

TIE-20100 Tietorakenteet ja algoritmit

Tentti 7.9.2015

Terhi Kilamo

Tentissä ei saa käyttää ylimääräistä kirjallista materiaalia, laskimia, tietokoneita tai muita lunttausvälineitä.

Muista vastata kaikkiin tehtäviin.

Kirjoita vastauksesi siistillä käsialalla lyhyesti - vastauksia ei arvostella viivoittimella. Vääristä vastauksista ei yleisesti vähennetä pisteitä, mutta tentin tarkastaja pidättää itsellään mahdollisuuden antaa miinuspisteitä täysin järjettömistä tai sisäisesti ristiriitaisista vastauksista (siis selvistä arvauksista).

1. a) Selitä lyhyesti käsiteett asymptoottinen analyysi ja amortisoitu ajoaika sekä miten ne liittyvät toisiinsa. (3p)
- b) Kurssilla käytiin läpi algoritmien suunnitteluperiaatteita. Yksi näistä oli *hajoita ja hallitse*. Kuvaile suunnitteluperiaate. Selitä lisäksi seuraavaa taulukkoa apunasi käyttäen, miten joku hajoita ja hallitse -periaatteeseen perustuva algoritmi järjestää taulukon.

11	5	4 ₁	3 ₁	4 ₂	9	10	3 ₂
----	---	----------------	----------------	----------------	---	----	----------------

(3p)

(6 p)

2. a) Määrittele binääripuu, binäärihakupuu ja tasapainotettu binäärihakupuu. Perustele, mitä etua tasapainoituksesta on ja anna esimerkki, miten se voidaan saavuttaa. (3 p)
- b) Millainen tietorakenne on hajautustaulu? (3 p)

3. a) Kelju K. Kojotti on kasaamassa uutta maantiekiihtäjänsaansa, kun Kelju tipauttaa ansassa tarvitut pultit ja mutterit sokin maahan. Miten epäonnen riivaama kajoottiparka saa pultit ja mutterit löytämään parinsa mahdollisimman vikkellästi. Kelju voi kokeilla, sopiiko pultti ja mutteri yhteen eli onko mutteri suurempi kuin pultti, pienempi vaiko sopiva. Kahta pulttia ja kahta mutteria ei voi vertailla keskenään. Mikä on ratkaisusi tehokkuus kertaluokkamerkinöillä esitettynä? (3 p)
- b) Yksi tiedonpakkauksen menetelmä on ns. Huffmanin koodaus, jossa merkkeille luodaan puumallilla bittiesitykset. Mihin suunnitteluperiaatteeseen oheinen algoritmi näyttäisi perustuvan?

HUFFMAN ALGORITMI

Askel 1 Alusta jokaiselle merkistön merkille oma yhden solmun puunsa ja merkitse kyseisen merkin esiintymistiheys merkin puun juureen. Se kertoo puun **painon**.

Askel 2 Toista seuraavaa kunnes yksi eheä puu on saatu rakennettua. Etsi kaksi puuta, joiden painot ovat pienimmät. Tee valitsemistasi puista uuden puun oikea ja vasen alipuu ja merkitse niiden painojen sunna uuden puun juureen sen painoksi.

Rakenna Huffmanin koodaus alla olevalle datalle:

merkki	A	B	C	D	—
todennäköisyys	0.4	0.1	0.2	0.15	0.15

ja koodaa merkkijono ABACABAD sen perusteella. (3 p)

(6 p)

4. Pitävätkö seuraavat väittämät paikkansa? (0.5 p/kohta)

- Syötteen alkioit kannattaa aina pitää järjestyksessä tietorakenteessa.
- Alkioiden järjestäminen on mahdollista tehdä $\Theta(n)$ ajassa.
- Rengaspuskuri on tapa toteuttaa jono tehokkaasti.
- Algoritmin muistinkäyttöä voi analysoida asympotoottisesti.
- Jos algoritmin suoritus aika on kertaluokassa $O(n)$, se on varmasti myös kertaluokassa $O(n^2)$.
- Jos algoritmin suoritus aika on kertaluokassa $\Omega(n^2)$, se on varmasti myös kertaluokassa $\Omega(n)$.
- Jos algoritmin suoritus aika on kertaluokassa $\Theta(n \log n)$, se on varmasti myös kertaluokassa $\Omega(\log n)$.
- Jos algoritmin suoritus aika on kertaluokassa $\Theta(n)$, se on varmasti myös kertaluokassa $O(n \log n)$.
- Jos algoritmin suoritus aika on kertaluokassa $\Theta(n^2)$, se on varmasti myös kertaluokassa $O(n^2)$.
- Jos algoritmin suoritus aika on kertaluokassa $\Theta(n)$, se on varmasti myös kertaluokassa $\Omega(n)$.
- Jos algoritmin suoritus aika on kertaluokassa $\Omega(\log n)$, se on varmasti myös kertaluokassa $\Theta(\log n)$.
- Jos algoritmin suoritus aika on kertaluokassa $O(n \log n)$, se on varmasti myös kertaluokassa $\Theta(n \log n)$. (6 p)

Handwritten notes on the right margin, including a small grid diagram and some calculations.

Handwritten notes on the right margin, including a small grid diagram and some calculations.

Handwritten notes on the right margin, including a small grid diagram.

Handwritten notes on the right margin, including a small grid diagram and some calculations.

5. a) Olet toteuttamassa uutta labyrintin suunnittelujärjestelmää, jonka avulla voidaan testata haastelabyrinttien vaikeustasoa. Suunnittele tietorakenne, jolla voit esittää labyrintteja. Voit käyttää apunasi C++:n standardikirjastoa. Kuva olisi kiva. (3 p)
- b) Kuvaile hakualgoritmi, joka löytää lyhimmän reitin ulos labyrintista. Kuvaile myös, miten tilanne muuttuu, kun labyrintin vaikeustasoa kasvatetaan lisäämällä matkalle kulkua hidastavia haasteita kuten örkkejä ja taikoja. (3 p)

(6 p)