

Refaktoroin reitinhakuluokkaa siistimmäksi pilkkomalla metodeja pienemmiksi. Samalla uudelleennimesin hiukan asioita. Lisäsin muutaman testin erilaisille reiteille. Luin Wikipedia-artikkelin prioriteettijonosta. Prioriteettijono ei nimestä huolimattaan ole jono, vaan abstrakti tietotyyppi, joka voidaan toteuttaa usealla eri tavalla. Yleisin toteutustapa lienee keko.

Kokeilin toteuttaa prioriteettijonoa ja luin eri lähteistä artikkeleita prioriteettijonosta. Jäi hiukan epäselväksi se, mikä prioriteettijono oikeastaan on. Onko se keko vai häh?

Tein kekoa hyödyntävän prioriteettijonon, joka pitää pienimmän prioriteettiluvun solmun jonon kärjessä. Jono pitää jonopointterilla kirjaa, missä kohtaa solmutaulukkoa on jonon pää ja lisää uuden solmun aina jonon loppuun. Jokaisen lisäyksen ja poiston jälkeen tarkistetaan kekoehto. Onkohan tämä oikea lähestymistapa?

Tein omalle jonolleni suorituskykytestin, jossa jonoon lisätään 3 miljoonaa solmua suurimmasta alkaen. Jono järjestää pienimmän prioriteetin solmun jononärkeen noin kahdessa sekunnissa. 300 000 solmulla operaatiossa menee 0,146 sekuntia, joten aikavaativuus lisäämiselle lienee  $O(n \log n)$ . Tämä lienee ihan ok?

Lisäsin kaikille julkisille metodeille Javadocit ja tein kaikille luokille lisää testejä.

Ohjelmasta on nyt valmiina toimintalogiikka ja oma tietorakenne.

Seuraavaksi olisi tarkoitus tehdä prioriteettijonolle lisää testejä ja korjata esiintulleita ongelmia. Olisi myös tarkoitus integroida prioriteettijono reitinhakuun, ja tehdä pääohjelmaan jonkinlainen käyttöliittymä.