

Heidi Jauhiainen
011512192
Ohjaaja: Kristiina Paloheimo
Tietorakenteiden harjoitustyö, loppukesä 2012
Helsingin yliopisto, Tietojenkäsittelytieteen laitos
31.8.2012

Testausdokumentti

WordI - Sanaindeksi Trie-puuta käyttäen

Ohjelmaa testattiin ohjelmoinnin aikana ja sen jälkeen JUnit testeillä. Ohjelman ollessa jo valmis sitä testattiin kahdella erillisellä testiluokalla, jotka testasivat eri metodien ajankäyttöä erilaisilla hakulausekkeilla. Lisäksi tekstipohjaisen käyttöliittymän toimintaa ja tietojen kirjoittamista muistiin seuraavaa käyttökertaa varten testattiin käsin.

JUnit testejä oli kaiken kaikkiaan 28 kappaletta. Niistä neljä testasi trie-puun toimintaa (PuuTest) ja kolme puun solmujen toimintaa (SolmuTest). Loput 21 JUnit testiä testasivat koko ohjelman toimintaa lukuunottamatta käyttöliittymää (SanaIndeksiTest). Näistä testeistä neljä testasi tiedostojen tallentamista järjestelmään ja kaksi tallennettujen tiedostojen nimien printtaamista. Loput 15 testasivat hakua erilaisilla hakutermeillä ja eri määristä tiedostoja. Testauskattavuus koko ohjelmassa oli 62 prosenttia. Kaikkien muiden luokkien paitsi käyttöliittymästä huolehtivan Tiralabra-luokan, sen apulaisen Odottaja-luokan, muistiin kirjoittamisesta huolehtivan Muistiin-luokan ja kahden testiluokan kattavuus oli 100%. Näitä mainittuja luokkia ei testattu ollenkaan JUnit testeillä vaan niiden toimivuus testattiin käsin.

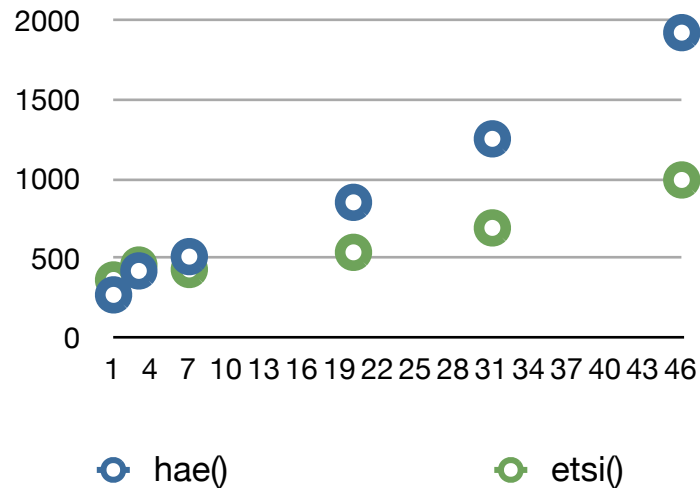
AikaTestit-luokalla testattiin ohjelman eri metodien viemää aikaa erilaisilla syötteillä. Ensimmäinen testi mittasi hae-metodin aikaa 1-4 hakusanalla, kun tallessa on ainoastaan muutamia rivejä tekstiä. Sen jälkeen lisättiin useita tiedostoja joissa oli yhteensä kymmeniä tuhansia rivejä. Sitten testattiin koko haun viemää aikaa tulosten printtaamisen kanssa. Hae-metodin aikaa testattiin vielä uudelleen 1-4 hakusanalla. Lopuksi hae-metodia ja etsi-metodia testattiin samoilla yksittäisillä eripituisilla sanoilla. Sanojen printtaus testattiin jokaisella luokan ajolla ainoastaan kerran ja tuloksia listattiin ylös vertailua varten. Kuten odottaa saattaa, sillä kuinka monta riviä haku tuotti tuloksenaan oli suuri vaikutus haun vaatimaan aikaan. Aluksi luokan muut testit tehtiin 10 kertaa kullakin syötteellä ja niin, että jokaisen testin aika printattiin näkyviin. Tästä näki, että luokka tekee jonkinlaisia alustuksia ensimmäisten testien aikana ja vielä seuraavien syötteiden ensimmäisen testin aikana (ks.

kuva 1). Lisäksi jotkut testikerrat tuottivat huomattavasti muita suuremmat tulokset, mikä vaikutti keskiarvoihin. Lopulliset testit tehtiin niin, jokaista syötettä testattiin 1100 kertaa, joista ensimmäiset 100 jätettiin huomioimatta keskiarvossa. Lisäksi luokan alussa tehtiin 100 kertaa 1000:n testin läpiajo kuudelle eri syönteelle. Vasta näiden alustusten jälkeen tehtiin varsinaiset testit.

