

## Tietorakenteet 2018

### Harjoitukset 3 (Viikko 39)

- **Huomioikaa, että yliopiston uuden tutkintosäännöksen mukaan kursista 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.**
- **Muistakaa ilmoittautua kurssille ja harjoitusryhmään Tietorakenteet kurssin kurssisivulla (<http://www.sis.uta.fi/~tira/>).**
- **Huomatkaa, että pseudokoodi ei tarkoita samaa kuin Java-koodi. Pseudokoodi on ohjelmointikielestä riippumaton esitys algoritmista.**

- 1.-2. Tässä tehtävässä tarvitset seuraavia java-tiedostoja: *Deque.java* (kaksiloppuinen jono), *DoubleLinkNode.java* ja *DequeTest.java* (kaksiloppuisen jonon testiohjelma). Toteuta javalla tiedostoon *Deque.java* operaatiot **insertLast**, joka lisää alkion jonon loppuun ja **removeLast**, joka poistaa ja palauttaa alkion jonon lopusta. Voit katsoa mallia toteutetuista operaatioista **insertFirst** ja **removeFirst**. Testaa toteutustasi testiohjelmalla poistamalla kommentit testiohjelman testien 2 ja 3 ympäriltä (kyseiset testit eivät toimi nykyisellä toteutuksella).
3. Toteuta edellisen tehtävän tiedoston *Deque.java* kaksiloppuiseen jonoon operaatio **Reverse**, joka kääntää jonon alkiot päinvastaiseen järjestykseen. Toteuta operaatio manipuloimalla jonon solmujen linkkejä (ei siis esim. siirtämällä alkioita toiseen tietorakenteeseen). Mikä on toteutuksesi aikakompleksisuus?

4. Alkeisoperaatioiden määrää laskematta selvitettävä seuraavien algoritmien aikakompleksisuus (iso- $\mathcal{O}$ ). Miksi kohdassa d ei voi päätellä, että  $\mathcal{O}(m) + \mathcal{O}(n^2) = \mathcal{O}(n^2)$ ?

- |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>a) <math>s \leftarrow 0</math><br/> <b>for</b> <math>i \leftarrow 1</math> <b>to</b> <math>n</math> <b>do</b><br/>           <math>s \leftarrow s + i</math><br/> <b>end for</b></p> <p>b) <math>s \leftarrow 0</math><br/> <b>for</b> <math>i \leftarrow 1</math> <b>to</b> <math>n</math> <b>do</b><br/>           <b>for</b> <math>j \leftarrow 1</math> <b>to</b> <math>i</math> <b>do</b><br/>               <math>s \leftarrow s + i + j</math><br/>           <b>end for</b><br/> <b>end for</b></p> | <p>c) <math>s \leftarrow 0</math><br/> <math>i \leftarrow n</math><br/> <b>while</b> <math>i \geq 1</math> <b>do</b><br/>           <math>s \leftarrow s + i</math><br/>           <math>i \leftarrow i - 1</math><br/> <b>end while</b></p> <p>d) <math>s \leftarrow 0</math><br/> <b>for</b> <math>i \leftarrow 1</math> <b>to</b> <math>m</math> <b>do</b><br/>           <math>s \leftarrow s + i</math><br/> <b>end for</b><br/> <b>for</b> <math>i \leftarrow 1</math> <b>to</b> <math>n</math> <b>do</b><br/>           <b>for</b> <math>j \leftarrow 1</math> <b>to</b> <math>n</math> <b>do</b><br/>               <math>s \leftarrow s + i + j</math><br/>           <b>end for</b><br/> <b>end for</b></p> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

5. Olkoon  $S$  pino ja  $Q$  jono. Kuvaa niiden tila seuraavien operaatiojonojen jälkeen, kun kunkin operaatiojonon alussa  $S$  ja  $Q$  ovat tyhjiä.

- a)  $S.push(a)$ ;  $S.push(b)$ ;  $S.push(c)$ ;  $S.pop()$ ;
- b)  $S.push(a)$ ;  $S.push(S.top())$ ;  $S.push(b)$ ;  $S.push(S.pop())$ ;  $S.push(S.top())$ ;
- c)  $Q.enqueue(a)$ ;  $Q.dequeue()$ ;  $Q.enqueue(b)$ ;  $Q.enqueue(c)$ ;
- d)  $S.push(a)$ ;  $Q.enqueue(b)$ ;  $Q.enqueue(S.top())$ ;  $S.push(Q.dequeue())$ ;  $S.pop()$ ;  
 $S.push(Q.front())$ ;

6. Miten toteuttaisit pinojen avulla jonon? Esitä pseudokoodilla jonon funktiot **Enqueue(x)** ja **Dequeue** käyttäen pinoja. Voit olettaa että pinon funktiot **Push(x)** ja **Pop** tunnetaan (ei tarvitse esittää niiden toteutuksia). Montako pinoa tarvitset?

7. Mitattaessa kohinaisen signaalin, kuten jonkin paikannusanturin ulostuloa, voidaan kohinan vaikutusta vaimentaa esim. soveltamalla liukuvaa keskiarvoa (moving average). Kuva pseudokoodilla **jonon** avulla algoritmi **MovingAverage**, joka laskee  $n$ :n peräkkäisen luvun keskiarvon mahdollisimman tehokkaasti. Esimerkiksi, kun  $n = 2$  (eli lasketaan kahden peräkkäisen luvun keskiarvo) ja herätteenä on signaali "1 1 2 3 3", saadaan tulos "0.5 1 1.5 2.5 3" kun oletetaan, että signaalin ensimmäistä arvoa on edeltäneet luvut ovat arvoltaan "0".
8. Kuva pseudokoodilla algoritmi, joka saa syöttekseen kaksi linkitetyillä listoilla kuvattua epänegatiivista kokonaislukua  $A$  ja  $B$  ja palauttaa listan, jossa on luku, joka saadaan laskemalla luvut yhteen. Luvut on kuvattu niin, että listan ensimmäinen alkio sisältää vähiten merkitsevän numeron. Esim. jos  $A = 986$ , joka esitettynä listana on  $(6, 8, 9)$  ja  $B = 103$  eli  $(3, 0, 1)$  niin tällöin tulos  $986 + 103 = 1089$  eli palautetaan lista  $(9, 8, 0, 1)$ .