

OHJ-2010 Tietorakenteiden käyttö

Tentti 31.1.2008

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 miinus pisteitä täysin järjettömistä tai sisäisesti ristiriitaisista vastauksista (siis selvistä arvauksista).

1. Selitä lyhyesti (max.3 riviä/kohta) seuraavat käsitteet.

- binääripuu (*binary tree*) (1 p)
- binäärihakupuu (*binary search tree*) (1 p)
- keko (*heap*) (1 p)
- Järjestä alla oleva taulukko käyttäen annettua algoritmia. Kirjoita, mitä algoritmi tulostaa. Kirjoita myös alaindeksit osoittamaan samanarvoisten alkioden keskinäistä järjestystä. (3 p)

6 ₁	2	5	4	6 ₂	1	9	6 ₃
----------------	---	---	---	----------------	---	---	----------------

```

QUICKSORT(A, p, r)
  if p < r
    then q ← PARTITION(A, p, r)
       QUICKSORT(A, p, q - 1)
       QUICKSORT(A, q + 1, r)

```

```

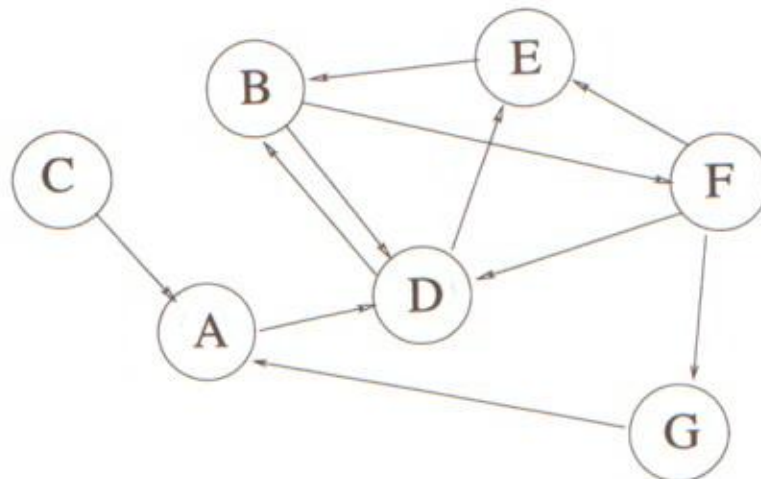
PARTITION(A, p, r)
  x ← A[r]
  i ← p - 1
  for j ← p to r - 1
    do if A[j] ≤ x
       then i ← i + 1
          exchange A[i] ↔ A[j]
  exchange A[i + 1] ↔ A[r]
  print A
  return i + 1

```

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

- Jos algoritmin suoritus aika on kertaluokassa $\Theta(\lg n)$, se on varmasti myös kertaluokassa $O(\lg n)$.
- Jos algoritmin suoritus aika on kertaluokassa $\Theta(\lg n)$, se on varmasti myös kertaluokassa $\Omega(\lg n)$.
- Jos algoritmin suoritus aika on kertaluokassa $O(n \lg n)$, se on varmasti myös kertaluokassa $\Theta(n \lg n)$.
- Jos algoritmin suoritus aika on kertaluokassa $\Omega(n \lg n)$, se on varmasti myös kertaluokassa $\Theta(n \lg n)$.
- Jos algoritmin suoritus aika on kertaluokassa $\Omega(n \lg n)$, se on varmasti myös kertaluokassa $\Omega(n)$.

- f) Jos algoritmin suoritusaika on kertaluokassa $O(n)$, se on varmasti myös kertaluokassa $O(n \lg n)$.
- g) Jos algoritmin suoritusaika on kertaluokassa $\Theta(n^2)$, se on varmasti myös kertaluokassa $\Omega(n \lg n)$.
- h) Jos algoritmin suoritusaika on kertaluokassa $\Theta(n^2)$, se on varmasti myös kertaluokassa $O(n \lg n)$.
- i) Hajautustaulun läpikäyminen on kertaluokassa $\Theta(n)$.
- j) Hajautustaulun läpikäyminen avainten arvojärjestyksessä on kertaluokassa $\Theta(n)$.
- k) Hajautustaulusta poistaminen on kertaluokassa $\Omega(1)$.
- l) Hajautustaulun koon tulisi olla samaa luokkaa kuin avainten arvoalueen.
3. a) Piirrä laillinen puna-musta binäärihakupuu, jossa on avaimet 2, 4, 4, 8, 11, 14, 14, 23, 26, 27, 28, 28, 31, 38 ja joka ei ole täydellisesti tasapainotettu (mutta luonnollisestikin tyydyttää puna-mustan puun invariantin). (2 p)
- b) Tyhjään merkkijonopuuhun (*trie*) lisätään sanat luu, luukku, luku, kuu, kulu, kukkuu, kuuluu ja kuulu. Kielen aakkosto sisältää merkit k, l ja u. Piirrä puu lisäysten jälkeen. (2 p)
- c) 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) Mitä Funktio tekee kun sille annetaan parametrina lista? Anna funktion ajan ja muistin kulutus O - ja Ω -merkinnöillä. Voisiko saman tehdä tehokkaammin? Perustele ja esitä tarvittaessa tehokkaampi ratkaisu. (4 p)

```

Alkio* Funktio(Alkio* lista) {
    return Funktio_r(Alkio* lista, 0);
}

Alkio* Funktio_r(Alkio* a, Alkio* e) {
    Alkio* tmp = a->seuraava;
    a->seuraava = e;
    if(tmp == 0) {
        return a;
    }
    else {
        return Funktio_r(tmp, a);
    }
}

```

- b) Pesun jälkeen käsissäsi on läjä tummia sukkiä, jotka sinun pitäisi parittaa. Käytät seuraavanlaista algoritmia. Asetellaan kaikki sukat riviin. Aloitetaan ensimmäisestä ja verrataan sitä järjestyksessä seuraaviin kunnes löytyy toinen samanlainen. Jos paria ei löydy, sukka heitetään parittomien pinoon. Oletetaan, että kahden sukan vertailuun menee vakioaika, samoin sukan asettamiseen riviin. Anna algoritmin suoritus aika O - ja Ω -merkinnöillä ilmaistuna kun ruutuja on n kappaletta. (2 p)
5. Tehtävänä on toteuttaa ohjelma, joka tarkastaa, onko annettu tehtäväjoukko mahdollista aikatauluttaa rajoitteet täyttävällä tavalla. Rajoitteet annetaan ohjelmalle $n \times n$ matriisina, jossa n on tehtävien määrä.

Esimerkki:

```

0 0 0 0 0
0 0 1 1 0
1 1 0 0 0
1 0 0 0 0
0 1 0 0 0

```

Kohdassa i j on ykkönen, mikäli tehtävä i täytyy suorittaa ennen tehtävää j .

Luettuaan syötteen ohjelma tulostaa sanan "kyllä", jos aikatauluttaminen on mahdollista ja sanan "ei", jos aikatauluttamista ei voida suorittaa. (Huomaa, että ohjelman ei tarvitse osata itse aikatauluttamista.) Kuvaile pääpiirteittäin millaisia algoritmeja ja tietorakenteita käyttäisit, kun STL on vapaasti käytettävissäsi. Perustele ratkaisusi. (6 p)