

Yleinen kuulustelu

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)

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

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 $O(1)$, $O(\log n)$, $O(n)$, $O(m)$, $O(n\log n)$, $O(n\log m)$, $O(m\log n)$, $O(nm)$ 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) *myArrayList.addAll(ArrayList)*
- (b) *myArrayList.add(myArrayList.size()/2, Object)*
- (c) *myLinkedList.add(myLinkedList.size()/2, Object)*
- (d) *myArrayList.removeFirst()*
- (e) *myTreeMap.containsKey(Object)*
- (f) *myTreeSet.addAll(HashSet)*

Seuraavissa "kirjoita algorimi" -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 hakuoperaatiota, mutta puun solmulla on operaatio jolla voi asettaa vasemman lapsen.

3. Kirjoita tehokas algoritmi joka toimii kuten vakiokirjastojen operaatio *retainAll(A, B)*. Algoritmi siis poistaa listasta *A* kaikki ne alkiot jotka eivät esiinny listassa *B*. Voit käyttää valintasi mukaan Java API:n linkitettyä listaa (*java.util.LinkedList*), taulukkopohjaista listaa (*java.util.ArrayList*), tavallista taulukkoa tai tietorakennekirjastomme asemaperustaista linkitettyä listaa (*TrLinkedList*). Saat käyttää apurakenteita, mutta et saa käyttää minkään kokoelman valmista *retainAll()* -operaatiota etkä muuttaa listaa/taulukkoa *B*. Kerro ja perustele algoritminisi aikavaativuus. Aikavaativuus vaikuttaa arvosteluun. (10 p)
4. Kirjoita algoritmi *inorderNext(BTreeNode n)* joka etsii ja palauttaa sen puusolmuun joka on puusolmun n seuraaja sisäjärjestyksessä binääripuussa. Jos solmulla n ei ole seuraajaa, algoritmi palauttaa *null*. Mikä on algoritminisi aikavaativuus? (10 p)
5. **Vertaile lyhyesti** lomitussajittelua, pikalajittelua ja kantalukulajittelua. Erityisesti kunkin hyvät ja huonot puolet. (6 p)
6. **Toteuta** abstrakti tietotyyppi *pakka* Java -kielellä. Esittele siis tarvittava(t) luokka (luokat) ja toteuta konstruktori sekä operaatiot *addLast*, *removeFirst*, ja *isEmpty*. Pääohjelmaa tai muita operaatioita ei tarvitse toteuttaa. JavaAPI:n tai tietorakennekirjastomme valmiita kokoelma- luokkia ei saa käyttää. Toteuta operaatiot vakioaikaisiksi siten, että kaikki muutkin pakan perusoperaatiot voidaan toteuttaa (keskimäärin) vakioaikaisiksi. (12 p)