

## Harjoitus 2

## 1) Aseta seuraavat funktiot kasvunopeuden mukaiseen järjestykseen:

- |                       |                           |                    |                         |
|-----------------------|---------------------------|--------------------|-------------------------|
| a) $\sqrt{n}/\log(n)$ | f) $42 * \log\log(n)$     | k) $n^3 / \log(n)$ | p) $(\log(n))^2$        |
| b) $\sqrt{n} * n$     | g) $n^3 / \sqrt{n} + n^3$ | l) $2^n$           | q) $n!$                 |
| c) $n^{2.5}$          | h) $n^3$                  | m) $n * \log(n)$   | r) $\sqrt{n} + \log(n)$ |
| d) $n^{\log_2(7)}$    | i) 2009                   | n) $\log(n^2)$     |                         |
| e) $n$                | j) $(5/2)^n$              | o) $n^2 + n$       |                         |

valmistaudu perustelemaan peräkkäisten funktioiden järjestys.

## 2) Todista seuraavat väittämät määritelmään nojautuen (etsi c ja n0 ja todista oikeellisuus):

- a)  $3n^2 - 4n + 2\sqrt{n} = O(n^2)$   
 b)  $n(n-3) / 2 = \Omega(n^2)$   
 c)  $2n^2 - 2n + 2005 = \Theta(n^2)$   
 d)  $n^2 / 2 + n + \sqrt{n} = O(n^2)$

## 3) Määrä seuraavien ohjelmanosien suorituksen aikavaativuus kertaluokkana n:n suhteen.

- |                                                                                                                                              |                                                                                                                                                                                              |
|----------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| a) <pre>int sum = 0; for (int i = 0; i &lt; 3*n; i++)     sum = sum+1;</pre>                                                                 | b) <pre>int sum = 0; for (int i = 0; i &lt; n; i++)     for (int j = n; i &lt; j; j- -)         sum = sum + 1;</pre>                                                                         |
| c) <pre>int sum = 1; while (sum &lt; n)     sum = sum + sum;</pre>                                                                           | d) <pre>int sum = 0; for (int i = 4; i &lt; n-3; i++)     for (int j = -1; j &lt; n*2-5; j++)         sum = sum + 1;</pre>                                                                   |
| e) <pre>int j = 0; int i = n; while (j &lt; i) {     for (int k = 0; k &lt; i; k++)         sum = sum + 1;     i = i - 1; j = j + 1; }</pre> | f) <pre>int sum = 0; for (int j = 1; j &lt;= n; j++)     for (int i = 1; i &lt;= j; i++)         if (i == j)             for (int k = 0; k &lt; n; k++)                 sum = sum + 1;</pre> |

Harjoitustilaisuudessa ohjelmointi/algoritmitehtävät esitellään dataprojektorilla suoraan cs:ltä tai cc:ltä. Käännettävä ja suorittuva ohjelma on siis testattava keskuskoneella ja oltava sieltä harjoituksissa esitettävissä. Piirrä kuva kustakin tehtävästä. Käytä Java 1.5/6 kääntäjää ja virtuaalikonetta. cs:llä java5c tai java6c.

## 4) Kirjoita algoritmi (menetelmä) joka tutkii onko annetussa kokonaislukutaulukossa (Integer[]) jokin alkio kahteen kertaan. Mikä on algoritmisi aikavaativuus? Voiko sitä parantaa? Sovella X1-tehtävän pohjaa.

Seuraava X1 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 allaolevaa 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 TRAl\_Xl\_omakayttajatunnus missä omakayttajatunnus on sinun cs-käyttäjätunnukseksi (tai tunnus@cc). Helppointa lähettäminen on cs:ltä käyttäen komentoa:

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

missä omakayttajatunnus on cs-käyttäjätunnukseksi ja tunnus.java on ohjelmatiedosto joka sisältää vastauksesi. Jotta tehtävä kääntyisi, on pääohjelman luokan nimen oltava täsmälleen sama kuin käyttäjätunnukseksi (kuitenkin ilman @cc tms. osuutta). Ohjelman on siis oltava sähköpostin runkona sellaisenaan (ilman MIME/HTML -koodauksia, allekirjoituksia, uusia rivityksiä tms.). Tämä ensimmäinen X-tehtävä on samalla harjoitus X-tehtävien lähettämisestä. Ratkaisun pisteitys lasketaan suoraan mukaan kurssin pisteisiin.

X1) Kirjoita algoritmi (metodi) joka saa parametrinaan kaksi kasvavassa järjestyksessä olevaa kokonaislukutaulukkoa (Integer[]) ja joka poistaa ensimmäisestä kaikki alkiot jotka esiintyvät toisessa. Jäljelle jäävät alkiot jäävät taulukon alkuun tiiviisti edelleen kasvavaan järjestykseen. Taulukkoon tullut tyhjä tila täytetään null:eilla. Älä käytä valmiita kokoelmien remove(), removeAll(), tms. metodeja. Mikä on algoritmisi aikavaativuus? Algoritmin aikavaativuus (tehokkuus) vaikuttaa arvosteluun hieman.