

Tweede Hertoets Datastructuren

13 juli 2016, 13.30–15.30, Olympos-Hal2.

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. 3pt. T2 is totaal plus 1, gedeeld door 1,5.

- De MinStack:** Een *MinStack* kent naast de standaard Stack-operaties, een operatie **MinKey** die de kleinste waarde in de MinStack oplevert (maar niet verwijdert).
 - Wat zijn de standaard operaties om data toe te voegen aan en te verwijderen van een gewone Stack en in hoeveel tijd worden ze uitgevoerd?
 - Geef een implementatie (drie methoden) van MinStack die deze operaties en de **MinKey** in $O(1)$ tijd uitvoert.
- Master Theorem:** Bepaal de asymptotische oplossing van deze recurrenties met de Master Theorem. Geef de a , b en $f(n)$.
 - $V(n) = 4V(n/2) + 2n^2$.
 - $W(n) = 2W(n/2) + O(\lg n)$.
 - $X(n) = 9X(n/3) + 2^n$.
- Enumeratie met diepte:** Gegeven is een binaire zoekboom b waarin elke **Node** velden **key**, **left** en **right** heeft. Geef een methode die alle keys afdrukt van klein naar groot, en bij elke key ook de *diepte* afdrukt, dwz., hoeveel stappen hij onder de wortel zit. **Voorbeeld:** De wortel bevat het getal 6, de linkerdeelboom alleen het getal 4, de rechterdeelboom het getal 8 met links daaronder de 7. Dan moet de uitvoer zijn: 4(1) 6(0) 7(2) 8(1) .
- Hashen van Strings:** Nico bouwt een hashtabel om gegevens van ongeveer een miljoen studenten op te slaan. Hij wil snel kunnen toevoegen, zoeken en verwijderen op *naam*. Om deze opgave niet te moeilijk te maken, nemen we aan dat elke naam een rijtje is van hoogstens 12 letters (a t/m z).
 - Welke twee manieren zijn er om keys met gelijke hashwaarde af te handelen, en welke raad je aan voor Nico en waarom?
 - Nico geeft elke letter een getal (namelijk $a=1$, $b=2$, tot $z=26$) en berekent de HashCode als de lettersom (dus bv Tel heeft HashCode 37). Waarom werkt Nico's tabel erg traag?
 - Geef een functie die wel geschikt is als hash-functie.
- Recurrente betrekking:** Los op: $a_0 = 0$, $a_1 = 2$, $a_n = 2a_{n-1} + 3a_{n-2}$.