Tik–106.4100 Algoritmien suunnittelu ja analyysi, syksy 2007 Tentti, 14.12.2007

Kirjoita jokaisen palauttamasi paperin yläreunaan selvästi "T-106.4100 Algoritmien suunnittelu ja analyysi, 14.12.2007", nimesi, opiskelijanumerosi ja koulutusohjelmasi sekä palauttamiesi paperien kokonaismäärä.

1. a) (2p) Mitkä seuraavista väittämistä pitävät paikkansa ja mitkä eivät? Perustele vastauksesi matemaattisesti!

   i. $n^2 + 7n \in O(n^3)$

   ii. $14n^2 + 5n \log n + 100 \in \Theta(n^2)$

   b) (2p) Kerro muutamalla virkkeellä, miten voidaan laskea jonkun algoritmin keskimääräinen aikavaativuus. Kerro yleinen periaate, älä selosta vain jonkun tietyn algoritmin keskimääräisen aikavaativuuden laskemista. (Voit kuitenkin halutessasi selventää vastaustasi käyttämällä jotain algoritmia esimerkkinä, kunhan yleinen periaate käy selväksi.)

   c) (2p) Kerro, miten tasoitettu aikavaativuus (engl. amortized complexity) eroaa keskimääräisestä aikavaativuudesta.

2. a) (3p) Ratkaise seuraava rekursioyhtälö, kun n on kahden potenssi. Anna täsmällinen ratkaisu (suuruusluokka ei riitä).

$$T(n) = \begin{cases} 1, & \text{kun } n = 1 \\ 4T(n/2) + 3n^2 & \text{kun } n > 1 \end{cases}$$

   b) (3p) Arvaa hyvä ratkaisu seuraavalle palautuskaavalle ja todista ratkaisusi oikeaksi induktiolla.

$$T(n) \leq \begin{cases} 1, & \text{kun } 1 \leq n \leq 3 \\ T(n-3) + 2n & \text{kun } n > 3 \end{cases}$$

4→9 7→23 10→43
5→11 8→27 11→49
6→13 9→31 12→55

3. a) (1p) Mihin kekoa (engl. heap, esimerkiksi binääri-, binomi- tai Fibonacci-kekoa) voidaan käyttää Primin algoritmin toteutuksessa?

   b) (5p) Miten binomi- ja Fibonacci-keot poikkeavat toisistaan? Miten erot vaikuttavat eri operaatioiden aikavaatimuksiin näissä rakenteissa?

4. (6p.) Tarkastellaan suunnattua verkkoa $G = (V, E)$, jonka solmut ovat $v_1, v_2, \ldots, v_n$. Määritellään, että verkko on *järjestetty verkko*, jos sen solmuille ja kaarille pätee seuraavat kaksi ehtoa:

   - Verkon kaikki kaaret ovat muotoa (v_i, v_j) siten, että $i < j$. Toisin sanoen verkon kaikkien kaarten lähtösolmun indeksi on pienempi kuin tulosolmun indeksi. Verkon kaikkien tällaisten solmuparien välillä ei kuitenkaan välttämättä ole kaarta.
   - Verkon jokaisesta solmusta solmua v_n lukuunottamatta lähtee vähintään yksi kaari. Toisin sanoen jokaiselle $v_i, i = 1, 2, \ldots, n-1$ on vähintään yksi kaari, joka on muotoa (v_i, v_j).

   Laadi dynaamista ohjelmointia käyttävä algoritmi, joka etsii järjestetyssä verkossa G pisimmän solmusta v_1 solmuun v_n johtavan polun. Verkon kaarilla ei ole painoja, vaan polun pituudella tarkoitetaan polkuun kuuluvien kaarten määrää.

   Älä kirjoita algoritmisi koodia, vaan selitä algoritmisi periaate, esitä laskennassa käytettävät lausekkeet ja niiden arvojen laskentajärjestys sekä kerro, miten pisimmälle polulle kuuluvat solmut voidaan selvittää (pisimmän polun pituuden laskemisen lisäksi).

   Ratkaisusta ei saa täysiä pisteitä, jos se ei käytä dynaamista ohjelmointia.

5. (6p) Kirjoita pseudokoodi algoritmille, joka luokittelee suunnatun verkon kaaret puukaariin, eteneviin kaariin, takautuviin kaariin ja poikittaiskaariin. Algoritmin tulee tulostaa jokaisesta kaaresta (u, v) sen päätepisteet u ja v sekä se, mihin edellä mainituista luokista tämä kaari kuuluu. Mikä on algoritmisi aikavaativuus? Perustele aikavaativuus lyhyesti.

Muista vastata kurssin palautekyselyyn! Linkki kyselyyn on kurssin kotisivulla.