Itä-Suomen yliopisto — Tietorakenteet ja algoritmit 1

Tietojenkäsittelytieteen laitos — Uusintatentti 19.11.2010

Tehtävissä joissa pyydetään kirjoittamaan pseudokoodi, kelpaa myös esim. Java tai C.

---

1. [Yhteensä **24p**] Vastaa lyhyesti seuraaviin kysymyksiin:

   (a) Anna käsitteen "iso-O" täsmällinen määritelmä.

   (b) Luettele pinoon ja jonoon liittyvät operaatiot ja näiden aikavaativuudet, kun ne on toteutettu linkitettyinä rakenteina. Mitä hyötyä / haittaa linkityksestä on verrattuna taulukkototeutukseen?

   (c) Aluksi tyhjään AVL-puuhun lisätään avaimet 5, 3, 4, ja 6, tässä järjestyksessä. Lopuksi poistetaan avain 3. Piirrä välivaiheet.

   (d) Halutaan hajauttaa n avainta käyttämällä jakojäännösmenetelmää ja ylivuotoketjuja. Miten valitset hajautustaululle sopivan koon m?

   (e) Aluksi tyhjään maksimikekoon lisätään avaimet 1, 2, 3 ja 4, tässä järjestyksessä. Lopuksi poistetaan maksimi. Piirrä välivaiheet.

   (f) Kurssilla käsiteltiin (1) lisäysjärjestäminen (insertion sort), (2) kekojärjestäminen (heapsort), (3) lomitusjärjestäminen (mergesort) ja (4) pikajärjestäminen (quicksort). Mainitse jokaiselle algoritmille vähintään yksi hyvä puoli, jonka suhteen se eroaa edukseen kolmesta muusta.

2. [Yhteensä **12p**] Halutaan toteuttaa linkitettynä listana tietotyyppi S, johon kohdistuu kolmenlaisia operaatioita:

   - Search(S, k): avaimen k haku joukosta S;
   - Insert(S, x): alkion x lisäys joukkoon S;
   - Split(S, k, P, Q): Luodaan uudet joukot P ja Q. Joukkoon P tulee kaikki ne joukon S alkiot, joiden avain on korkeintaan k. Loput alkiot tulevat joukkoon Q. Operaatiossa joukko S tulee tyhjäksi.

   Aiotusta sovelluksesta tiedetään, että yleisimpiä operaatioita ovat alkion lisääminen ja joukossa olevan avaimen hakeminen; Split on harvinaisempi.

   (a) Minkä muunnelman linkitetystä listasta valitsisit? Perustele lyhyesti.

   (b) Esitä Split-operaation toteutus pseudokoodina.

   (c) Olisiko tietotyyppi mielestäsi kuitenkin parempi toteuttaa AVL-puilla? Entä hajautuksella? Esittele lyhyesti näiden ratkaisutapojen hyviä ja huonoja puolia. Älä mieti Split-operaation yksityiskohtaista toteutusta, vaan käsittele asiaa yleisellä tasolla; voit myös vedota näiden tietorakenteiden tunnettuihin ominaisuuksiin kuvaamatta niitä tarkasti.

3. [Yhteensä **12p**] Olkoon annettu taulukko A jonka koko on n. Taulukko sisältää positiivisia kokonaislukuja satunnaisessa järjestyksessä.

   (a) Kirjoita algoritmi, joka palauttaa jonkin sellaisen luvun joka *ei* esiinny taulukossa A. Analysoi algoritmisi aikavaativuus.

   (b) Olkoon annettu lisäksi luku x. Kirjoita algoritmi joka tutkii päteekö $x = A[i] + A[j]$ joillekin i ja j. Saat käyttää apuna mitä tahansa kurssilla käsiteltyjä algoritmeja ja tietorakenteita kirjoittamatta niiden pseudokoodia. Analysoi algoritmisi aikavaativuus. Algoritmisi tehokkuus vaikuttaa arvosteluun merkittävästi.

4. [Yhteensä **12p**] Halutaan liittää AVL-puun jokaiseen solmuun x kokonaisluku $x.size$, joka on solmun x jälkeläisten lukumäärä.

   (a) Kirjoita algoritmi (pseudokoodi), jolla *size*-arvot saadaan lasketuksi, kun annetussa AVL-puussa nämä kentät ovat aluksi tyhjiä. Analysoi algoritmisi aikavaativuus.

   (b) Oletetaan, että edellisen kohdan mukaiset *size*-arvot ovat nyt solmuissa. Kirjoita tehokas algoritmi (pseudokoodi), joka kertoo kuinka mones jokin annettu solmu x on sisäjärjestyksessä (x on siis viite / osoitin johonkin puun solmuun). Analysoi algoritmisi aikavaativuus.

   (c) Kuvaa lyhyesti (pseudokoodia ei tarvita), miten *size*-arvojen ylläpitäminen voitaisiin ottaa huomioon AVL-puun lisäys- ja poisto-operaatioissa, siten että näiden operaatioiden aikavaativuus ei muutu.

   Arvosteluun vaikuttaa paitsi algoritmin toimivuus, myös tehokkuus.