Deeltentamen DataStructuren 19 mei 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. Wees zo verstandig om een opgave eerst helemaal door te lezen voordat je een oplossing gaat opschrijven.

1. **(2.5 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 in A of B (of 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).

2. **(2.5 punt)**
   Gegeven het volgende algoritme:

   Algorithm $weird(A)$:
   Input: een array A van n^2 integers; A heeft grenzen 0 en n^2-1
   Output: een array S van n^2 integers berekend volgens onderstaand methode;

   for $i \leftarrow 1$ **to** $n \times n$
   do $xx \leftarrow 0$;
   integer $logi \leftarrow \lfloor \log_2 i \rfloor$; $\lfloor\ \rfloor$ rondt naar beneden af
   for $j \leftarrow 0$ **to** $n \times n \times logi - 1$
   do $xx \leftarrow xx + A[(3 \times j) \bmod (n \times n)] + logi$ **enddo**
   $S[i-1] \leftarrow \lfloor xx/((i+1) \times (logi+1)) \rfloor$
   enddo
   return S

   Maak een zo goed mogelijke tijdanalyse (worst-case) van algoritme *weird*, in termen van grote O van een functie in n.
   Je mag ervan uit gaan dat $\lfloor \log_2 i \rfloor$ in tijd $O(\log i)$ berekend wordt uit i, dat **return** S in constante tijd plaats vindt, en dat de operaties mod en $\lfloor\ \rfloor$ elk ook in constante tijd werken.

1

3. **(2 punt)** Stel we hebben een enkelvoudig gelinkte lijst L met header en trailer sentinels (schildwachten) en n knopen. De knopen van L bevatten elk een integer. L is niet noodzakelijkerwijs geordend naar deze integer waarden.
Geef in pseudo code een algoritme dat, gegeven een pointer naar de header van L, gegeven een integer p en gegeven een integer q, het volgende teweeg brengt in L:

   > Zij pp een knoop in L die p bevat, en qq een knoop in L die q bevat; verwissel in L de knopen pp en qq. Als p of q niet aanwezig is in L, dan mag het algoritme geen verandering aanbrengen in L[1].

   Je algoritme moet werken in tijd $O(n)$ en mag naast de lijst L slechts een constante hoeveelheid geheugen gebruiken.

   Beargumenteer dat je algoritme correct is en voldoet aan de tijd- en geheugengrens.

4. **(3 punt)**

   (a) Stel we hebben een array A van n integers. A is ooit gesorteerd van klein naar groot, maar sindsdien is precies één element van A gewijzigd. We weten echter niet welk element, en ook niet zijn index.
   Ontwerp een $O(n)$ tijd algoritme dat A sorteert (van klein naar groot). Beargumenteer dat je algoritme correct is en voldoet aan de tijdgrens.

   (b) Zelfde situatie als bij opgave **4 (a)**, maar nu zijn er precies d elementen van A gewijzigd. d is bekend, maar we weten niet welke elementen het betreft, noch hun indices. Ontwerp een $O(dn)$ tijd algoritme dat A sorteert. Beargumenteer dat je algoritme correct is en voldoet aan de tijdgrens.

——————————- *einde toets* ——————————

[1] Tijdens het tentamen, ongeveer halverwege, is door de docent gemeld dat bij meerdere voorkomens één knoop met p verwisseld moet worden met één knoop met q.

2