

# Tweede Toets Datastructuren

28 juni 2017, 13.30 – 15.30, Educ- $\beta$ .

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.

**Cijfer:** Elke vraag is 2pt. Cijfer T2 is totaal plus 0,5 gedeeld door 1,35. Maak vraag 1 en 2 op pagina 1, 3 en 4 op pagina 2, 5 en 6 op pagina 3 en vraag 7 op de achterkant.

- Binaire Fruitmand:** Floris gebruikt een uint `fm` om een deelverzameling van 30 fruitsoorten op te slaan. De  $i^{\text{de}}$  bit van `fm` staat op `true` wanneer element  $i$  aanwezig is.
  - Geef een instructie die object  $i$  toevoegt (of niets doet als  $i$  er al is) in  $O(1)$  tijd.
  - Floris schrijft de instructie `fm &= (~ 1) << i`; om fruit  $i$  te verwijderen. Waarom klopt deze niet en hoe moet het wel?
- Stack met Pointers:** Een pointer-implementatie van de stack bevat in elke node een stack-element en een pointer naar een andere node (evt. `null`).
  - Als de Node met element  $x$  een pointer naar de Node met  $y$  heeft, is  $x$  dan eerder of later dan  $y$  toegevoegd aan de stack? Waarom?
  - Geef de naam en pseudocode van de methode die element  $z$  toevoegt aan de stack.
- Twee, Acht, Log en Wortel:** Geef de asymptotische oplossing van
  - $W(n) = 2W(n/4) + 2\lg n$ ,
  - $X(n) = 8X(n/4) + n\sqrt{n}$ ,en geef aan hoe je eraan komt.
- Heap:** De rij `A = [12, 15, 20, 19, 17, 32, 26, 23, 21, 18, 20]` is een MinHeap. Zeg stap voor stap wat er gebeurt, en geef de heap na afloop, bij:
  - `Insert(18)` op heap `A`;
  - `ExtractMin()` op heap `A`.
- Zoeken in Hashtabel:** In een hashtabel (met chaining) heeft *succesvol* zoeken een andere analyse dan *succesloos* zoeken. Ga uit van een hashtabel met  $n$  keys en  $m$  lijsten, en load factor  $\alpha = n/m$ .
  - Bij welk van deze twee gevallen druk je de zoektijd uit in de *gemiddelde lijstlengte*? Geef een uitdrukking voor de zoektijd.
  - Bij welk van de twee gevallen hangt de zoektijd af van het *gemiddelde kwadraat van de lijstlengte*? Leg uit waarom.
- Meer dan  $x$ :** Geef een methode `Tree meerdan(Tree t, key x)` die, voor key  $x$  en binaire zoekboom  $t$ , de kleinste waarde  $y$  in  $t$  zoekt die *strikt groter* dan  $x$  is. Lever een pointer naar de knoop waar  $y$  zit, of `null` als er geen key  $> x$  is.
- Recurrente betrekking:** Los op:  $a_0 = 1$ ,  $a_1 = 2$ ,  $a_n = 4a_{n-1} - 3a_{n-2}$ .