Deeltentamen DataStructuren 11 juni 2003
docent Marinus Veldhorst

Schrijf niet met rood en niet met potlood; zet op elk in te leveren vel:

- je naam (met initialen),
- collegekaartnummer,
- je werkcollegeleider; liefst de werkcollegeleider die je tentamen van 28 mei j.l. heeft nagekeken. Deze werkcollegeleider zal je werk nakijken en het (zo mogelijk) tijdens het werkcollege van 16 of 18 juni teruggeven.
- Zet op het eerste vel het totaal aantal vellen papier dat je inlevert.

Je hebt 1 uur en 3 kwartier de tijd voor dit deeltentamen.

5. **Traversals**

   Zij T een propere (nette) binaire boom van n knopen. We zeggen dat een interne knoop v in T een **x-node** is als geldt

   $$5 \cdot h(v) \leq N(v) \leq 10 \cdot h^2(v)$$

   waarbij $h(v)$ de hoogte van v is, en $N(v)$ het aantal afstammelingen van v is. Bedenk dat v een afstammeling van zichzelf is.

   Interne knopen in T hebben de volgende inhoud: een pointer naar het linker kind, een pointer naar het rechter kind, een info-veld (printable in $O(1)$ tijd), maar **geen** parent pointer.

   (a) **1 punt**
   Teken een nette binaire boom met minstens 3 x-nodes.

   (b) **2 punten**
   Geef in pseudo-code een algoritme die van alle x-nodes in T het info-veld print. Je algoritme moet werken in $O(n)$ tijd. Geef (zo mogelijk) je algoritme als één traversal van T.
   Leg uit dat je algoritme correct is, en voldoet aan de gestelde tijdgrens.

   (c) **1 punt**
   Kun je een nette binaire boom aangeven waarbij voor de wortel r geldt dat:

   $$N(r) \geq 2 + 10 \cdot h^2(r) \text{ ?}$$

   Beargumenteer je antwoord.

Z.O.Z.

6. **(1.5 punt)**
   Zij T een binaire boom (niet noodzakelijkerwijs proper), en zij v en w knopen in T; v en w kunnen dezelfde knoop zijn. We willen het volgende probleem oplossen:

   Quest: gegeven pointers v en w die elk naar een knoop in dezelfde boom T wijzen; return **true** dan en slechts dan als in een preorder traversal van T de knoop aangewezen door v eerder dan of gelijktijdig met de knoop aangewezen door w bezocht **zou worden**.

   Elke interne knoop x in T heeft de volgende inhoud: naar elk kind van x een pointer, en een pointer naar de parent van x.

   (a) Geef voor **Quest** een algoritme die werkt in $O(1 + \max\{d(v), d(w)\})$ tijd waarbij voor elke knoop x in T $d(x)$ de diepte van x in T is.
   Beargumenteer dat je algoritme correct is, en voldoet aan de tijdgrens.

7. **(1.5 punt)**
   Zij T een binaire boom; T is niet noodzakelijkerwijs proper, oftewel, T kan knopen bevatten die precies één kind hebben.
   Zij D_2 het aantal knopen in T die elk 2 kinderen hebben, D_1 het aantal knopen in T die elk 1 kind hebben, en D_0 het aantal knopen die elk 0 kinderen hebben.

   (a) Bewijs met inductie dat $D_2 \leq D_0 - 1$.

   Besteed zorg aan de preciese opbouw van je bewijs alsmede aan de precisie van formuleringen.

8. **(3 punt)**
   Stel we hebben een **sequence** S van n integers. Neem aan dat S geen dubbelen bevat. S was geordend van klein naar groot.

   Op een of andere wijze zijn de k grootste elementen uit S genomen en op willekeurige plekken in S teruggezet. Hiermee heeft S wel zijn zelfde omvang behouden, maar de ordening van S is verloren gegaan. Ga ervan uit dat deze gewijzigde S in een array van lengte n is opgeslagen.

   (a) Ontwerp een sorteeralgoritme die in $O(nk)$ tijd S sorteert van klein naar groot, waarbij k het bovengenoemde aantal verwijderde en teruggezette elementen is; dit getal k is invoer voor je algoritme. Beargumenteer de correctheid van je algoritme, en dat het voldoet aan de gestelde tijdgrens.
   OPMERKING Je algoritme (excl. de verder gevraagde argumentatie) mag niet meer dan één A4-kantje tekst omvatten. Als je algoritme langer wordt (dreigt te worden), moet je je beperken tot hoofdlijnen en essentiële details van je algoritme.

   (b) Kun je dezelfde vraag ook beantwoorden als niet zozeer de k grootste elementen weggehaald en willekeurig terug geplaatst zijn, maar k willekeurige elementen? Zo ja, doe dat dan; zo nee, waarom zou er dan geen $O(nk)$ sorteeralgoritme zijn? In geval je een algoritme geeft, dien je ook een correctheidsargumentatie en een tijdanalyse te geven.
   Ook nu kun je het aantal k gebruiken als parameter voor je algoritme.

——————————- *einde deeltentamen* ——————————

2