Tik–106.4100 Algoritmien suunnittelu ja analyysi, syksy 2012 Tentti, 6.3.2013

Kirjoita jokaisen palauttamasi paperin yläreunaan selvästi "T-106.4100 Algoritmien suunnittelu ja analyysi, 6.3.2013", 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. $5n^4 + 12n + 5 \in O(n^3)$
   ii. $n + 3n \log n + 8n \in \Omega(n^2)$
   iii. $8n^8 + 5 \in \Theta(n^2)$

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

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

$$T(n) = \begin{cases} 1, & \text{kun } n = 1 \\ 8T(n/4) + 2n & \text{kun } n > 1 \end{cases}$$

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

   c) (2p) 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 *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.) Tarkastellaan suunnattua verkkoa $G = (V, E)$. Kirjoita psedokoodi algoritmille, joka tutkii, sisältääkö verkko G syklin. Algoritmin aikavaatimuksen tulee olla $O(|V| + |E|)$. Algoritmin ei tarvitse tulostaa sykliin kuuluvia solmuja. Riittää, että algoritmi kertoo, onko verkossa vähintään yksi sykli. Vinkki: vain takautuvat kaaret voivat aiheuttaa verkkoon syklejä.