

# VIKKORAPORTTI 2

Aineopintojen harjoitustyö: Tietorakenteet ja algoritmit

Helsingin yliopisto, Tietojenkäsittelytieteen laitos

**Tekijä:**

Tomi Virtanen

tomi.virtanen@helsinki.fi

014140105

## Tiralabra viikkoraportti 2

Koska palautukset ovat torstaisin, ajattelin jaksottaa palautettavia viikkoraporttien "viikot" kulkemaan torstaista torstaihin.

### Perjantai 14.12

Perjantaina aloin refaktoroimaan koodia lisää ja lisäämään kommentointia. Generoin myös olemassa olevat JavaDocit ja liitin dokumentaatiokansioon. Rakensin myös sokkelolle lisää testejä tarkastamaan sen validiutta. Koodilla itsellään on nyt DFS joka tarkistaa jokaisen sokkelon ja myös testeissä on DFS joka tarkastaa valmiin sokkelon läpäistävyyden.

### Lauantai 15.12

Pakkausrakenne ja luokkarakenne mietitty pidemmälle. Algoritmien luokat luotu omaan pakkaukseensa. Sokkelon ja paikkaluokan testit ja kommentoinnit ovat pitkälti valmiita. Seuraavaksi ajatuksena oli lähteä toteuttamaan algoritmeja yksi kerrallaan. Toteutin myös brute forcella toimivan huonon algoritmin vertailua varten.

Huomasin että olisi selvästi tarvetta suuremmille syötteille sillä 160x160 taulukon läpikäynti vie vain 16 millisekuntia. Eli sokkeloiden sisäinen läpäistävyydentutkinta on muutettava iteratiiviseksi. Tämä vaatii myös pinon käyttöä ja toteuttamista siis myöhemmin. Iteratiivisen DFS:n toteutus olikin aika triviaali ja nyt järjestelmä pystyy huomattavasti suurempiin syötteisiin. Aloitin myös javadocsien kirjoittamisen bruteforcelle.

Tavoitteena on seuraavaksi saada brute forcen testit ja dokumentoinnit valmiiksi, jonka jälkeen alan toteuttamaan joko bellman-fordia tai dijkstraa. Tavoitteena saada ensiviikolla toteutettua kaikki algoritmit, jotta voin aloittaa tarvittavien tietorakenteiden rakentamisen.

### Maanantai 17.12

Bellman-Fordin testien ja javadocsien teko ja pientä refaktorointia. Bellman-Fordin algoritmi pilkottu luokan sisällä keskeisiin komponentteihin copy-pasten välttämiseksi ja toimintojen selkeyttämiseksi. Päädyin myös suorittamaan kaikenlaista refaktorointia koodin suhteen. Paikkaluokkaa paranneltu, Bellman Fordista korjattu bugeja. Bellmanista oli jäänyt epähuomiossa V-1 määrä suorituksia pois joka aiheutti sitten jo melkolaisia ongelmia, ilmeni vasta suuremmilla syötteillä. Myös Dijkstra on nyt toteutettu. Brute Forcea olen hieman miettinyt, vertailukohtana se on ehkä nopeuden puolesta hieman huono sillä se ei todellakaan etsi nopeinta reittiä, sitten taas toisaalta jos sen suorittaisi useita kertoja tulisi siitä todella, todella hidas joka ei välttämättä olisi tarkoituksenmukaista. Seuraavaksi tarkoituksena olisi lähteä toteuttamaan A\*.

## **Tiistai 18.12**

A\* kasailua ja säätöä. Illalla tajusin että kovakoodattu lopetuspaikka ko. sokkelossa olisi kovin hyvä idea, sillä tässä tapauksessa toteutuisi todennäköisesti A\*:n pahin tilanne ja se joutuisi käymään kaikki vaihtoehdot lävitse.

## **Keskiviikko 19.12**

Armotonta refaktorointia, joka aiheutui kovakoodatun lopetuspaikan muutoksesta, rikkoi todella paljon asioita, jotka pitäisi nyt olla paremmin toteutettu. Tutkailin myös ehdotusta LinkedList toteutuksesta luopumisesta ja toteutinkin sen HashSetinä, joka paransi merkittävästi A\*:n nopeutta. Vielä ehkä hieman epäselväksi jäi voisiko sitä parantaa vielä jotenkin. Kunhan olen testaukset ja javadocsit saanu aikaiseksi tarkoitus olisi aloittaa tietorakenteiden toteutuksia.

Testaukset ja javadocsit tähän mennessä tehdyille luokille valmiita. Loin tietorakenne luokat valmiiksi ja sain tehtyä ja otettua käyttöön pinon.

## **Torstai 20.12**

Kekoluokka saatu pienellä taistelulla valmiiksi. Vaati paikkaluokkien refaktorointia ja yhdistämistä ja pientä kokeilua, että keon sai implementoitua oikein. Testattu myös keon toimintaa niin että vertasin priority queueen tuloksiin sekä bellman-ford algoritmin antamiin tuloksiin. Ajatuksena on seuraavaksi alkaa hahmottelemaan HashSetin implementointia ja toteutusta ja toteuttaa se, sekä kirjoittaa testit ja javadocsit kaikille luokille loppuun.

Hajautustaulu rakenteineen on nyt myös maalissa. Yllättävää oli huomata miten paljon A\*:n suoritus parani kun otti käyttöön eka HashSetin ja sen jälkeen vaihtoi omaan hajautustauluun ja vielä lopulta hajautustaulun sisäiset linkitetyt rakenteet omaan. 100 x 100 sokkelolla A\*:n aikaavievyyys putosi n. 40ms:sta 19 ms:iin. Tämän jälkeen tarkoituksenani on aloittaa kirjoittamaan testejä lopuille luokista ja myös javadocsit. Osa virheistä ja bugeista ja varsinkin rakennellessa olleista ongelmista on jo löytynyt muuten testaillessa koodia.