Tik–106.4100 Algoritmien suunnittelu ja analyysi, syksy 2008

Tentti 4.9.2009

Kirjoita jokaisen palauttamasi paperin yläreunaan selvästi "T-106.4100 Algoritmien suunnittelu ja analyysi, 4.9.2009", nimesi, opiskelijanumerosi ja koulutusohjelmasi sekä palauttamiesi paperien kokonaismäärä.

1. a) (3p) Mitkä seuraavista väittämistä pitävät paikkansa ja mitkä eivät? Perustele lyhyesti! (Lyhyt sanallinen perustelu riittää, tarkkoja matemaattisia todistuksia ei vaadita.)
   i. $8n^3 + 5n + 7 \in O(n^4)$
   ii. $3n \log n + 8n \in \Omega(n^2)$
   iii. $7n^2 + 4n \log n + 100 \in \Theta(n^2)$

   b) (3p) Selitä, mikä ero on keskimääräisellä vaativuudella (averge case complexity) ja tasoitetulla vaativuudella (amortized complexity).

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

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

   b) (3p) Arvaa hyvä ratkaisu seuraavalle palautuskaavalle ja todista ratkaisusi oikeaksi induktiolla (c_1 ja c_2 ovat vakioita).

$$T(n) \leq \begin{cases} c_1, & \text{kun } n = 1 \\ 3T(n-1) + c_2 & \text{kun } n > 1 \end{cases}$$

3. a) (2 p) Kerro lyhyesti, mihin Union–Find-rakenteita käytetään. Esitä myös esimerkki algoritmista, jossa voidaan käyttää apuna UNION-FIND-rakennetta ja selitä, mihin rakennetta käytetään ko. algoritmissa. Algoritmia ei tarvitse kuvata yksityiskohtaisesti vaan riittää, että UNION-FIND-rakenteen käyttötarkoitus ko. algoritmissa käy selväksi.

   b) (2 p) Miksi Union–Find-rakenteissa pyritään siihen, että yhtä joukkoa kuvaava puurakenne on mahdollisimman matala?

   c) (2 p) Kerro, miten Union–Find-rakenteissa voidaan suorittaa Union- ja Find-operaatiot niin, että edellisen kohdan tavoitteeseen päästään. Riittää, että esität kumpaakin operaatiota kohti yhden järkevän tavan. Käytä kuvaa havainnollistamaan esitystäsi.

4. (6 p.) Tarkastellaan seuraavaa ongelmaa: n sanaa, joiden pituudet ovat $l_1, l_2, \ldots l_n$ halutaan tulostaa annetussa järjestyksessä peräkkäisille riveille mahdollisimman kauniisti (määritelty myöhemmin). Sanat tulostetaan siten, että kahden samalla rivillä olevan peräkkäisen sanan välissä on tasan yksi välilyönti. Riville mahtuu M merkkiä. Jos riville tulostettavista sanoista ja niitä erottavista välilyönneistä tulee yhteensä vähemmän kuin M merkkiä, tulostetaan rivin loppuun ylimääräisiä välilyöntejä tarvittava määrä. Mahdollisimman kauniissa tulostuksessa rivien lopuissa olevien ylimääräisten välilyöntien määrän kuutioiden summa, siis summa $\Sigma_{i=1}^{k} {t_i}^3$ on mahdollisimman pieni. Lausekkeessa t_i on i:nnen rivin lopussa olevien ylimääräisten välilyöntien määrä ja k tulostuksessa tarvittavien rivien määrä (huomaa, että rivien määrä vaihtelee sen mukaan, mihin rivinvaihdot on päätetty sijoittaa). Esitä periaate dynaamiseen ohjelmointiin perustuvalle algoritmille, joka laskee, miten sanat pitää tulostuksessa jakaa eri riveille, jotta tulostus olisi mahdollisimman kaunis. Älä kirjoita algoritmiasi pseudokoodina, vaan selitä vain sen periaate sekä se, miten ja missä järjestyksessä tarvittavien lausekkeiden arvot lasketaan. (Tehtävän helpottamiseksi myös viimeisen rivin lopussa olevat ylimääräiset välilyönnit otetaan huomioon minimoitavassa summassa, vaikka käytännössä olisi järkevää jättää ne pois.)

5. (6p) Tämä tehtävä liittyy verkon maksimaalisen virtauksen laskemiseen.

   Olkoon annettu suunnattu verkko $G = (V, E)$ siten, että jokaiseen kaareen $(u, v) \in E$ on liitetty sen kapasiteetti $c(u, v)$ ja kaarta pitkin tällä hetkellä menevä virtaus $f(u, v)$. Olkoon edelleen jo laskettu verkon tämänhetkistä virtausta vastaava jäännösverkko (engl. residual network) $G(f)$. Kirjoita pseudokoodi algoritmille, joka laskee jäännösverkon avulla yhden lyhimmän täydennyspolun (engl. augmenting path) solmusta s solmuun t ja täydennyspolkua vastaavan täydennyksen. Täydennyspolun ei tarvitse olla paras mahdollinen, mutta sen on oltava mahdollisimman lyhyt.

   Algoritmi saa siis syötteenä jäännösverkon sekä solmut s ja t ja sen tulee tulostaa täydennyspolkuun kuuluvat kaaret sekä polkua vastaava täydennys.