

Tietorakenteet ja algoritmit I

29.11.2019

Kurssikuulustelun uusinta

Yleinen kuulustelu

Kurssikuulustelun uusinta kestää kolme (3) tuntia, yleinen kuulustelu kestää neljä (4) tuntia. Kuulustelussa ei saa käyttää laskinta, eikä mitään materiaaleja.

1. Aseta seuraavat funktiot kasvunopeuden mukaiseen järjestykseen. Ei tarvitse perustella. (5 p)

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

2. Anna seuraavien operaatioiden aikavaativuuksien kertaluokat kun n on kyseisessä kokoelmassa olevien alkioden määrä ja m parametrikokoelman alkioden määrä (niissä kohdissa joissa parametrina on kokoelma). Huomioi kunkin kohdan parametrin tyyppi. Kunkin muuttujan (viittauksen) *myXX* luokka on *XX*.

Vihje: kunkin operaation aikavaativuus on joko $O(1)$, $O(\log n)$, $O(n)$, $O(n \log n)$, $O(n \log m)$, $O(m \log n)$, $O(nm)$ tai $O(n^2)$. Aikavaativuuksia ei tarvinnut opetella ulkoa, vaan ne pystyy pääättelemään kunkin kokoelman toteutustavasta.

Arvostelu: oikea vastaus: 2 p, väärä vastaus: -1 p, tyhjä: 0 p. Koko tehtävän maksimi 12 p, minimi 0 p.

- (a) *myLinkedList.add(myLinkedList.size(), Object)*
- (b) *myArrayList.add(myArrayList.size(), Object)*
- (c) *myArrayList.set(myArrayList.size()/2, Object)*
- (d) *myLinkedList.removeFirst()*
- (e) *myHashMap.containsKey(Object)*
- (f) *myHashSet.addAll(TreeSet)*

Seuraavissa "kirjoita algoritmi" -tehtävissä on tarkoitus kirjoittaa täsmällistä Java:n tapaista algoritminotaatiota kuten olemme kurssilla käyttäneet. Täsmällinen syntaksi ei ole tärkeää, mutta käytettyjä tietorakenteita on käytettävä oikein. Esimerkiksi binääripuulla ei ole lisäys- tai hakuperiaatiota, mutta puun solmulla on operaatio jolla voi asettaa vasemman lapsen.

3. Kirjoita algoritmi joka poistaa annetusta listasta alkioden moninkertaiset esiintymät. Alkiot ovat samanlaiset jos niiden *equals()* -metodi palauttaa toden. Samanlaisista alkioista ensimmäinen säilytetään ja muut poistetaan. Jäljellejääneiden alkioden järjestys säilyy samana. Apurakenteita saa käyttää. Mikä on algoritmisi aikavaativuus? Aikavaativuus vaikuttaa arvosteluun. (11 p)
4. Kirjoita algoritmi joka tarkastaa onko annettu binääripuu sisäjärjestyksessä vai ei. Siis onko kussakin solmussa talletettu alkio siten, että alkiot ovat puussa sisäjärjestyksessä. Parametrina binääripuu, palautusarvona totuusarvo. Alkioilla on *compareTo()* -metodi keskinäisiä vertailuja varten. Mikä on algoritmisi aikavaativuus? Aikavaativuus vaikuttaa arvosteluun. (12 p).

Jos teet seuraavat tehtävät 5-7 tai(ja) käytät yli kolme tuntia, kurssi arvostellaan vain tämän kuulustelun (7 tehtävää) perusteella. Jos teit syksyn 2019 kurssilla X-tehtävät ja haluat huomioidavan X-tehtävien tuloksen ja muiden demotehtävien bonukset tämän kuulustelun (4 tehtävää) arvosteluun, älä tee seuraavia tehtäviä 5-7 ja palauta paperit kolmen tunnin kuluessa.

5. **Piirrä kuva** ja selitä hyvin lyhyesti (parilla sanalla) kukin seuraavista käsitteistä. Kuvalla on suurempi paino kuin selityksellä. Kuvan piirtoteknistä laatua ei arvostella, vain sitä miten hyvin kuva esittää asiaa. (4p)

(a) Mediaanin haku pikalajittelun jaottelualgoritmilla.

(b) Tasapainotettu hakupuu (ei kuitenkaan kuvaa kierto-operaatiosta).

6. **Vertaile lyhyesti** kuplajittelua, lomituslajittelua ja kantalukulajittelua. Erityisesti kunkin hyvät ja huonot puolet. (6p)

7. **Toteuta** abstrakti tietotyyppi **jono** Java -kielellä. Esittele siis tarvittava(t) luokka (luokat) ja toteuta konstruktori ja operaatiot *addLast*, *removeFirst*, ja *isEmpty*. Pääohjelmaa tai muita operaatioita ei tarvitse toteuttaa. JavaAPI:n tai tietorakennekirjastomme valmiita kokoelmaluokkia ei saa käyttää. Toteuta operaatiot vakioaikaisiksi siten, että kaikki muutkin jonon perusoperaatiot voidaan toteuttaa (keskimäärin) vakioaikaisiksi. (10p)