

# Tietorakenteet ja algoritmit I

## Kurssikuulustelu

2.11.2018

Kokeen kesto on kolme (3) tuntia.

Kokeessa ei saa käyttää apuna muistiinpanoja, muita materiaaleja tai laskinta.

1. Aseta seuraavat funktiot kasvunopeuden mukaiseen järjestykseen. Ei tarvitse perustella. (4 p)

$n$	$n \log n$	$(\log n)^2$	$n/4$
$n\sqrt{n}$	$n^3$	$2^n$	$n^2 + \log n$
$n^2/\log n$	$n^4$	$n/\log n$	$n^2 \log n$

2. **Vertaile lyhyesti** pikalajittelua, lomituslajittelua ja kasalajittelua. Erityisesti kunkin hyvät ja huonot puolet. (6 p)

Seuraavissa "kirjoita algoritmi" -tehtävissä on tarkoitus kirjoittaa täsmällistä Java:n tapaista algoritminotaatiota kuten olemme kurssilla käyttäneet. Täsmällistä Javan syntaksia tai operaatioiden nimiä ei arvostella, mutta annettuja/valittuja tietorakenteita on käytettävä oikein. Esimerkiksi binääripuulla ei ole lisäys- tai hakuoperaatiota, mutta puun solmulla on operaatio jolla voi asettaa vasemmaksi lapseksi jonkin toisen solmun. Jos olet epävarma jonkin operaation nimestä tai parametreista, käytä selkeää toimintaa kuvaavaa nimeä ja/tai erillistä selitystä. Alkiot ovat samat jos niiden `.equals()` -metodi antaa toden.

3. Kirjoita algoritmi joka etsii ja palauttaa kahdesta järjestetystä listasta ne alkiot jotka ovat yhdessä listassa, muttei toisessa listassa (XOR). Syötteenä siis listat  $A$  ja  $B$ , palautusarvona uusi lista  $U$ . Jos jokin alkio on yhdessä listoista  $A$  tai  $B$  yhden tai useamman kerran, mutta toisessa ei lainkaan, se tulee myös tulostamaan  $U$  yhtä monta kertaa kuin se oli siinä listassa jossa se esiintyi. Jos jokin alkio on molemmissa listoista  $A$  ja  $B$  yhden tai useamman kerran, se ei tule tulostamaan  $U$  lainkaan. Käytä listana valintasi mukaan joko `java.util.LinkedList`, `java.util.ArrayList`, tai `TraLinkedList` listaa. Älä muuta syötelistoja  $A$  tai  $B$ . Vihje: apurakenteita saa käyttää (mutta ei ole pakko). Mikä on algoritmisi aikavaativuus? Aikavaativuus vaikuttaa arvosteluun. (10 p)
4. Kirjoita algoritmi joka tarkastaa onko kahdessa sisäjärjestetyssä binääripuussa samat alkiot vai ei. Syötteenä siis kaksi binääripuuta  $A$  ja  $B$ , palautusarvona totuusarvo `true` jos puissa on täsmälleen samat alkiot, muuten `false`. Mikä on algoritmisi aikavaativuus? Aikavaativuus vaikuttaa arvosteluun. (10 p)