

## Tietorakenteet 2018

### Harjoitukset 9 (Viikko 45)

- **Huomioikaa, että yliopiston uuden tutkintosäännöksen mukaan kursista voi saada arvosanan hylätty rekisteriin.**

Mikäli opiskelija ei osallistu opetukseen eikä peru kurssipaikkaansa tai keskeyttää kurssin, hänen opintasuorituksensa arvioidaan arvosanalla hylätty.

- **Harjoitusryhmiin osallistuvien opiskelijoiden tulee olla paikalla ennen kuin harjoitusryhmä alkaa (klo 12.15/14.15/16.15). Myöhässä tulevat opiskelijat eivät saa rasteja tehdyistä tehtävistä.**
- **Katsokaa hyvissä ajoin ennen harjoitusryhmään tuloa ratkaistujen tehtävien numerot! Näin säästetään aikaa rastilistan täyttämisessä.**
- **Huomattavaa, että pseudokoodi ei tarkoita samaa kuin Java-koodi. Pseudokoodi on ohjelmointikielestä riippumaton esitys algoritmista.**

**Huom!** Tehtävissä 1.-3. käytettävissä tiedostoissa `Sort.java` on paikka muidenkin lajitteluiden toteutusta varten, mutta näistä ei tarvitse välittää. Samaa tiedostopohjaa tullaan käyttämään muillakin harjoituskerroilla.

1. Toteuta tiedostoon `Sort.java` (kansio Problem 1) lomituslajittelu (mergeSort). Lajittelu saa syötteekseen viitteen lajiteltavan listan ensimmäiseen solmuun. Käytä apuna annettuja metodeja
  - *split*, joka jakaa syötteenään saaman listan kahtia (1. osa on parametrina annetussa muuttujassa ja 2. osa paluuarvona eli `b = split(a)` jakaa listan a osiin a ja b, muuttujassa a oleva lista siis lyhenee) ja
  - *merge*, joka lomittaa kaksi jo järjestyksessä olevaa listaa (`c = merge(a, b)` tuottaa listan c, joka on järjestyksessä, jos a ja b ovat järjestyksessä, listoja a ja b ei pitäisi käyttää kutsun jälkeen).
- 2.-3. Toteuta tiedostoon `Sort.java` (kansio Problem 2-3) paikallaan toimiva lomituslajittelu (mergeSort). Lajittelufunktio saa syötteekseen lajiteltavan kokonaislukutaulukon.

4. Lajittelun sanotaan toimivan paikallaan, jos se lajittelee alkiot saman taulukon sisällä, missä ne olivat alunperinkin. Lisäksi käytetään korkeintaan vakiomäärä muuta tilaa (esim. apumuuttujat), joten alkioita ei esimerkiksi kopioida välillä toiseen taulukkoon.

Kuvaa pseudokoodilla algoritmi  $\text{HeapSort}(A)$ , joka järjestää  $n$ -alkioisen taulukon  $A$  alkiot nousevaan järjestykseen käyttäen **paikallaan toimivaa kekolajittelua**. Käytä hyväksesi maksimikekoa (keko, jossa arvot kasvavat kohti juurta, suurin alkio juuressa). Voit olettaa, että keon perusoperaatiot (esim.  $\text{BubbleDown}(A, i)$ / $\text{PercolateDown}(A, i)$ ) ovat jo valmiiksi käytettävissä.

5. Kuvaa miten (2,4)-puu muodostuu, kun siihen lisätään avaimet 1, 5, 9, 10, 2, 3, 7, 6, 4, 11, 12, 13, 8, 14, 15, 16.
6. Kuvaa miten edellisen tehtävän (2,4)-puusta poistetaan avaimet 5, 9, 16, 15, 12, 13, 14, 2, 3.
7. Kuvaa miten lomituslajittelu ja pikalajittelu lajittelevat sekvenssin 7, 9, 6, 4, 0, 1, 2, 5, 3, 8.