Tietorakenteet 2018
Harjoitukset 8 (Viikko 44)

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

  Mikäli opiskelija ei osallistu opetukseen eikä peru kurssipaikkaansa tai keskeyttää kurssin, hänen opintosuorituksensa 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.**
- **Huomatkaa, että pseudokoodi ei tarkoita samaa kuin Java-koodi. Pseudokoodi on ohjelmointikielestä riippumaton esitys algoritmista.**

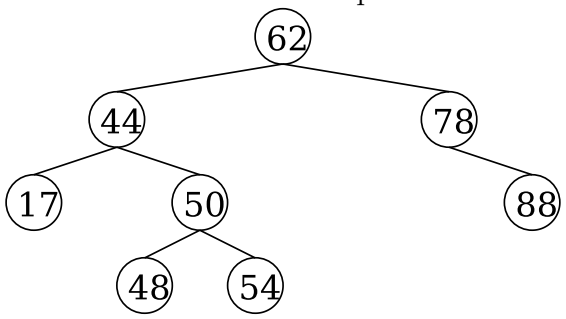
1.-3. Toteuta `insert`- ja `extractMin`-operaatiot kokonaislukuja sisältävälle keolle, joka on toteutettu taulukossa, tiedostoon `MinHeap.java`. Testaa kekoa `MinHeapTest` ohjelmalla (tiedostossa `MinHeapTest.java`); joudut muuttamaan testiohjelmaa hieman. `MinHeapTest` oletuksena visualisoi keon rakentumista, ja itse testi on kommentoitu pois.
Toteuta ensin operaatiot `percolateUp` ja `percolateDown`, jotka vievät alkion omalle paikalleen keossa joko ylöspäin tai alaspäin. `PercolateUp` toimii seuraavasti: avainta viedään keossa ylöspäin vaihtamalle se vanhempansa kanssa, kunnes vanhemmassa on pienempi avain tai avain tulee juureksi (monisteessa käytetään nimeä up-heap-bubbling). `PercolateDown` tekee saman, mutta vie avainta alaspäin kunnes molemmat lapsista ovat suurempia tai avain menee pohjalle (monisteessa down-heap-bubbling).
Käytä toteutuksissa apuna annettuja operaatioita `parent`, `leftChild` ja `rightChild`, jotka laskevat solmun vanhemman ja lasten indeksit solmun indeksistä ja operaatiota `swap`, joka vaihtaa kahden solmun sisällöt. Keko binääripuuna alkaa taulukon *table* indeksistä 1 ja *last* sisältää keon viimeisen alkion indeksin (esim. kun keon koko on 1 on siinä vain juuri ja $last = 1$).

4. Kuvaa miten AVL-puu muodostuu, kun alunperin tyhjään puuhun lisätään avaimet

   a) 1, 2, 3, 4, 5, 6, 7, 8

   b) 1, 8, 2, 7, 3, 6, 4, 5

5. Kuvaa miten kuvan AVL-puu muuttuu kun siitä poistetaan avain 62.



6. Miten toteuttaisit poisto- ja haku-operaatiot suljetussa hajautuksessa? Tutki tilanteita joissa tapahtuu törmäyksiä ja poistetaan avaimia, jotka ovat törmänneet. Muuttuuko myös lisäysoperaatio? Miksi poistoa ei voi tehdä "suoraviivaisesti" vain poistamalla avain lokerosta?

7. Käytetään 11-alkioista hajautustaulua, jonka hajautusfunktio on

$$h(i) = (3i + 2) \mod 11.$$

   Tauluun lisätään avaimet 3, 20, 100, 15, 9, 16, 45, 11, 5, 33. Millainen tilanne syntyy, kun törmäykset ratkaistaan

   a) ketjuttamalla?

   b) lineaarisella hajautuksella?

   c) neliöllisellä hajautuksella?