Tentamen Algoritmen en Datastructuren

dinsdag 29 januari 2008, 14 - 17 uur

Het tentamencijfer T is (p/10) + 1, waarbij p het totaal aantal behaalde punten is.

Met de zinsnede 'geef een algoritme' in een opgave wordt bedoeld:

beschrijf een algoritme in pseudocode (dus niet in Java),
licht de werking ervan toe,
beargumenteer de correctheid.

1. (30 punt) Deze opgave gaat over ongeordende rijen (sequences) van getallen. De getallen in een rij hoeven niet verschillend te zijn. Zij n het aantal elementen van rij R.
   Selectie is het vinden van het k-de kleinste element in de rij, dwz. van een element x van de rij met de eigenschap dat er minder dan k elementen in de rij kleiner zijn dan x, en hoogstens $n - k$ elementen groter dan x.
   Partitie van rij R, gegeven een getal k, is het splitsen van R in drie deelrijen $K = (m \in R \mid m < k), I = (m \in R \mid m = k)$ en $G = (m \in R \mid m > k)$.

   (a) Leg uit hoe selectie vrij gemakkelijk in $O(n \log n)$ tijd kan worden uitgevoerd.

   (b) Geef een (eenvoudig) algoritme voor partitie. De volgende $O(1)$-methoden voor rijen zijn beschikbaar: isEmpty(), first(), size(), insert(x), remove(p) (x is een getal, p is een positie). Wat is de tijdscomplexiteit van het algoritme?

   (c) Geef het algoritme quickSelect, het non-deterministische algoritme voor selectie dat gebruik maakt van partitie. Gebruik de in het vorige onderdeel genoemde methoden, en de nondeterministische methode random() die een willekeurig element uit een rij kiest.

   (d) Beargumenteer dat de worst-case tijdscomplexiteit van quickSelect $O(n^2)$ is.

   (e) Bewijs dat de verwachte tijdscomplexiteit van quickSelect $O(n)$ bedraagt. (Hint: onderscheid goede en slechte partities, en beargumenteer dat verwacht mag worden dat de helft van de partities goed is.)

2. (30 punt) Gegeven zijn n positieve gehele getallen $c_1, \cdots, c_n$ en een positief geheel getal K. Gevraagd: is er een deelverzameling S van $\{1, \cdots, n\}$ met de eigenschap

$$\sum_{i \in S} c_i = K \ ?$$

   (a) Beargumenteer dat een naïeve aanpak van dit probleem leidt tot een algoritme met exponentiële tijdscomplexiteit. (Je hoeft het algoritme niet in pseudocode te beschrijven.)

   (b) Geef een algoritme voor dit probleem dat gebruik maakt van dynamisch programmeren.

   (c) Geef een definitie van het begrip *pseudo-polynomiaal* in de context van tijdscomplexiteit. Laat zien dat het in (b) gevonden algoritme pseudo-polynomiale tijdscomplexiteit heeft. Laat ook zien dat het *geen* polynomiale tijdscomplexiteit heeft.

3. (30 punt) Deze opgave gaat over complexiteitsklassen en de beslissingsproblemen SAT en CIRCUIT-SAT. SAT is als volgt gedefinieerd:

   > gegeven een formule A in de propositielogica die alleen de connectieven $\neg, \wedge, \vee$ bevat, is er een toekenning van waarheidswaarden aan de variabelen in A zodanig dat A de waarde *true* krijgt?

   CIRCUIT-SAT is als volgt gedefinieerd:

   > gegeven een logische schakeling S opgebouwd uit niet-, en- en of-poorten, is er een toekenning van nullen en enen aan de invoerdraden van S zodanig dat de uitvoerdraad van S de waarde 1 krijgt?

   (a) Geef definities van de complexiteitsklassen P (polynomiaal) en NP (niet-deterministisch polynomiaal).

   (b) Toon aan dat CIRCUIT-SAT in NP zit.

   (c) Er geldt dat SAT NP-volledig (NP-*complete*) is. Leg uit wat dit betekent.

   (d) Gebruik het feit dat SAT NP-volledig is om aan te tonen dat CIRCUIT-SAT NP-volledig is. (NB: let op de goede richting van de probleemreductie.)

   (e) Leg uit waarom het onwaarschijnlijk wordt gevonden dat SAT in P zit.