Tietorakenteet ja algoritmit II

31.10. - 1.11.2024

Harjoitus 1

Tehtävät X1, 1 ja 3 ovat kertausta TRA I -kurssilta, ne voi tehdä jo ennen kurssin alkua (ja samalla tulee hyvää kertausta TRA I kurssikuulusteluun). Tehtävässä X1 tarvitset kurssin tietorakennekirjastoa jonka käyttö- ja asennusohjeet löytyvät Moodlesta.

Piirrä kuva kustakin tehtävästä ja suunnittele algoritmisi toiminta kuvan kanssa ennen toteuttamisen aloittamista. Kaikissa "kirjoita algoritmi joka" tehtävissä tehdään toimiva Java-metodi joka saa parametrinaan syötteen ja palauttaa tehtävänannon mukaisen palautusarvon, mutta ei tee mitään muuta. Ei siis esimerkiksi muuta syötteitään ellei syötteen muuttamista erikseen pyydetä. Lopullinen versio itse algoritmimetodista ei myöskään tulosta mitään. Ota syötteen generoiva ja metodia kutsuva pääohjelma Moodlesta. Koska algoritmitehtävät on tarkoitus tehdä toimiviksi asti, tuo toimivat ohjelmasi testattavaksi harjoitustilaisuuteen USB-tikulla, Moodleen tai muualle verkkoon tallennettuna tai omalla tietokoneella esitettäväksi. Karelian ja Savonian etäryhmä palauttaa kunkin tehtävän ratkaisunsa erikseen Moodleen to 07:30 mennessä.

Seuraava tehtävä X1 on pakollinen kaikille opiskelijoille. X-tehtävien ratkaisujen pitää olla kunkin opiskelijan **itse** tekemiä. Saman ratkaisun kopioita ei hyväksytä (versioitunakaan). Vastaukset pitää lähettää sunnuntaina 3.11. klo 20:00 mennessä allaolevaa ohjetta käyttäen. Saat automaattisen vastauksen pian onnistuneen lähetyksen jälkeen. Jollet saa kuittausta, on lähetyksessä mennyt jotain väärin. Jos kuittauksessa on kääntäjän virheilmoituksia, on tiedostossa tai sen lähetyksessä jotain väärin. Lähetä tällöin **korjattuna uudestaan**. Vastauksen on sisällettävä ohjelman kommenteissa lyhyt **itsearviointi** jossa arvioit ratkaisun toimivuutta, aikavaativuutta ja mahdollisia parannusmahdollisuuksia. Oikea itsearviointi (jonkinlaiseen ratkaisuun) on yhden pisteen arvoinen.

Lähetä ratkaisusi www-lomakkeella jonka osoitteen ja tunnukset saat erillisessä sähköpostissa kurssin alkaessa. Vastauksen tulee olla kääntyvä Java-tiedosto nimeltään `TRAII_24_X1_`*`tunnus`*`.java`, missä *`tunnus`* on alkuosa siitä sähköpostiosoitteesta joka sinulla Pepissä on ja johon X-tunnukset on lähetetty. Alkuosa päättyy ensimmäiseen pisteeseen (.), tavuviivaan (-) tai @-merkkiin. *`tunnus`* kaikki pienillä kirjaimilla. Esim. `sjuva@uef.fin` → `TRAII_24_X1_sjuva.java` ja `simo.juvaste@uef.fin` → `TRAII_24_X1_simo.java`. Jotta ohjelma kääntyisi, on myös luokan nimen oltava täsmälleen sama kuin tiedoston nimi, huomioiden isot ja pienet kirjaimet. Luokan on myös toteutettava rajapinta `TRAII_24_X1` jonka löydät Moodlesta. Rajapintamäärittelyä tai testi/pääohjelmaa ei pidä sisällyttää mukaan lähetykseen, ne ovat eri tiedostossa.

Ota runko vastaukseen Moodlesta äläkä muuta varsinaisen tehtävämetodin otsikkoa (nimeä, parametreja, tyyppejä). Älä myöskään tee luokastasi pakettia. Moodlessa on myös testiohjelma jota käyttämällä ja muokkaamalla voit testata oman ratkaisusi toimintaa.

X1. Kirjoita algoritmi joka palauttaa annetun binääripuun kaikki haarautumissolmut (siis muut solmut kuin lehtisolmut) joukkona. Algoritmi siis saa parametrinaan binääripuun ja palauttaa solmujen joukon. Mikä on algoritmisi aikavaativuus? Aikavaativuus vaikuttaa arvosteluun. Vihje: läpikäynti rekursiolla.

Käännä →

$\sqrt[3]{n}$, $\sqrt{n}\cdot\log n$, n, $13n$, $n^{1.5}$, $n^2/\log n$, $42n^2$, $n^2\log\log n$, $n^2\cdot\log n$

$(4/3)^n$, 2^n

1. Aseta seuraavat funktiot kasvunopeuden mukaiseen järjestykseen:

$n^2\log n$	n	$n^{1.5}$	2^n	$n^2\log\log n$	$42n^2$
$(4/3)^n$	$\sqrt[3]{n}$	$n^2/\log n$	$13n$	$\sqrt{n}-\log n$	$\sqrt{n}\,n^2$

2. Todista seuraavat väittämät määritelmään nojautuen todeksi tai epätodeksi (johda epäyhtälöt siten, että ylhäältä rajoittaminen tulee näytetyksi ja c ja n_0 selviävät):

   $n_0 = 1$

   (a) $n^2+n+1=O(n^2)$ $\quad n^2+n+1 \le n^2+n\cdot n+1\cdot n^2 = 3n^2$

   (b) $n+\sqrt{n}n+9=O(n^2)$ $\quad n+\sqrt{n}n+9 \le n\cdot n+\sqrt{n}n\cdot\sqrt{n}+9n^2 = 11n^2$

   (c) $n+\sqrt{n}n+1=O(n\sqrt{n})$ $\quad n+\sqrt{n}\cdot n+1 \le n\cdot\sqrt{n}+n\sqrt{n}+1\cdot n\sqrt{n} = 3n\sqrt{n}$

   (d) $n^3+4n-5=O(n^3)$ $\quad n^3+4n-5 \le n^3+4n \le n^3+4n\cdot n^2 = 5n^3$

3. Kirjoita algoritmi, joka saa syötteenään kokoelman (*Collection<E>* C) ja kokonaisluvun k ($k>0$), ja joka palauttaa tuloksenaan uuden listan niistä alkioista jotka esiintyivät kokoelmassa C **tasan** k kertaa. Kukin k kertaa esiintynyt alkio tulee tuloslistaan yhden kerran. Älä muuta syötettä. Alkiot ovat samat jos niiden *.equals*() -metodi palauttaa *true*. Käytä apuna kuvausta (*Map<E, Integer>*). Mikä on algoritmisi aikavaativuus? Ota pohjaa Moodlesta.

4. Vertaa tehtävän 3 toiminnan nopeutta kun apuvälinekuvauksena on (a) *HashMap* tai (b) *TreeMap*. Kirjoita ohjelma joka mittaa ja vertaa näiden nopeutta kun syöte kasvaa. Miten selität tulokset? Ota pohjaa ja esimerkkiä Moodlesta.