

## Yleistä

Testaus suoritettiin JUnit-testeillä. Tein testit lähes kaikesta keksimästäni, testasin, että kekorakenne säilyy operaatioiden eri vaiheissa, että keon huipulla oleva alkio on se, joka halutaankin sen olevan, että keon koko on niin kuin pitää ja että solmujen lapset/sisarukset/vanhemmat/ ovat oikein. Useasti kävi niin, että pienillä syötteillä kaikki toimi niin kuin pitääkin, mutta suuremmilla se menikin pieleen. Tällöin testejä ajamalla etsin sen kohdan, missä homma menee metsään, jolloin tiesi, mikä oli pienin mahdollinen tapaus, missä operaatio ei toimi ja simuloida kyseistä tilannetta paperilla. Näin virhe usein löytyikin muutaman tunnin ahellyksen jälkeen.

## Mitä testattiin

Testasin jokaisen keon jokaista metodia vähintään muutamalla testillä. Testasin yleensä aluksi jotain hyvin triviaalia, esimerkiksi, että keko on alussa tyhjä, tai että siihen voi lisätä yhden solmun, ja lähdän sen jälkeen kehittämään ohjelmaa niin, että siihen voisi lisätä enemmänkin solmuja, tai että sen koko kasvaa niitä lisättäessä.

Lopullisena varmituksena keon toimivuudesta testasin, että kekoihin voi lisätä solmuja järjestyksessä, käänteisessä järjestyksessä sekä satunnaisessa järjestyksessä, ja poistettaessa ne tulevat oikeassa suuruusjärjestyksessä ulos.

Binäärikeolle tein jonkin verran käsin testausta hioessani metodia insert siten että lisäsin kekoon monta alkiota ja tulostin taulukon sisällön aina lisäyksen jälkeen. Taulukkorakenteen vuoksi, tämä oli kätevä ja luonteva tapa etsiä, missä vaiheessa virhe tapahtui. Muille keoille en tehnyt mitään käsintestausta, osittain laiskuudesta (ei ollut main-metodia valmiina) ja osittain siksi, että toteutuksen erilaisuudesta johtuen ei ollut yhtä kätevää tapaa tulostaa keon sisältöä kokonaisuudessaan.

## Suorituskykyvertailusta

Suorituskyky vertailu suoritettiin siten, että jokaiseen kekkoon lisättiin 10 000 - 10 000 000 alkiota järjestyksessä aikaa mitaten kymmenen kertaa, ja otettiin saaduista ajoista keskiarvo. Testi oli niin raskas, että sen suorittaminen kesti ikuisuuksia tai jopa kaatui, jos testeja yritti tehdä esim. 100, joten päätin tyytyä 10:een. Syötin saadut arvot taulukkolaskentaohjelmaan, ja sain aikaiseksi hienoja kuvaajia operaatioon kuluneesta ajasta eri tietorakenteiden osalta. Tein vastaavan vertailun myös operaatiolle deleteMin. Otin vertailuihin mukaan myös javan PriorityQueue:n. Kuvaajat ovat nähtävillä toteutusdokumentissa.