

Tweede Hertoets Datastructuren

14 juli 2017, 13.30 – 15.30, Educatorium-Γ.

Motiveer je antwoorden *kort!* Stel geen vragen over deze toets; als je een vraag niet duidelijk vindt, schrijf dan op hoe je de vraag interpreteert en beantwoord de vraag zoals je hem begrijpt. Elke vraag telt even zwaar nl 2pt. Maak vraag 1 en 2 op pagina 1, vraag 3 en 4 op pagina 2, en vraag 5 en 6 op pagina 3. T2 is totaal plus een half gedeeld door 1,2.

1. **Invoer-reversie met Stack:** Geef een programmafragment dat 100 tekstregels inleest en in omgekeerde volgorde weer afdrukt. Je programma *moet* een Stack gebruiken om de regels tussentijds op te slaan.
2. **Master Theorem:** Geef de oplossing met de Master Theorem:
(a) $T(n) = 7T(n/2) + O(n^2)$ (b) $V(n) = 3V(n/2) + n \lg n$.
3. **Delete uit Heap:** Geef een methode `void Delete(int i)`, die uit een MinHeap A het element op positie *i* verwijdert (in logaritmische tijd; je mag `Rootify` en `Heapify` als gegeven veronderstellen).
4. **Range in Hashtabel:** Gegeven is een hashtabel, waarin collisions zijn opgelost met chaining, en de keys zijn integers.
(a) Geef een methode `RQ(int a, int b)` die alle keys tussen *a* en *b* afdrukt.
(b) Wat is de looptijd van je query, uitgedrukt in *n* en *m* (aantal keys en lijsten). Waarom?
5. **Range Query:** Gegeven is een binaire zoekboom, geaugmenteerd met het aantal keys per deelboom. Elke knoop bevat een integer `key`, deelbomen `left` en `right`, en een integer `size` die het aantal keys in de deelboom geeft.
(a) Geef een *recursieve* methode `int AK(Node b, int x)` die het aantal keys *strikt kleiner dan x* in boom *b* oplevert. De tijd moet lineair zijn in de hoogte van de boom (dus $O(h)$).
(b) Geef een methode `int AR(Node b, int from, int to)` die het aantal keys in *b* in de *range* van *from* tot *to* oplevert (dwz het aantal keys *k* waarvoor $from \leq k < to$).
6. **Logaritmes Berekenen en Bewijzen:**
(a) Hoeveel is $^2 \log 5$ en hoeveel is $^3 \log 7$? Hoe kun je dit uitrekenen mbv. de $^{10} \log$?
(b) Bewijs dat $a^{b \log n} = n^{b \log a}$.