Tentamen Gevorderde Algoritmen en Datastructuren

woensdag 18 april 2012, 14 - 17 uur

Je hoeft slechts 4 van de 5 opgaven te maken: de opgave met het minste aantal punten telt niet mee. Elke opgave levert maximaal 25 punten op. Het tentamencijfer T is (p/10), waarbij p de som is van de vier hoogste aantallen punten per opgave.

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

> **beschrijf een algoritme in pseudocode, licht de werking ervan toe, beargumenteer de correctheid**.

1. Deze opgave gaat over het gebruik van een hash-tabel $T[0..m-1]$ voor het opslaan van niet–negatieve gehele getallen. De hashfunctie h is gedefinieerd door $h(i) = i \bmod m$. We gebruiken *open addressing* met *linear probing* voor het vermijden van botsingen (*collisions*). Dat wil zeggen: als getal i in de tabel geplaatst moet worden en positie $h(i)$ is niet leeg, dan worden de posities $h(i+1), h(i+2), \ldots, h(i+m-1)$ langsgelopen en wordt i geplaatst in de eerste lege positie. We nemen aan dat er altijd een lege positie in de tabel is.
   Het algoritme voor het opzoeken van een getal in de hash-tabel is als volgt.

   algoritme ZoekOp(n, T)
       invoer getal n, hash-tabel $T[0..m-1]$
       uitvoer de positie k met $T[k] = n$ als die er is, anders de melding 'is er niet'
       $k \leftarrow n \bmod m$
       while $T[k] \neq NULL$ **do**
           if $T[k] = n$ **then**
               return k
           $k \leftarrow (k+1) \bmod m$
       return 'is er niet'

   Geef efficiënte algoritmen voor het toevoegen en verwijderen van een getal in resp. uit de hash-tabel. Houd rekening met het ontstaan van 'gaten' bij het verwijderen van getallen door gebruik te maken van de speciale waarde BESCHIKBAAR.

2. Deze opgave gaat over ongeordende rijen R van getallen. De getallen in een rij hoeven niet verschillend te zijn. Zij n het aantal elementen van R.
   Selectie is het vinden van het k-de kleinste element in R, dwz. van een element x van R met de eigenschap dat er minder dan k elementen in R 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(R) die een willekeurig element uit rij R kiest.

(d) Laat zien dat de *worst case* tijdscomplexiteit van quickSelect $O(n^2)$ is.

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

3. Zij gegeven een ongerichte, samenhangende, enkelvoudige (dwz. zonder selfloops en parallelle kanten) gewogen graaf G, waarin alle gewichten verschillend zijn.

   (a) Wat is een opspannende boom (spanning tree) van G? En een minimale opspannende boom?

   (b) Zij V_1, V_2 een partitie van de knopen van G. Beschouw alle kanten van G die een eindpunt in zowel V_1 als in V_2 hebben, en laat e van die kanten het laagste gewicht hebben. Laat zien: elke minimale opspannende boom van G bevat e.
   Aanwijzing: stel dat er wel een minimale opspannende boom T is die e niet bevat, voeg dan e toe aan T. Wat gebeurt er dan met T? Hoe kan je nu tegenspraak krijgen met de minimaliteit van T?

   (c) Geef een algoritme dat een minimale opspannende boom van G vindt, met tijdscomplexiteit $O(m \log n)$ (n het aantal knopen, m het aantal kanten). Je mag gebruik maken van een efficiënte priority queue.

4. Gegeven zijn n positieve gehele getallen $c_1, \cdots, c_n$ en een positief geheel getal K. Gevraagd: is er een deelverzameling S van $\{1, \ldots, n\}$ 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) Wat is de tijdscomplexiteit van het in de vorige deelvraag gevonden algoritme? Beargumenteer je antwoord.

5. (a) Leg uit wat het beslissingsprobleem CIRCUIT-SAT inhoudt.

   (b) Geef een definitie van de complexiteitsklasse NP.

   (c) Laat zien dat CIRCUIT-SAT in NP zit.

   (d) Zit CIRCUIT-SAT ook in de complexiteitsklasse P? Beargumenteer je antwoord.