58131 Tietorakenteet ja algoritmit (kevät 2015)

Kurssikoe 1 (2.3.2015)

Tentissä saa olla mukana käsin kirjoitettu yksi A4-kokoinen "lunttilappu", jonka molemmilla puolilla saa olla tekstiä.

Vastaa kuhunkin tehtävään **erilliselle** konseptipaperille. Kirjoita jokaisen paperin yläkulmaan **kurssin nimi, kokeen päivämäärä, nimi, nimikirjoitus ja opiskelijanumero**. Vaikka jättäisit johonkin tehtävään vastaamatta, tulee kysymyksen vastauspaperi siinäkin tapauksessa palauttaa.

Tehtävissä, joissa pyydetään algoritmia, voit käyttää luentojen (Cormenin) tyyppistä pseudokoodia tai muita ymmärrettäviä pseudokoodityylejä, tai oikeaa ohjelmointikieltä, esim. Javaa. Jos käytät oikeaa ohjelmointikieltä, selitä erityisen hyvin, mitä ohjelmassasi tapahtuu, äläkä käytä mitään kielen erikoista piirrettä tai valmiita kirjastoja.

1. [*4 pistettä*]

   (a) Pitääkö paikkansa, että $5 \log_2 n = \mathcal{O}(n^3)$?

   (b) Pitääkö paikkansa, että $5 \cdot 2^n = \mathcal{O}(n^3)$?

   Kummassakin kohdassa perustele vastauksesi täsmällisesti $\mathcal{O}$-merkinnän määritelmästä lähtien.

   Tehtävän pitäisi olla ratkaistavissa käyttämällä määritelmien ja potenssien peruslaskusääntöjen lisäksi esim. epäyhtälöä $2^n > n$ kaikilla $n \in \mathbb{N}$. Voit halutessasi käyttää muitakin logaritmi- ja eksponenttifunktion perusominaisuuksia, kunhan ilmoitat selvästi, mitä ominaisuuksia käytät.

2. [*6 pistettä*] Pikajärjestäminen. Selitä miten pikajärjestäminen toimii ja mikä on sen aikavaativuus pahimmassa tapauksessa ja keskimäärin. Sopiva pituus tekstille on korkeintaan 2 konseptisivua. Mitään todistuksia ei tarvitse esittää.

Käännä!

3. [*6 pistettä*] Halutaan liittää binäärihakupuun jokaiseen solmuun x kokonaisluku $x.\,size$, joka on solmun x jälkeläisten (eli seuraajien) lukumäärä.

   (a) Esitä algoritmi, jolla *size*-arvot saadaan lasketuksi, kun annetussa binäärihakupuussa nämä kentät ovat aluksi tyhjiä. (Muista, että solmu itse lasketaan omaksi jälkeläisekseen.)

   (b) Oletetaan, että edellisen kohdan mukaiset *size*-arvot ovat nyt solmuissa. Esitä tehokas algoritmi, joka annetulla avaimella k palauttaa puussa olevien avainta k pienempien avainten lukumäärän. Voit olettaa, että mikään avain ei esinny puussa useammin kuin kerran. Algoritmin aikavaativuuden tulee olla $\mathcal{O}(h)$, missä h on puun korkeus.

   (c) Kuvaa lyhyesti, miten binäärihakupuun **insert**- ja **delete**-operaatioihin voitaisiin lisätä *size*-arvojen päivittäminen. Aikavaativuuden tulee pysyä suuruusluokassa $\mathcal{O}(h)$.

4. [*6 pistettä*] Syötteenä annetaan jono, jossa on n kokonaislukua $x_1, \ldots, x_n$. Tehtävänä on poistaa jonosta *toistuvat alkiot* (duplikaatit) eli tilanteet, joissa $x_i = x_j$ joillain $i \neq j$. Toisin sanoen halutaan tulostaa jono $y_1, \ldots, y_m$, jossa esiintyy tarkalleen samat luvut kuin jonossa $x_1, \ldots, x_n$, mutta jokainen luku esiintyy tasan kerran. Jono $y_1, \ldots, y_m$ saa olla missä tahansa järjestyksessä.

   Syötteenä annetut luvut x_i eivät ole jonossa missään erityisessä järjestyksessä, ja niiden lukumäärä n on n. viisi miljoonaa. Kirjoitettavan ohjelman on tarkoitus toimia normaalissa pöytätietokoneessa. Esitä periaatteellisella tasolla tehtävälle tehokas ratkaisumenetelmä, kun

   (a) luvut ovat kokonaislukuja väliltä $0 \leq x_i \leq 10\,000$, mutta muuten ne voivat olla mitä tahansa

   (b) luvut ovat kokonaislukuja väliltä $0 \leq x_i \leq 10^{12}$, mutta muuten ne voivat olla mitä tahansa

   (c) luvut ovat kokonaislukuja väliltä $0 \leq x_i \leq 10^{12}$ ja niiden lisäksi oletetaan jakautuneen tälle välille tasaisesti.

   Menetelmän valinnassa kiinnitä kohdissa (a) ja (b) huomio pahimpaan tapaukseen ja kohdassa (c) keskimääräiseen tapaukseen. Perustele valintasi.

   Yksityiskohtaisia pseudokoodeja ei tarvitse esittää; kuvaa vain ratkaisujesi keskeiset tietorakenteet ja niiden käyttötapa. Voit olettaa tunnetuiksi kaikki kurssilla esitetyt tietorakenteet ja algoritmit.