

Testausdokumentti

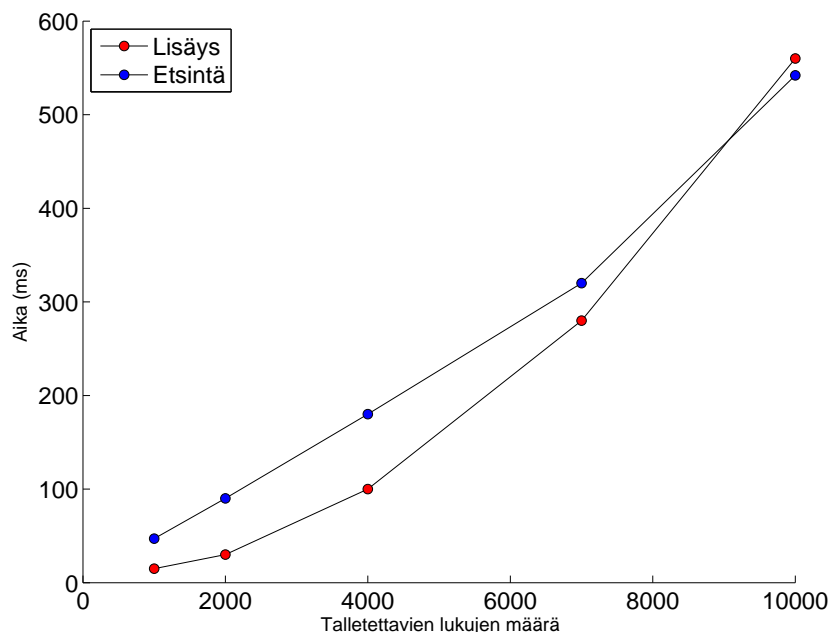
Tomi Heiskanen

14. joulukuuta 2014

Testit

Tietorakenteita on testattu kahdella tavalla, ensiksi koodia kirjoittaessa on suoritettu JUnit-testejä ja saatu toimiva koodi ja sen jälkeen on suoritettu suorituskykytestaukset. JUnit-testeissä on lisätty, poistettu ja etsitty solmuja ja tulostettu puun solmujen tietosisältöjä järjestyksessä. Suorituskykytestauksessa on generoitu suuri määrä kokonaislukuja taulukkoon ja sitten lisätty puihin järjestyksessä. Lisäämiseen kulunut aika on mitattu. Lisäksi on mitattu aika, joka kuluu tietyn arvon etsimiseen, poistamiseen ja suurimman ja pienimmän alkion etsimiseen.

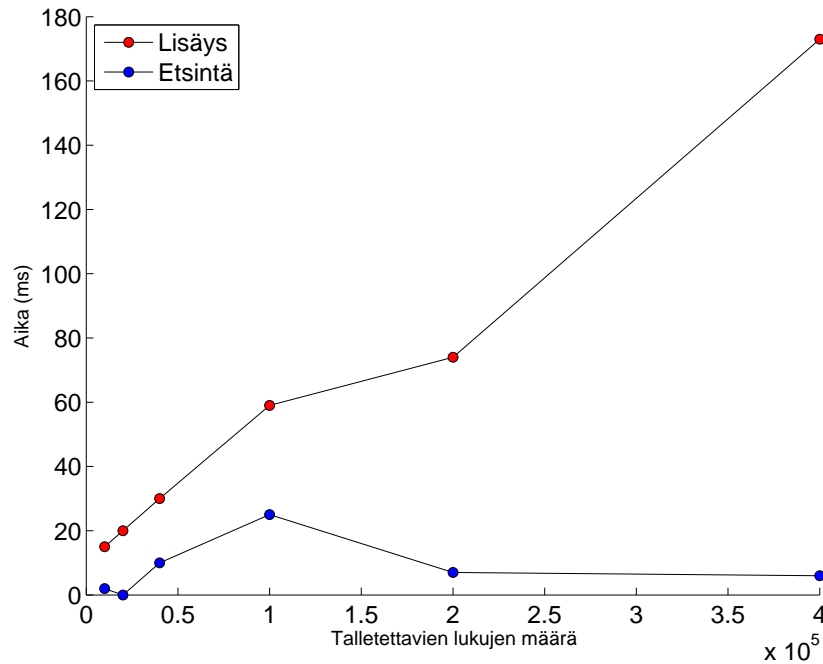
Suorituskykytestaukset tehdään suorittamalla testipääohjelma kustakin puusta erikseen. Suorittamalla se, luodaan puu ja siihen aletaan lisäämään solmuja. Ensin luodaan kokonaislukutaulukko, joka on järjestyksessä, ja aletaan lisäämään solmuja, joiden tietokentän arvoksi tulee kokonaisluku, puuhun. Etsimisen testaamiseksi luodaan kokonaislukutaulukko, jossa on satunnaisesti tuotettuja kokonaislukuja 10000 ja katsotaan kuinka kauan niiden etsimiseen menee aikaa. Suurimman ja pienimmän luvun etsiminen tapahtui sen verran nopeasti, että jokaisella alkioden määrällä sain ajaksi 0 ms.