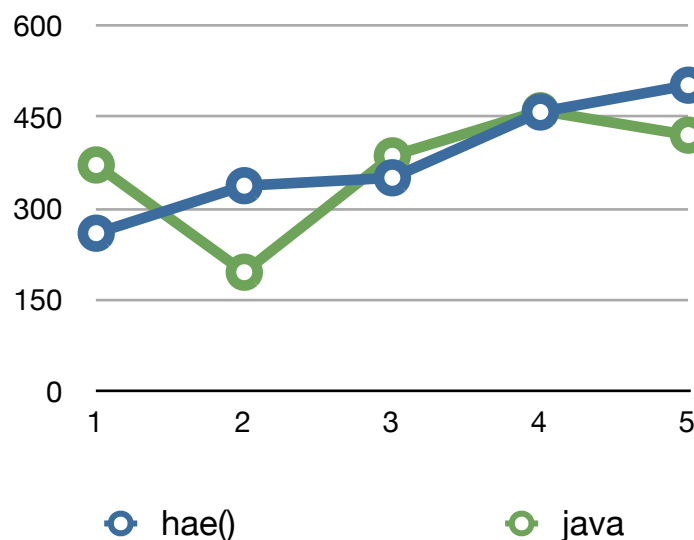
sun	sun&him*	brother/(sun&him*)	(brother/a)/(sun&him*)
33825000	52000	30000	67000
11000	16000	34000	57000
7000	16000	28000	55000
7000	41000	28000	73000
7000	16000	28000	67000
6000	16000	27000	41000
7000	16000	34000	59000
6000	15000	27000	40000
7000	15000	27000	41000
6000	22000	27000	258000
19855000	50000	27000	45000
11000	16000	31000	39000
6000	15000	26000	39000
7000	23000	26000	44000
7000	15000	26000	39000
7000	16000	26000	39000
6000	15000	30000	44000
6000	16000	25000	38000
6000	14000	26000	40000
7000	27000	26000	39000

Kuva 1. AikaTestit-luokan ensimmäisen metodin tuloksia kahdelta eri kierrokselta testien alkuvaiheessa.

Hae-metodin testit osoittivat, kuten odottaa saattaa, että useammasta rivistä hakeminen vie enemmän aikaa kuin vain muutamasta. Jostain syystä myös yhden sanan haku vei enemmän aikaa vaikka tähän ei rivien määrällä pitäisi olla vaikutusta. Hae- ja etsi-metodien vertailu samoilla eripituisilla sanoilla osoitti, että hae-metodi vie hieman enemmän aikaa pidemmillä sanoilla kuin etsi-metodi, mikä tietysti johtuu siitä, että hae-metodi käy hakusanan kaikki kirjaimet läpi jo ennen kuin sanat haetaan puusta (kuva 2).

Kuva 2. Hae- ja etsi-metodien vertailu

Vertailu-luokalla verrattiin ohjelman hae-metodin aikavaativuutta verrattuna javan valmiin TreeSet-rakenteen käyttämään aikaan. Luokassa syötettiin kumpaankin puuhun 100000 numeroa, joilla oli kullakin vain yksi rivi, jolla ne "esiintyivät". Ennen varsinaisia testejä tehtiin 200 kertaa 1000 testin läpiajo viidelle eri syötteelle. Lisäksi molempien puiden hausta hylättiin sata ensimmäistä tulosta. Lopuista 1000 hausta otettiin keskiarvo. Kuvasta 3 näkyy kuinka TreeSet, joka on punamusta puu, kahden pituista hakua lukuunottamatta pysyy melko tasaisena. Javan TreeSet puun aikavaativuus onkin $O(\log n)$, jossa n on solmujen määrä puussa [Java™ Platform, Class TreeSet<E>]. Koska solmujen määrä ei vaihtele testin aikana, vaihtelut hauan ajassa johtuvat ainoastaan siitä kuinka nopeasti sana oikeasti löytyi. Hae-metodin käyrä puolestaan kuvaa aikavaativuutta $O(m \log 1)$, jossa m on kirjainten määrä haetussa sanassa.

Kuva 3. Hae-metodin verrattuna punamusta-puuhun

Testien tulokset näkyvät myös Testit.xls:ssä, joka sijaitsee samassa kansiossa kuin muut työn dokumentit.

Lähteet:

Java™ Platform Standard Ed. 6, Class TreeSet<E>. <http://docs.oracle.com/javase/6/docs/api/index.html?java/util/TreeSet.html>. Haettu 31.8.2012.