

Tietorakenteet (syksy 2012)

Harjoitus 2

1. Määritellään kivojen maiden joukko seuraavasti:
 - Suomi on kiva maa.
 - Jos M on kiva maa ja N on maan M rajanaapuri, niin myös N on kiva maa.Osoita, että Italia on kiva maa.
2. Erään algoritmin suoritus vie 1 ms, kun syötteen koko on $n = 100$. Kuinka kauan suoritus kestää, kun syötteen koko on 10000 ja algoritmin aikavaativuus on suuruusluokkaa
 - (a) $O(\log n)$
 - (b) $O(n \log n)$
 - (c) $O(n^2)$
 - (d) $O(2^n)$?

Kussakin tapauksessa oletetaan, että alemman kertaluvun termit aikavaativuudessa voidaan jättää huomiotta; ts. esim. kohdassa (c) oletetaan, että aikavaativuus on cn^2 jollain vakiolla c .

3. Perustele määritelmään vedoten mitkä seuraavista väitteistä ovat tosia ja mitkä eivät?
 - (a) $2n^2 + 7n + 3 = O(n^3)$
 - (b) $2n^3 + 7n + 3 = O(n^2)$
 - (c) $\log n = O(10^6)$
 - (d) $10^6 = O(1)$
 - (e) $\log(n^2) = O(\log n)$
4. Joidenkin ongelmien ratkaisemiseen on olemassa erilaisia "ratkaisukaavoja" jotka nopeuttavat ongelman laskemista huomattavasti. Esimerkiksi biljoonan ensimmäisen luonnollisen luvun yhteenlaskeminen on kovin työlästä ja tylsää yksitellen. Onneksi yksi seuraavista ratkaisukaavoista pystyy suoriutumaan tästä ongelmasta leikiten. Osoita kaksi seuraavista kaavoista a)-c) vääräksi ja yksi oikeaksi peräkkäisten lukujen yhteenlaskemiseen. Todista kaavan oikeellisuus induktiolla tai todista että kaava ei toimi.
 - (a) $1 + 2 + 3 + \dots + n = 2 \cdot n - 1$
 - (b) $1 + 2 + 3 + \dots + n = \frac{n \cdot (n+1)}{2}$
 - (c) $1 + 2 + 3 + \dots + n = \frac{\log(n) \cdot n}{2 \cdot n}$
 - (d) Laske biljoonan ensimmäisen luonnollisen luvun summa. Voit käyttää merkintää biljoona = 10^{12}

Hyöty: Induktiolla voi todistaa mm. erilaisia ratkaisukaavoja, joiden avulla laskennallisesti vaativa tehtävä muuttuu lasten leikiksi. Kaavan oikeellisuus tulee kuitenkin todistaa ennen kuin sitä voi käyttää, eihän sitä muuten voi olla varma toimiiko se.

Mietittävä: Riittääkö mielestäsi muutamalla satunnaisella luvulla kokeileminen todistukseksi kaavan oikeellisuudesta.

5. (a) Seuraava algoritmi laskee summan $1 + 2 + 3 + \dots + n$:

```
summa = 0
for i = 1 to n
    summa = summa + i
```

Määritä algoritmin aika- ja tilavaativuus.

Mikä on for-lauseen invariantti? Perustele invariantin avulla, että algoritmi toimii oikein.

- (b) Summan $1+2+\dots+n$ voi laskea myös rekursiivisesti seuraavasti:

```
Summa(k)
    if k==0
        return 0
    else
        return summa(k-1) + k
```

Määritä algoritmin aikavaativuus ja tilavaativuus.

- (c) Voiko summan laskea niin, että aikavaativuus on vain $\mathcal{O}(1)$?

6. Mieti ja piirrä tässä tehtävässä, älä vain kopioi koodia ja katso mitä se tekee.

Tarkastellaan seuraavaa koodia:

```
//Luodaan 2-ulotteinen taulukko jonka jokaisella viidellä rivillä on numerot 0-4
int[][] taulukko = {{0,1,2,3,4},{0,1,2,3,4},{0,1,2,3,4},{0,1,2,3,4},{0,1,2,3,4}};

for (int i = 0; i < taulukko.length; i++) {
    for (int j = 0; j < taulukko[0].length; j++) {
        if (*)
            System.out.println(i + "," + j);
    }
}
```

Mitkä lukuparit tulostuvat jos merkin * kohdalla on

- (a) $i == 3 \ \&\& \ j == 3$
- (b) $i == 3 \ || \ j == 3$
- (c) $i != 3 \ \&\& \ j != 3$
- (d) $i != 3 \ || \ j != 3$
- (e) $!(i == 3 \ \&\& \ j == 3)$
- (f) $!(i == 3 \ || \ j == 3)$