Tik–106.4100 Algoritmien suunnittelu ja analyysi, syksy 2011

Tentti 14.12.2011 **Ei apuvälineitä eikä laskimia**

Kirjoita jokaisen palauttamasi paperin yläreunaan selvästi kurssin koodi ja nimi sekä tentin päivämäärä, nimesi, opiskelijanumerosi ja tutkinto-ohjelmasi 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. $4n^3 + 5n \log n \in O(n^4)$
   ii. $3n \log n + 5n \in \Omega(n)$
   iii. $2n \log n + 5n + 16 \in \Theta(n^2)$

   b) (3p) Selitä, mitä tarkoitetaan algoritmin keskimääräisellä aikaavaativuudella (engl. average case time complexity), mitä tarkoitetaan algoritmin tasoitetulla aikavaativuudella (engl. amortized time complexity) ja mitä eroa näillä kahdella on.

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

$$T(n) = \begin{cases} 1, & \text{kun } n = 1 \\ 3T(n/5) + 2n & \text{kun } n > 1 \end{cases}$$

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

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

3. a) (2p) Kerro, mihin kekoja (esim. binomikekoja) käytetään. Anna esimerkki jostain algoritmista, joka käyttää tärkeimpiä keko-operaatioita. Itse algoritmia ei tarvitse tässä kuvata perusteellisesti. Riittää, että kerrot lyhyesti, mihin algoritmia käytetään ja mihin tarkoitukseen algoritmi käyttää tärkeitä keko-operaatioita.

   b) (4p) Miten binomi- ja Fibonacci-keot poikkeavat toisistaan? Minkä keko-operaatioiden aikavaatimukset ovat näissä rakenteissa erilaiset? Esitä lyhyesti eri keko-operaatioiden aikavaatimukset kummassakin rakenteessa ja perustele lyhyesti, mistä nämä aikavaatimukset johtuvat (tarkkoja matemaattisia todistuksia ei tarvitse esittää).

4. (6 p) Tarkastellaan *rahanvaihto-ongelmaa*: Olkoon annettu rahasumma n ja m erilaista kolikon arvoa $\{d_1, d_2, \ldots, d_m\}$. Kunkin arvoisia kolikoita on käytettävissä rajaton määrä. Rahasumma n halutaan muodostaa kolikoiden avulla niin, että tarvittavien kolikoiden määrä on mahdollisimman pieni. (Määrää tarkastellessa vain kolikoiden yhteismäärä merkitsee. Kolikoiden arvoilla ei ole tässä merkitystä.)

   Laadi dynaamista ohjelmointia käyttävä algoritmi, joka selvittää, miten käytettävissä olevista kolikoista saadaan muodostettua haluttu rahasumma niin, että tarvittavien kolikoiden määrä on mahdollisimman pieni. Älä kirjoita algoritmiasi pseudokoodina, vaan esitä laskennassa käytettävät lausekkeet, kerro kuinka ja missä järjestyksessä niiden arvot lasketaan ja miten voidaan lopulta päätellä, montako kappaletta kutakin kolikkoa tarvitaan (tai että rahasumman muodostaminen käytettävissä olevista kolikoista ei ole lainkaan mahdollista).

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

   Vinkki: tarkista, että ratkaisusi toimii oikein esimerkiksi silloin, kun $n = 110$ ja käytettävissä olevat kolikoiden arvot ovat $\{20, 50\}$.

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.

Muista vastata kurssin palautekyselyyn! Linkki kyselyyn on kurssin Noppa-sivulla.