TIE-20100 Tietorakenteet ja algoritmit

Tentti 19.10.2015

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 käsitteet

- algoritmi
- keko
- hajoita ja hallitse

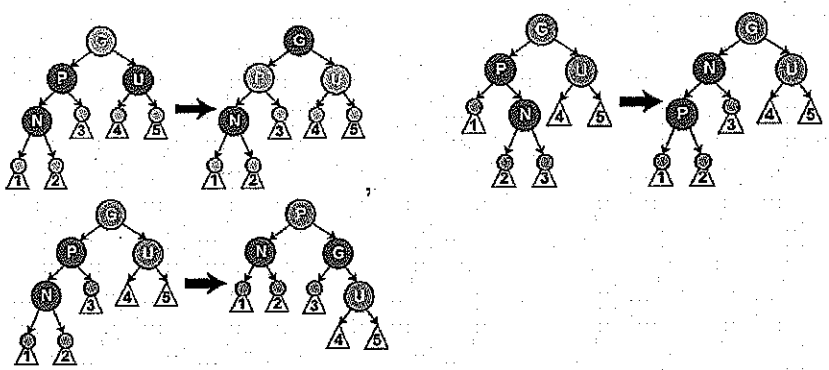
(3p)

b) Kurssilla käytettiin järjestämistä yhtenä algoritmiesimerkkinä. Mikä on tehokkain tapa järjestää miljoona 64-bittistä kokonaislukua, jotka on talletettu indeksoitavaan taulukkorakenteeseen? Perustele. Kuvaile lyhyesti valitsemasi algoritmin toimintaperiaate. Huom! Vastaukseksi on olemassa useita sopivia vaihtoehtoja, riittää antaa yksi perustelu. (3p)

(6 p)

2. a) Mikä on hajautustaulu? Mihin sitä käytetään ja miten tehokas se on? (3 p)

b) Puna-musta puu on yksi kurssilla käsitellyistä binäärihakupuista. Kolme vaihetta avaimen lisäämisestä puna-mustaan puuhun on kuvattu alla. Selitä, mitä kuvissa tehdään ja miksi esitetyt vaiheet ovat tarpeen. (3 p).



3. a) Millainen rakenne on graafi? Anna esimerkki ongelmasta, jossa graafirakenteesta on apua. Millaisella algoritmiratkaisulla ongelma ratkeaa? (3 p)

b) Mikä on alla olevan algoritmin tehokkuus? `[]()&#123;&#125;` on lambda-funktio eli nimetön funktio, jossa `[]` nappaa tietoa ympäristöstä, `()` antaa parametrit ja `&#123;&#125;` määritteleen funktion rungon. `find_if` kutsuu lambdaa jokaiselle alkiolle kunnes sopiva on löytynyt.

```
vector<int> data;
int val = 0;
while(cin >> val) {
    auto iter = find_if( data.begin(),
                         data.end(),
                         [val](int curr) { return ( val < curr ); });
    data.insert( iter, val );
    cout << ''Smallest: '' << *( data.begin() )
         << '' Largest: '' << *( data.end()-1 )
         << '' Latest: '' << val << endl;
}
cout << ''Mid value: '' << data[data.size()/2 - 1] << endl;
```

Voiko saman toiminnallisuuden tehdä tehokkaammin? Miten? (3 p)

(6 p)

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

a) Jos algoritmin suoritusaika on kertaluokassa $\Omega(n)$, se on varmasti myös kertaluokassa $\Omega(nlogn)$.

b) Jos algoritmin suoritusaika on kertaluokassa $O(n)$, se on varmasti myös kertaluokassa $O(nlogn)$.

c) Jos algoritmin suoritusaika on kertaluokassa $\Theta(nlogn)$, se on varmasti myös kertaluokassa $O(nlogn)$.

d) Jos algoritmin suoritusaika on kertaluokassa $\Theta(nlogn)$, se on varmasti myös kertaluokassa $O(n)$.

e) Jos algoritmin suoritusaika on kertaluokassa $\Theta(n)$, se on varmasti myös kertaluokassa $\Omega(n)$.

f) Jos algoritmin suoritusaika on kertaluokassa $\Theta(n^2)$, se on varmasti myös kertaluokassa $\Omega(n)$.

g) Jos algoritmin suoritusaika on kertaluokassa $\Omega(nlogn)$, se on varmasti myös kertaluokassa $\Theta(nlogn)$.

h) Jos algoritmin suoritusaika on kertaluokassa $O(n^2)$, se on varmasti myös kertaluokassa $\Theta(n^2)$.

i) Alkioiden järjestäminen vertailemalla on $\Omega(nlogn)$.

j) Kaikki jono-operaatiot rengaspuskurina toteutettuna ovat $O(1)$.

k) Alkion haku binäärihakupuusta on $O(lgn)$.

l) Asymptoottinen analyysi antaa hyvän kuvan algoritmin suorituskyvystä kaikilla syötteillä.

(6 p)

5.

1. Joulupukki ylläpitää listaa kilteistä ja tuhmista lapsista. Joulupukkiin uskovia alkaa vaan olla maailmanlaajuisesti niin paljon, että Pukilta alkaa mennä liikaa aikaa vanhanaikaisien listojensa kanssa. Listoissa viimeisimpänä tuhmien listalle joutunut pitää voida sujuvimmin siirtää takaisin kiltteihin. Kuitenkin kaikki voivat tehdä parannuksen eli ei saa estää paluuta tuhmista kiltteihin. Joulupukin tulee myös voida tulostaa kilttien listalla olijat lahjatoiveineen pakatakseen lahjareki ajallaan ja jakaakseen lahjat. Lahjojen järjestyksellä tai aattoyön lentoreitillä ei ole merkitystä, Joulupukki hoitaa sen puolen.

   Kuvaile tehokas toteutus Joulupukin uudelle Kiltit ja tuhmat -lista-appikselle, jonka tehtävä on helpottaa Joulupukin päivätyötä joulun alla. Käytä toteutuksessa C++:n standardikirjastoa. Pyri välttämään tiedon tallettamista moneen kertaa. Kuva ois kiva. (6 p)