

Tietorakenteiden testaus

Vertailin omien tietorakenteiden suorituskykyä javan omiin muuntelemalla dataStructureTests.txt tiedostossa olevaa koodia tarpeen mukaan.

ArrayList

ArrayListiä testasin suorituskyky testauksissa tutuilla readme ja medium tiedostoilla. Testin ideana oli että tiedostossa luetaan ensin kaikki merkit ja lisätään ne arraylistaan, sitten käydään tämä lista läpi kokonaan ja haetaan jokainen merkki listalta ja muodostetaan tästä stringi. Eli selkeämmin sanottuna add ja get testaus.

ArrayList versio	Oma	Java
README.TXT (ms)	14	11
medium.txt (ms)	110840	61860

Kuten taulukosta huomaa on Javan ArrayListi melkein 2 kertaa nopeampi kuin omani. Add operaation hitaudesta tämä varmaan johtuu. Jos aikaa jää yritän ehkä parannella sitä viimeisellä viikolla.

PriorityQueue

PriorityQueuea testasin huffmanin koodauksesta tutulla tyylillä, koska priorityqueueeni on tehty vain sitä tarkoitusta varten eikä se toimi oikein missään muussa tapauksessa. Eli ideana oli että saadaan TreeBuilder methodit saavat parametrina hashmapin johon on lisätty kaikki tekstistä löytyvät merkit ja vastaavat frekvenssit. Eli jouduttiin pyörimään hyvin pienissä jonon kooissa koska ASCII merkkejä on se 256 ja yleensä englanninkielisessä tekstissä on vielä vähemmän niistä käytöissä. Tästä testin tulos.

PriorityQueue versio	Oma	Java
bible.txt	2	1
medium.txt	2	1
README.txt	2	1

Koska jonon koot ovat pieniä ei voida sanoa varmuudella mitään tästä tuloksesta, mutta veikkasin että on se javan oma sen kaksi kertaa nopeampi myös silloin jos modifioisin priorityqueueeni toimimaan muilla kuin char Nodeilla.

HashMap

HashMapia testailin lisäämällä tekstistä merkkien esiintyvyydet hashmapin eli put-operaation. Alla testin tulokset .

HashMap versio	Oma	Java
medium.txt	354	334
bible.txt	6464	6112

Olin positiivisesti yllättänyt hashmappini tehosta, päästään hyvin lähelle javan oman suorituskykyä.

Get ja containsKey operaatiot näyttäisi toimimaan myös kummassakin HashMapissa yleensä vakioajassa ja pahimmassa tapauksessa lineaarisessa ajassa.