaire zoekboom (binary search tree) T op n natuurlijke getallen. De ordening op de elementen van T is de standaardordening op de natuurlijke getallen. Geef een zo efficiënt mogelijk algoritme *Dubbelen* dat bepaalt of T dubbele elementen bevat. Analyseer de looptijd van je algoritme.

Opgave 5 *(2 punten)*

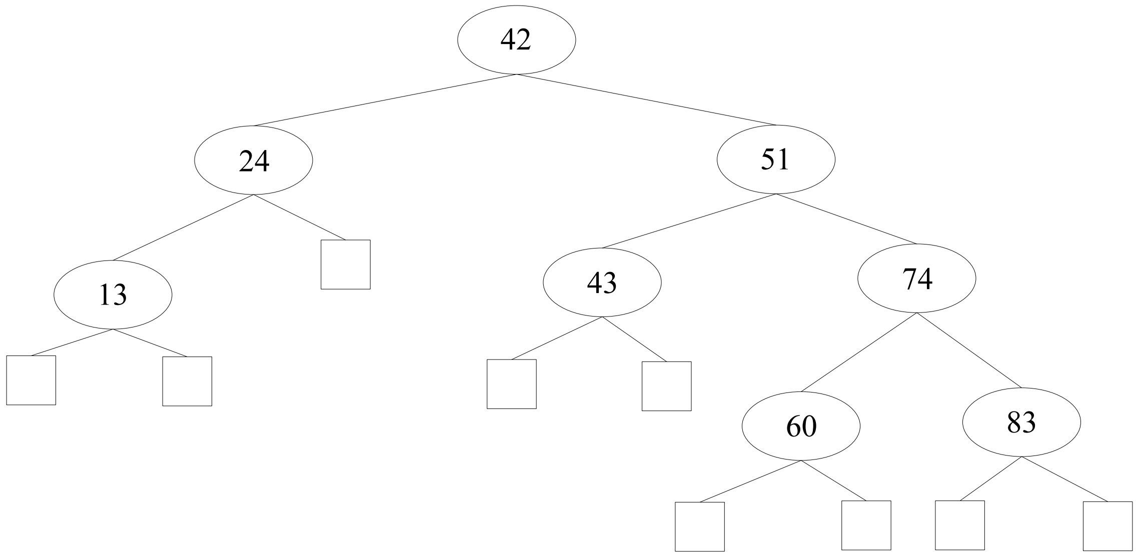
Gegeven zijn een Dequeue A met m positieve integers en een Dequeue B met n positieve integers. De elementen van A zitten niet-dalend in A, dus elk element is tenminste even groot als zijn voorganger. De elementen van B zitten niet-dalend in B. Geef een zo efficiënt mogelijk algoritme *SamenHonderd* dat telt hoeveel paren bestaand uit een element van A en een element van B er zijn waarvan de som precies 100 is. *SamenHonderd* zal dus bijvoorbeeld voor de Dequeues $A = (2, 19, 21, 24, 28, 28, 33)$ en $B = (4, 48, 82, 73, 81)$ het getal 3 opleveren.
Maak voor je algoritme slechts gebruik van de standaardoperaties (ofwel methods):
insertFirst(e), insertLast(e), removeFirst(), removeLast(), first(), last(), 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 *(2 punten)*

Gegeven is een nette (propere) binaire boom T met integer elementwaarden in alle n knopen. Voor elke knoop v definiëren we $SE(T, v)$ als de som van alle even elementwaarden in de deelboom van T met v als wortel, en $SO(T, v)$ als de som van alle oneven elementwaarden in de deelboom van T met v als wortel. We noemen een knoop w een *buitengewone knoop* als $|SO(T, w)| \leq 1$. Geef een zo efficiënt mogelijk algoritme *TelBuitengewoneKnopen(T)* dat telt hoeveel buitengewone knopen er zijn in de nette binaire boom T. Analyseer de looptijd van je algoritme.

Opgave 7 *(1 punt)*

Gegeven is de afgebeelde AVL-boom T.



a) Geef de AVL-boom T_a die ontstaat uit T door verwijdering van de key 51.

b) Geef de AVL-boom T_b die ontstaat uit T door toevoeging van de key 94.