

Harjoitus 1

Tehtävät X1-3 ovat kertausta TRAI -kurssilta, ne kannattaa tehdä jo ennen kurssin alkua (ja samalla tulee hyvää kertausta TRAI kurssikuulusteluun niille jotka siihen ovat osallistumassa).

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 tarkoitettu tehdä toimiviksi asti, tuo toimivat ohjelmasi testattavaksi harjoitustilaisuuteen USB-tikulla, verkkoon tallennettuna tai omalla tietokoneella esitettäväksi.

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 keskiviikkona 30.10. klo 21:00 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 kommentteissa 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 olet saanut erillisessä sähköpostissa. Vastauksen tulee olla kääntyvä Java-ohjelma nimeltään `TRAI_X1_tunnus.java`, missä *tunnus* on alkuosa siitä sähköpostiosoitteesta jonka olet tälle kurssille Oodissa antanut. Alkuosa päättyy ensimmäiseen pisteeseen (.), tavuviivaan (-) tai @-merkkiin. *tunnus* kaikki pienillä kirjaimilla. Esim. `sjuva@uef.fi` → `TRAI_X1_sjuva.java` ja `simo.juvaste@uef.fi` → `TRAI_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 `TRAI_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. Moodlesta on myös testiohjelma jota käyttämällä ja muokkaamalla voit testata oman ratkaisusi toimintaa.

X1. Kirjoita algoritmi joka tarkistaa onko annettu binääripuu kasa vai ei. Kasa on binääripuu jossa jokaisen solmun sisältämä alkio edeltää (on ”pienempi”) tai on yhtäsuuri kuin minkään sen jälkeläisen sisältämä alkio. Jos binääripuussa on alkioita jotka eivät ole keskenään oikeassa järjestyksessä, algoritmi palauttaa epätoden, muuten toden. Parametrina binääripuu, palautusarvona totuusarvo. Mikä on algoritmisi aikavaativuus?

1. Lauantai-illan lottoarvonnassa tuloslaskenta ”kestä” pari minuuttia. Suunnittele, miten täysosumien (pienemmät voitot voit unohtaa) määrä voitaisiin laskea tietokoneella *mahdollisimman nopeasti* (siis *todella* nopeasti) oikean rivin selvittyä. Huomaa, että etukäteistyötä voidaan tehdä miten paljon tahansa (ts. kymmeniä minuutteja), eli pelatut rivit voidaan tallettaa minkälaiseen muotoon tahansa tietokoneen muistiin. Mieti erityisesti tätä talletusrakennetta. Lottoruudukossa on 40 numeroa, joista arvotaan 7. Lisänumeroita tai plusnumeroita ei täysosumassa ole, järjestelmät voit olettaa tallennetun useana tavallisena rivinä. Pelattuja

rivejä on viikoittain noin 15 miljoonaa [<https://www.veikkaus.fi/fi/lotto#!/ohjeet/saannot/loton-saannot>].

2. Kertaa yksiköt s, ms, μ s, ns, MHz, GHz. Arvioi edellisen tehtävän voitonhakualgoritmin kesto sekunneissa (tai sekunnin osissa) kun ajatellaan, että moderni tietokone tekee yhden keskusmuistia käyttävän operaation noin 50 ns:ssa, ja muistia käyttämättömän noin 0,5 ns:ssa.

3. Aseta seuraavat funktiot kasvunopeuden mukaiseen järjestykseen:

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

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

(a) $n^2 + 2n - 1 = O(n^2)$

(b) $n + n\sqrt{n} + 9 = O(n^2)$

(c) $n + 2n\sqrt{n} + 3 = O(n\sqrt{n})$

(d) $n^3 + 2n^2 - 2n = O(n^3)$