

1.

Tällä viikolla opin miten Javassa voi mitata suoritukseen kuluvaan aikaa. Opin myös (taas), että linkitettyjen rakenteiden kanssa pitää olla todella tarkkana, varsinkin jos linkkejä on niinkin paljon kuin minulla on fibonaccikeossa (parent, lapsi, seuraava ja edellinen).

2.

Minua jäi vähän mietityttämään olenko tehnyt fibonaccikeon jotenkin huonosti. Sinänsä rakenne on ihan samanlainen kuin binomikeossa (käytän mm. samaa Solmu-luokkaa molemmissa), mutta järjestämisessä fibonaccikeko on aika surkea. Tulokset eräältä ohjelmani ajokerralta:

```
Kekojärjestämisen viemä aika eri keoilla
Binäärikeko: 17 ms
d-arykeko(4): 13 ms
Binomikeko: 20 ms
Fibonaccikeko: 192 ms
```

Ohjelmani siis järjestää jokaisen keon avulla saman 10 000-alkioisen int-taulukon jonka arvot on saatu randomilla. Voiko tämä tulos johtua pelkästään siitä, että järjestämisessä joudutaan niin usein suorittamaan deleteMin-operaatio, joka fibonaccikeolla oli vain tasoitettussa analyysissä luokkaa  $O(\log n)$ ?

Mietin myös, onko vertailua tehty tarpeeksi jos lopullinen ohjelma suoritettaessa antaa juuri tämäntyyppisen tulosteen, vai pitäisikö käyttäjän jotenkin päästä vaikuttamaan mitä kekoja vertaillaan ja miten? Tarvitaanko käyttöliittymää? Kuinka monella eri tavalla olisi hyvä vertailla kekoja, riittävätkö Dijkstran algoritmi jo toteutettu kekojärjestäminen?

3.

Sain keot valmiiksi, ja toteutin kekojärjestämisen jokaisella keolla. Tein main-luokkaan metodin, joka mittaa järjestämisessä kuluvaan aikaa eri kekoja käytettäessä. Aloitin Dijkstran algoritmin tekemistä.

4.

Seuraavaksi paneudun Dijkstran algoritmin toteutukseen, jotta voin vertailla miten keot suoriutuvat siitä. Myös dokumentaatiota pitää laajentaa.