

## Harjoitus 3

Ei X-tehtävää.

1. Määrä seuraavien aliohjelmien suorituksen aikavaativuus parametrin  $n$  suhteen. Aliohjelma  $Lin(n)$  on aikavaativuudeltaan  $O(n)$  ja  $Nel(n)$  on aikavaativuudeltaan  $O(n^2)$ .

```

a) void A(int n) {
    int a = 1, b = 1;
    a = 1;
    b = 1;
    while (a <= n) {
        if (b == n * n) {
            b = 1;
            a = a + 2;
        }
        else
            b = b + 1;
    }
}

b) void B(int n) {
    int a = 0;
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++) {
            Nel(j);
            a = a + 1;
        }
    for (int k = 0; k < a*a; k++)
        Lin(n);
}

c) void C(int n) {
    int a = 0;
    if (n > 1) {
        C(n-1);
        for (int i = 0; i < n; i++)
            a = a + 1;
    }
}

d) void D(int n) {
    int a, b;
    a = 1; b = 1;
    while (a <= n)
        if (b == n) {
            a = a + 1;
            Lin(a);
        } else
            b = b + 1;
}

```

Annettu tehtävä ("kirjoita algoritmi joka") tulee kirjoittaa omaksi menetelmäkseen (aliohjelmakseen). Algoritmin testaamiseksi pääohjelma tuottaa sopivat syötteen, kutsuu menetelmää ja tulostaa tuloksen. Valmis algoritmi itse ei siis lue tai tulosta mitään. Testausvaiheessa toki voi käyttää välitulostuksia. Ota pääohjelmien pohja joko omista aiemmista harjoituksista tai kurssin esimerkkisivulta. Havainnollista algoritmin toimintaa piirtämällä.

Tehtävissä 2 ja 3 käytetään tietorakennekirjaston TraLinkedList-kokoelmaa ja tehtävässä 6 tietorakennekirjaston LinkedList -kokoelmaa. Tehtävissä 4 ja 5 käytetään Java API:n valmista LinkedList -kokoelmaa. Ohjeet tietorakennekirjaston asentamisesta ja käytöstä [www-sivulla](http://www.sivulla).

- Kirjoita algoritmi joka kääntää listan (TraLinkedList) alkiot käänteiseen järjestykseen. Aikavaativuus?
- Kirjoita algoritmi joka etsii kahdesta listasta (TraLinkedList) samat alkiot ja luo uuden listan, jossa on vain ne alkiot jotka ovat vain jommassakummassa listassa, mutteivat toisessa (XOR). Jos jokin alkio esiintyy lähdelistoissa useasti, se kuitenkin lisätään kohdelistaan vain kerran. Aikavaativuus? Tilavaativuus?

4. Kirjoita algoritmi joka kääntää listan (LinkedList) alkiot käänteiseen järjestykseen. Aikavaativuus?
  
5. Kirjoita algoritmi joka etsii kahdesta listasta (LinkedList) samat alkiot ja luo uuden listan, jossa on vain ne alkiot jotka ovat vain jommassakummassa listassa, mutteivat toisessa (XOR). Jos jokin alkio esiintyy lähdelistoissa useasti, se kuitenkin lisätään kohdelistaan vain kerran. Aikavaativuus? Tilavaativuus?
  
6. Kirjoita algoritmi joka kääntää pinon k päällimmäistä alkioita, eli vaihtaa pinon k:n päällimmäisen alkion järjestyksen päinvastaiseksi. Jos pinossa on  $\leq k$  alkioita, algoritmi kääntää koko pinon. Aikavaativuus? Tilavaativuus?