ITÄ-SUOMEN YLIOPISTO
Tietojenkäsittelytiede
Tietorakenteet ja algoritmit I (3621213) **kurssikuulustelu 31.10.2014**

1) a) **Vertaile** tasapainotettua binääripuuta ja hajautusta *joukon toteutusrakenteina*. Erityisesti käyttäjän (sovellusohjelmoijan) kannalta niiden hyvät ja huonot puolet. Huomioi perusoperaatiot, läpikäynnit, kahden joukon väliset operaatiot, operaatioiden tehokkuudet, jne. (3p).

b) **Vertaile** taulukkoa ja dynaamista linkitettyä listarakennetta *pakan toteutusrakenteina*. Erityisesti käyttäjän (sovellusohjelmoijan) kannalta niiden hyvät ja huonot puolet. Huomioi perusoperaatiot ja rakenteen tehokkuus jne. (2p).

2) a) Aseta seuraavat funktiot kasvunopeuden mukaiseen järjestykseen (ei tarvitse perustella). (4p)

n^3 n^2 $n\log n$ $\sqrt{n}$ n

2^n $n/\log n$ $\log n$ $\sqrt{n}/\log n$ $n^2\log n$

b) Mitkä seuraavista väittämistä pitävät paikkansa? Luettele numerot, ei tarvitse perustella. (3p)

1) $n\times(n+3) = O(n^2)$
2) $n\times(n-\sqrt{n})\times\log n = O(n^2)$
3) $n^2 + n\sqrt{n} + n = \Theta(n^2)$
4) $n^2+3n^2\log n = O(n^2)$
5) $\frac{1}{2}n + \log n = \Omega(n)$
6) $n\times\sqrt{n} / \log n = o(n^2)$

Seuraavissa "kirjoita algoritmi" -tehtävissä on tarkoitus käyttää Java-kielen tapaista pseudokoodia. Täsmälliseen Java:n syntaksiin ei toki tarvitse päästä käsin kirjoittaen, mutta käytä vain ohjelmointikielessä ja kirjastoissa (Java API ja TRA-kirjastomme) olevia rakenteita. Koska kokeessa testataan abstraktien tietotyyppien hallitsemista, käytä niitä oikein. Täsmällinen syntaksi, operaatioiden nimet tai parametrien järjestys ei ole olennainen, vaan operaatioiden käyttötapa. Jollet muista jonkin operaation nimeä, keksi oma nimi ja selitä mitä se tekee (mutta älä keksi uusia operaatioita ellet myös toteuta niitä). Pääohjelmia tai tietorakenteiden toteutuksia ei tarvitse sisällyttää vastaukseen.

3) Kirjoita algoritmi joka etsii *kahden listan erilliset alkiot*. Algoritmi saa siis parametrinaan listat *A* ja *B*, ja muodostaa sekä palauttaa uuden listan *C*, jossa on sellaiset *A*:n alkiot, joita ei ole *B*:ssä ja sellaiset *B*:n alkiot, joita ei ole *A*:ssa. Älä muuta syötelistoja *A* ja *B*. Voit käyttää listoina joko Java API:n listatoteutuksia (*LinkedList*, *ArrayList*, *Vector*) tai tietorakennekirjastomme asemaperusteista *TraLinkedList*:iä. Mikä on algoritmisi aikavaativuus? Aikavaativuus vaikuttaa arvosteluun. (6p)

4) Kirjoita algoritmi joka etsii binääripuun *sisäjärjestyksessä edellisen solmun*. Algoritmi saa siis parametrina solmun ja palauttaa sen edeltäjän sisäjärjestyksessä tai *null*, jos solmu oli sisäjärjestyksessä ensimmäinen. Aloitettaessa sisäjärjestyksessä viimeisestä solmusta ja toistuvasti kutsuttaessa puu tulisi läpikäytyä kokonaan. Mikä on algoritmisi aikavaativuus? Mikä on kokonaisaikavaativuus kun algoritmisi avulla läpikäydään puun solmut kokonaan läpi? (7p)