Tik–106.410 Algoritmien suunnittelu ja analyysi, syksy 2004

Tentti, 21.12.2004

Kirjoita jokaisen palauttamasi paperin yläreunaan selvästi "T-106.410 Algoritmien suunnittelu ja analyysi, 21.12.2004", nimesi, opiskelijanumerosi ja koulutusohjelmasi sekä palauttamiesi paperien kokonaismäärä.

Huom! Vastaa vain viiteen kysymykseen. Jos vastaat kuuteen kysymykseen, parhaat pisteet saanut vastaus pudotetaan pois loppupisteitä laskettaessa.

1. a) (3p) Mitkä seuraavista väittämistä pitävät paikkansa ja mitkä eivät? Perustele lyhyesti!
   i. $3n^2 + 11 = \Omega(n \log n)$
   ii. $5n^3 + 8n^2 = O(n^4)$
   iii. $7n^2 + 5n \log n + 1000000 = \Theta(n^2)$

   b) (3p) Selitä, miten tasoitettu vaativuus (amortized complexity) ja keskimääräinen vaativuus (average case complexity) eroavat toisistaan. Miksi tasoitetun vaativuuden laskemiseen käytetään usein pankkitili- eli potentiaalimenetelmää sen sijaan, että vaativuus laskettaisiin suoraviivaisemmin?

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

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

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

$$T(n) \leq \begin{cases} c_1, & \text{kun } n = 1 \\ T(n/2) + c_2 & \text{kun } n > 1 \end{cases}$$

3. a) (2p) Kerro, miten Union-Find-rakenteissa voidaan suorittaa Union- ja Find-operaatiot niin, että kutakin joukkoa kuvaava puurakenne pysyy matalana? Riittää, että esität kumpaakin operaatiota kohti yhden järkevän tavan. Käytä kuvaa havainnollistamaan selitystäsi. Mitä hyötyä puurakenteen mataluudesta on?

   b) (4p) Esitä kaksi Fibonacci-keon operaatiota, joilla tasoitettu aikavaativuus on eri kuin pahimman tapauksen aikavaativuus. Anna näiden operaatioiden molemmat aikavaativuudet ja kerro, mihin nämä aikavaativuustulokset perustuvat (tarkkoja matemaattisia johtoja ei tarvitse esittää).

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.

Viimeinen tehtävä on seuraavalla sivulla

6. (Tätä tehtävää ei pysty ratkaisemaan vain kurssin opetusmonisteen tiedoilla, vaan tarvitset lisäksi marraskuun luennoilla esitettyjä asioita.)

a) (3p) Oletetaan, että sinun pitäisi osoittaa kauppamatkustajan päätösongelman (sisältääkö annettu täydellinen verkko Hamiltonin syklin, jonka pituus on korkeintaan k) olevan NP-täydellinen. Miten voit periaatteessa osoittaa sen? Älä esitä itse todistusta, vaan kerro vain, mitä sinun pitää periaatteessa tehdä. Perustele myös, miksi kuvaamasi todistus riittää osoittamaan kauppamatkustajan päätösongelman NP-täydellisyyden.

b) (3p) Selitä muutamalla virkkeellä, miten joko alkuperäinen LZ78- tai LZW-tekstintiivistysalgoritmi toimii. (Selitä siis näistä valintasi mukaan vain toinen.) Anna myös lyhyt esimerkki, joka selventää algoritmin toimintaa.

Muista vastata kurssin palautekyselyyn kurssin kotisivulla! Kyselyyn vastaaminen lasketaan neljäksi laskuharjoitustehtäväksi.