

OHJ-2010 Tietorakenteiden käyttö

Tentti 05.02.2013

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.

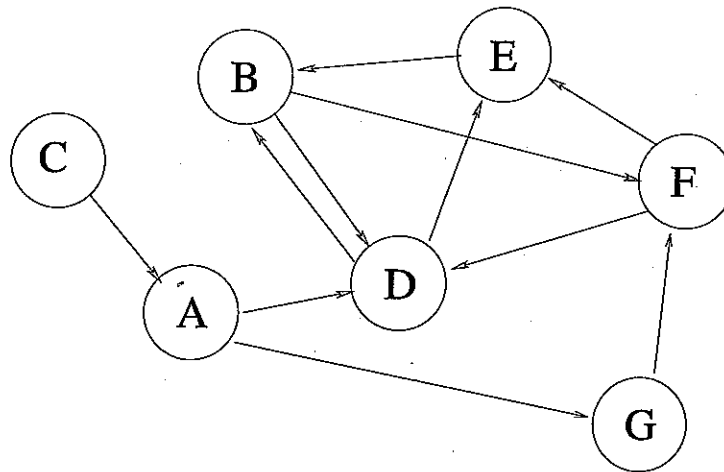
Vääristä vastauksista ei yleisesti vähennetä pisteitä, mutta tentin tarkastaja pidättää itsellään mahdollisuuden antaa miinuspisteitä täysin järjettömistä tai sisäisesti ristiriitaisista vastauksista (siis selvistä arvauksista).

1. a) Selitä lyhyesti (max.3 riviä/kohta) seuraavat käsitteet.
 - i. algoritmi (*algorithm*) (1 p)
 - ii. tietorakenne (*data structure*) (1 p)
 - iii. amortisoitu ajoaika (*amortized running time*) (1 p)
- b) Kurssilla käytiin läpi erilaisia algoritmien suunnitteluperiaatteita. Millainen periaate on hajoita ja hallitse (*divide and conquer*). Alla olevaa taulukkoa apuna käyttäen, kerro, miten joku kyseisen suunnitteluperiaatteen mukainen järjestelyalgoritmi toimii. (3 p)

10	5 ₁	2 ₁	5 ₂	2 ₂	9	1	3
----	----------------	----------------	----------------	----------------	---	---	---

2. Pitävätkö seuraavat väittämät paikkansa? (0.5 p/kohta)
 - a) Jos algoritmin suoritusaika on kertaluokassa $\Omega(\lg n)$, se on varmasti myös kertaluokassa $\Theta(\lg n)$.
 - b) Jos algoritmin suoritusaika on kertaluokassa $O(\lg n)$, se on varmasti myös kertaluokassa $\Theta(\lg n)$.
 - c) Jos algoritmin suoritusaika on kertaluokassa $\Theta(n \lg n)$, se on varmasti myös kertaluokassa $\Omega(n \lg n)$.
 - d) Jos algoritmin suoritusaika on kertaluokassa $\Theta(n \lg n)$, se on varmasti myös kertaluokassa $O(n \lg n)$.
 - e) Jos algoritmin suoritusaika on kertaluokassa $O(n)$, se on varmasti myös kertaluokassa $O(\lg n)$.
 - f) Jos algoritmin suoritusaika on kertaluokassa $\Omega(n)$, se on varmasti myös kertaluokassa $\Omega(\lg n)$.
 - g) Jos algoritmin suoritusaika on kertaluokassa $\Theta(n)$, se on varmasti myös kertaluokassa $O(n \lg n)$.
 - h) Jos algoritmin suoritusaika on kertaluokassa $\Theta(n)$, se on varmasti myös kertaluokassa $\Omega(n \lg n)$.
 - i) Binäärihakupuun korkeus on kertaluokassa $O(\lg n)$
 - j) Alkion etsiminen keosta avaimen perusteella on kertaluokassa $O(\log n)$
 - k) Taulukon alkiot voidaan aina järjestää ajassa $\Omega(n)$
 - l) Algoritmin suoritusaika voidaan aina esittää Θ -notaatiolla

3. a) Puna-musta puu on ns. tasapainotettu binäärihakupuun. Millainen rakenne se siis on eli mikä on binäärihakupuun ja miten puna-mustuus ilmenee? Millaisilla operaatioilla puun puna-mustuus pidetään voimassa? Anna esimerkki laillisesta puna-mustasta puusta. (4 p)
- b) Kerro missä järjestyksessä alla olevan graafin solmut väritetään leveyteen ensin -haussa, kun aloitussolmu on A ja solmun naapurisolmut käydään läpi aakkosjärjestyksessä. Kirjoita vastauksesi tyyliin: "P harmaaksi, Q harmaaksi, P mustaksi ...". (2 p)



4. a) Mikä on alla olevan algoritmin suoritusajaksi sopivilla kertaluokkamerkinnöillä ilmaistuna, kun sort on $O(n \lg n)$? Voisiko algoritmin ajankäyttöä tehostaa ja jos, niin mikä tehostetun version suoritusajan kertaluokka olisi? Huom! saat tehdä muutoksia tiedon tallennustapaan ja apunasi on standardikirjasto. (3 p)

```

int main() {
    vector<int> temp;
    int tmp = 0;

    while(cin >> tmp) {
        temp.push_back(tmp);
        sort(temp.begin(), temp.end());
        cout << "Lowest: " << *(temp.begin())
              << " Highest: " << *(temp.end()-1)
              << " Current: " << tmp << endl;
    }
    cout << temp.at(temp.size()/2 - 1) << endl;
}

```

- b) Olet menossa treffeille ja tarvitset asuusi mustat sukat puolisoskandidaatin täydelliseen hurmaamiseen. Olet eilen pessyt koneellisen mustia sukkia, jotka ovat nyt myyttynä kuivausrummussa. Kuinka kauan sinulla menee O ja Ω -merkinnöillä ilmaistuna löytää pari sukkamytystä. Huomaa, kaikille mytyn sukille ei välttämättä edes ole paria – ainakaan mytyssä – ja sukat eroavat toisistaan hiukan, jolloin mitkä tahansa kaksi mustaa eivät käy toistensa pariaksi. (3 p)
5. Olet toteuttamassa junailijan pikkuapuria eli palvelua, jolla voi hakea mahdollisimman nopeita tai mahdollisimman halpoja junayhteyksiä kahden eurooppalaisen kaupungin välille. Suunnittele C++:n standardikirjastoa apuna käyttäen tietorakenne yhteyksien tallettamiseksi. Kuvaile miten toteuttaisit yhteyksien haun? Pyri välttämään tiedon tallettamista useaan kertaan. Kuva ois kiva. (6 p)

