

Harjoitus 4

- 1) Kirjoita algoritmi joka tulostaa annetusta listasta kaikki vähintään k:n alkion mittaiset kasvavat osajaksot. Jos siis kukin k:sta alkioista on edeltäjänsä suurempi, tulostetaan ko. alkioita kunnes jokin alkio on pienempi (tai yhtä suuri) kuin edellinen tai lista loppuu. Vihje: kerää kasvavaa sarjaa jonoon.
- 2) Kirjoita algoritmi joka saa parametrinaan kaksi kokonaislukulistaa (kukin luku väliltä 0..9) ja joka laskee listojen summan, eli siis uuden listan jonka alkiot esittävät kokonaislukua, joka saadaan alkuperäisten listojen summasta. Huomaa ylivuodot (siirrot). Esim. listoista (5 2 3 4 5) ja (6 7 8 9 0) tulee lista (1 2 0 2 3 5). Huomaa, etteivät listat välttämättä ole saman mittaisia. Aikavaativuus? Voit käyttää joko TraLinkedList:iä tai java.util.LinkedList:ä.
- 3) Toteuta ADT jono listaoperaatioita käyttäen (vrt. monisteen kohdan 2.20 esimerkki) ja kirjoita pääohjelma, jolla testataan jonoa monipuolisesti.

Seuraava X2 tehtävä on pakollinen kaikille opiskelijoille.

X-tehtävien ratkaisujen pitää olla kunkin opiskelijan itse tekemiä. Saman ratkaisun kopioita ei hyväksytä(versioitunakaan). Vastaukset pitää lähettää edeltävänä päivänä keskiviikkona klo 14.00 mennessä sähköpostitse alla olevaa ohjetta käyttäen. Saat automaattisen vastauksen pian onnistuneen lähetyksen jälkeen. Vastauksen on sisällettävä lyhyt itsearviointi jossa arvioit ratkaisun toimivuutta, aikavaativuutta ja mahdollisia parannusmahdollisuuksia. Oikea itsearviointi (jonkinlaiseen ratkaisuun) on yhden pisteen arvoinen.

Lähetä ratkaisusi cs:n käyttäjälle mhk käyttäen viestin otsikkona merkkijonoa TRA1_X2_omakayttajatunnus missä omakayttajatunnus on sinun cs-käyttäjätunnuksesi. Helpointa lähettäminen on cs:ltä käyttäen komentoa:

```
/usr/ucb/mail -s TRA1_X2_omakayttajatunnus mhk < omakayttajatunnus.java
```

missä omakayttajatunnus on cs-käyttäjätunnuksesi ja omakayttajatunnus.java on ohjelmätiedosto joka sisältää vastauksesi. Jotta tehtävä kääntyisi, on pääohjelman luokan nimen oltava täsmälleen sama kuin käyttäjätunnuksesi (kuitenkin ilman @cs tms. osuutta). Ohjelman on siis oltava sähköpostin runkona sellaisenaan (ilman MIME/HTML -koodauksia, allekirjoituksia, uusia rivityksiä tms.). Ratkaisun pisteytys lasketaan suoraan mukaan kurssin pisteisiin.

X2)

Toteuteta tehtävän X1 poistaminen sekä TraLinkedList:lle, että java.util.LinkedList:lle. Kirjoita siis algoritmit (metodit) jotka saavat parametreinaan kaksi kasvavassa järjestyksessä olevaa kokonaislukulistaa (a) TraLinkedList:lle, (b) java.util.LinkedList ja jotka poistavat ensimmäisestä kaikki alkiot jotka esiintyvät toisessa. Jäljelle jäävät alkiot jäävät listaan edelleen kasvavaan järjestykseen. Älä käytä valmiita kokoelmien remove(element), removeAll(), tms. metodeja. Mitkä ovat algoritmiesi aikavaativuudet? Algoritmien aikavaativuudet (tehokkuudet) vaikuttavat arvosteluun hieman. Ota pääohjelman sisältävä pohja kurssin www-sivulta.