Aanvullende toets DataStructuren 1 september 2006
docent Marinus Veldhorst

Schrijf niet met rood en niet met potlood; zet op elk in te leveren vel:
- je naam (met initialen),
- collegekaartnummer,

Besef dat de aangegeven punten bij een opgave niet noodzakelijkerwijs de zwaarte van de opgave aangeven; punten per onderdeel van een opgave kunnen verschillen. Wees zo verstandig om een opgave eerst helemaal door te lezen voordat je een oplossing gaat opschrijven.

1. **(1 punt)**
   Zeg van elk van onderstaande beweringen of hij waar (W) of onwaar (N) is. Schrijf je antwoord bij de bewering zelf, en lever het vel bij verlaten van de zaal in.
   P.S. k goede antwoorden levert een score van $\frac{\max(0,k-2)}{6}$ punt op.

   (a) Bewering: Voor elke integer $h \geq 0$ en voor elke propere binaire boom T van hoogte h is er een knoop $u \in T$ zodanig dat $depth(u) + height(u) = h$.

   (b) Zij $\mathbb{N}$ de verzameling van positieve gehele getallen (dus zonder het getal 0). Zij f en g elk een functie van $\mathbb{N}$ naar $\mathbb{N}$. Zij $h : \mathbb{N} \to \mathbb{N}$ gedefinieerd als $h(n) = \lceil \frac{f(n)}{g(n)} \rceil$.
   Bewering: Als f is $O(n^4)$ en g is $O(n^2)$ dan geldt dat h is $O(n^2)$.

   (c) Bewering: Voor elke heap H geldt dat als men H op inorder wijze doorloopt, men de items in H in stijgende volgorde bezoekt.

   (d) Bewering: Het algoritme binary search op een geordende array werkt in tijd $\Omega(\log(n^2))$.

   (e) Bewering: Quicksort is stabiel en in-place.

   (f) Bewering: Er is een algoritme dat, gegeven een array A met daarin van klein naar groot n keys, in $O(n)$ tijd een heap bouwt met daarin de keys van A.

   (g) Bewering: Gebruik van hashing met open adressering en collision handling door middel van double hashing leidt tot secondary clustering (bij elke functie die men voor double hashing gebruikt).

   (h) Zij T een functie met als parameter een niet-negatief geheel getal en als uitkomst een niet-negatief geheel getal. Stel $T(0) \leq T(1) \leq T(2) \leq 4$ en voor $n \geq 3$ is $T(n) = 2T(\lfloor n/3 \rfloor) + \lceil 2(n^{3/2}) \rceil$.
   Bewering: T is $\Theta(n)$.

2. **(1 punt)** 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 waarvoor geldt dat x òf in A òf in B voorkomt, maar niet in beide voorkomt.
   Na afloop van je algoritme moet A hetzelfde zijn als bij de start van het algoritme. Dit geldt ook voor B. Na afloop van je algoritme moeten de integers in C van klein naar groot in C zitten, en mag C geen dubbelen bevatten.
   Geef je algoritme in **pseudocode**, en gebruik voor het access van A, B en C alleen de operaties behorend bij het abstract data type **queue**. Verder is er nog de eis dat je algoritme werkt in tijd $O(n + m)$ 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).

1

3. **Sorteren**

   (a) **(0.5 punt)** Geef de definitie wanneer een sorteer algoritme een comparisonsort is. Noem een sorteeralgoritme die een comparisonsort is en noem een sorteeralgoritme die geen comparisonsort is. Beide algoritmen moeten behandeld zijn in de cursus.

   (b) **(2 punten)** Bewijs dat elk comparisonsort algoritme toegepast op n items in het slechtste geval $\Omega(n \log n)$ tijd vergt.
   OPMERKING. Je mag hier niet meer dan één kantje A4 in **klein** handschrift over vol schrijven. Wees dus compact in je formuleringen en bedenk wat je wel en niet opschrijft.

4. **AVL-bomen**

   (a) **(0.5 punt)** Geef de definitie van een zoekboom en van een AVL boom.

   We beschouwen nu propere binaire bomen opgeslagen in een linked structure: er is een pointer naar de wortel, en per interne knoop is opgeslagen een integer waarde, een pointer naar het linker kind en een pointer naar het rechter kind (er is geen pointer naar de parent).

   (b) **(2 punten)** Geef in **pesudo-code** een lineaire tijd algoritme die voor een gegeven propere binaire boom T de waardes van alle interne knopen v print waarvoor geldt dat $T(v)$ een ALV-boom is. Doe dit met één traversal over T.
   Met $T(v)$ bedoelen we die deelboom van T die v als wortel heeft.

5. **Heap-achtige structuren**
   In deze opgave kijken naar een (2,3)-heap implementatie van het abstracte data type PRIORITY QUEUE.

   Definitie. Een *(2,3)-heap* H van hoogte h is een boom (met wortel r) van hoogte h waarin elke interne knoop graad[1] 2 of graad 3 heeft; de boom voldoet verder aan de volgende 3 eisen:

   - alle externe knopen van H hebben diepte h,
   - elke interne knoop $v \in H$ met graad g ($g = 2$ of $g = 3$) bevat $g - 1$ items (type integer), aangeduid met $v_1, \ldots, v_{g-1}$,
   - voor elke interne knoop $v \in H$ ongelijk aan de wortel r van H geldt: elk item in v is minstens zo groot als elk item in de parent van v.

   Einde definitie

   Je mag er van uitgaan dat (2,3)-heaps opgeslagen zijn in een linked structure met (als je dat wenst) ook de parent pointers.

   (a) **(0.5 punt)** Geef een voorbeeld van een (2,3)-heap H met de twaalf items $1, 2, 3, 4, \ldots, 12$.

   (b) **(0.5 punt)** Bewijs dat een (2,3)-heap met n items een hoogte heeft van $O(\log n)$.

   (c) **(1 punt)** Geef een $O(\log n)$ tijd algoritme dat uit een (2,3)-heap met n items een kleinste item verwijdert.
   Beargumenteer dat je algoritme correct is en voldoet aan de tijdgrens.

   (d) **(1 punt)** Zij gegeven twee (2,3)-heaps H_1 en H_2 met respectievelijk n en m items. We willen H_1 en H_2 samenvoegen tot een (2,3)-heap H_3 waarbij H_1 en H_2 na afloop van de samenvoeging beide leeg zijn. Geef een algoritme dat deze samenvoeging uitvoert in tijd $O(\log n + \log m)$.
   Beargumenteer dat je algoritme correct is en voldoet aan de tijdgrens.

einde toets

[1] Tijdens het tentamen is meegedeeld dat de graad van een knoop v in een boom met een wortel is gedefinieerd als het aantal kinderen van v.

2