

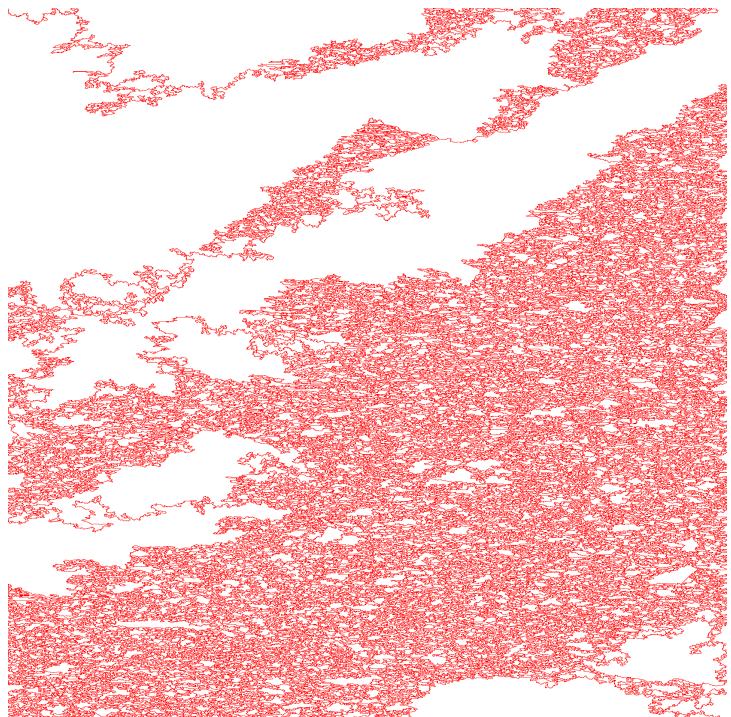
# Viikkoraportti 2

## Mitä opin ja tein tällä viikolla?

Tällä viikolla keskityin omien tietorakenteiden tekemiseen, sekä sain niitä käyttäen toteutettua myös mielestäni toimivan A\* algoritmin. Omia tietorakenteitani on NodeList, NodeStack ja Heap. Suurimman oivalluksen koin tehdessäni NodeListiä. Tein siten, että NodeListin taulukko alustetaan aina neljän kokoiseksi, koska nodella on ruudukossa aina 4 taulukkoa. Ajattelin että tästä ei koituisi suuria muutoksia, mutta ohjelman suoritusnopeus suurella syötteellä putosi viidestä sekunnista neljään sekuntiin, kun vaihdoin oman listani ArrayListin tilalle. Stackin toteuttaminen onnistui ulkomuistista ja Heappia jouduin luntaamaan pseudokoodista. Heappia toteuttaessani tutustuin javan system.arraycopyn toimintaan, kun yritin selvittää onko olemassa  $O(1)$  tapaa kopioida lista. En sellaista kuitenkaan löytänyt joten operaatio on nyt  $O(n)$ . Siinä missä javan priorityqueueella keskiarvo on noin 5.1s, minun heapillani keskiarvo on 5.32. Uskon eron johtuvan siitä että system.arraycopy tekee listojen kopioinnin fiksummin?

Omien tietorakenteiden lisäksi toteutin A\* algoritmin omilla tietorakenteillani. A\* toteutus vei yllättävän paljon aikaa, ja sainkin ensin mitä ihmeellisimpiä tuloksia nopeimmaksi reitiksi. Nyt syöte näyttää samalta kun Dijkstrassa. Olin käsittänyt että A\* vaatisi uusia tietorakenteita, mutta vanhat työkaluni riittivät sen toteuttamiseen.

*A starin ensimmäisen yrittelyn sekavahko reittiehdotus esteettömällä kartalla pisteestä pisteeseen.*



## Mitä jäi epäselväksi?

Toimiiko A\* algoritmini niinkuin A\* algoritmin kuuluukin toimia? Ihmettelin kun en

tarvinnut uusia tietorakenteita. Tekeekö system.arraycopy oikeasti listojen kopioinnin nopeammin kun oma metodini, yritin tutkia asiaa Googlella, mutta jäi silti aavistuksen epäselväksi. Pitääkö tietorakenteille koodata metodeita joita Dijkstrassa tai A starissa en käytä, kuten Heapin increase key tai Listin remove jonka jo koodasinkin. Pitääkö myös testit kommentoida javadoceilla? Osan testeistä voi kuitenkin jo päätellä testin nimestä. Pitääkö tietorakenteet toteuttaa siten että esim NodeListini voisi tallettaa mitä tahansa muutakin kun Node olioita? Mikäli pitää, mitenhan se tehdään? Onko alue nyt tarpeeksi suuri korkeampiin arvosanoihin?

### **Mitä ensiviikolla?**

A\* on vasta saatu kasaan, sitä täytynee siistiä, javadocata ja testata. Lisäksi haluan testata toimiiko a\* nopeammin kun dijkstra ja tehdä siitä merkintöjä testidokumenttiin. Myös yleistä koodin siistimistä, muutamia testejä ja loput dokumentit kuntoon.