Kuva 1: Binäärihakupuuhun lisäämiseen ja siitä hakemiseen kuluvat ajat

Binäärihakupuuhun lisäämisen, poistamisen ja etsimisen aikavaativuus on pahimmillaan $O(n)$. Kuvan perusteella näin näyttäisi olevan. Kuitenkin kun lisättävien alkioden määrä lisääntyy huomattavasti, aikavaativuus näyt-

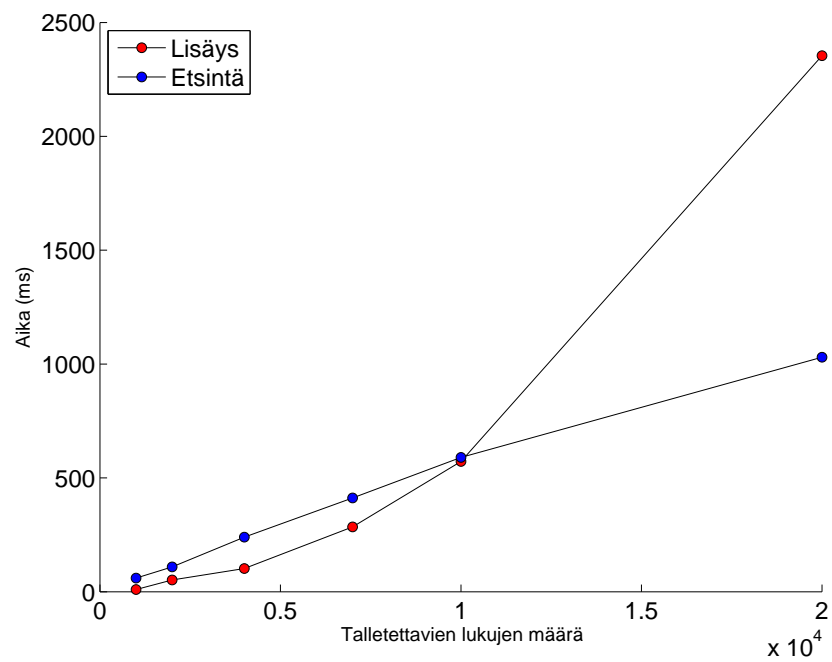
täisi olevan pahempi, kuin $O(n)$. En osaa selittää tätä, mutta arvelisin sen liittyvän muistin määrään. Etsimisen aikavaativuudelle näin ei näyttäisi tapahtuvan.



Kuva 2: AVL-puuhun lisäämiseen ja siitä hakemiseen kuluvat ajat

AVL-puun lisäämisen, poistamisen ja etsimisen aikavaativuus on pahimmillaan $O(\log n)$ ja näin näyttäisi testien alkupään mukaan olevan. AVL-puun testissä käy samalla tavalla kuin binäärihakupuulla, jolla solmujen määrän lisääntyessä huomattavasti solmujen lisäämisen aikavaativuus näyttäisi olevan pahempi kuin se on. Arvelisin syyn olevan saman kuin binäärihakupuulla.

Punamustapuun testit menevät edellisten mukaan. Suurilla solmujen määrällä testitulokset lisättävien solmujen osalta poikkeaa aikavaativuusanalyysin tuloksesta. Syy lienee sama kuin edellisilläkin puilla.



Kuva 3: Punamustapuuhun lisäämiseen ja siitä hakemiseen kuluvat ajat