

Testausdokumentti

Ohjelman testaaminen on suhteellisen hankalaa, ja paras tapa seurata sen toimintaa on aloittaa harjoitusottelu Vindiniumissa. Tämä tapahtuu syöttämällä ohjelmalle ensin oma koodi, sitten numero 1 ja sitten sana "basic". Tämän jälkeen ohjelma antaa url-osoitteen missä voi seurata miten botti liikkuu. Lisäksi tekstikäyttöliittymästä näkee botin jokaisen vuoron päätöksen ja missä se itse on.

Yksittäisiä osia varten on Junit testejä. Ne saa ajettua normaalien testien tapaan. Jokaisella tietorakenteella on yksi testi nimeltään speedTest() jossa verrataan tietorakenteita Javan valmiiksi tarjoamiin ratkaisuihin. Tiedot tulevat tekstikäyttöliittymään, ja tämän dokumentin lopussa on 10 testauksen keskiarvoista tehdyt kaaviot, joista näkee miten eri tietorakenteet vertaavat.

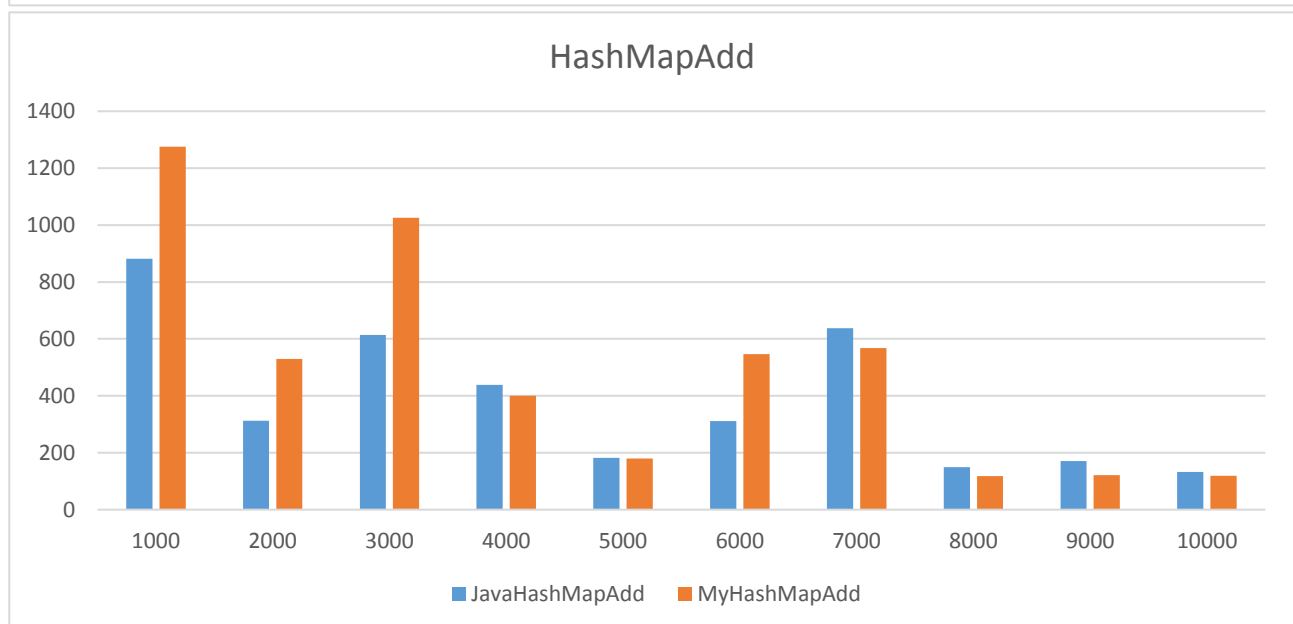
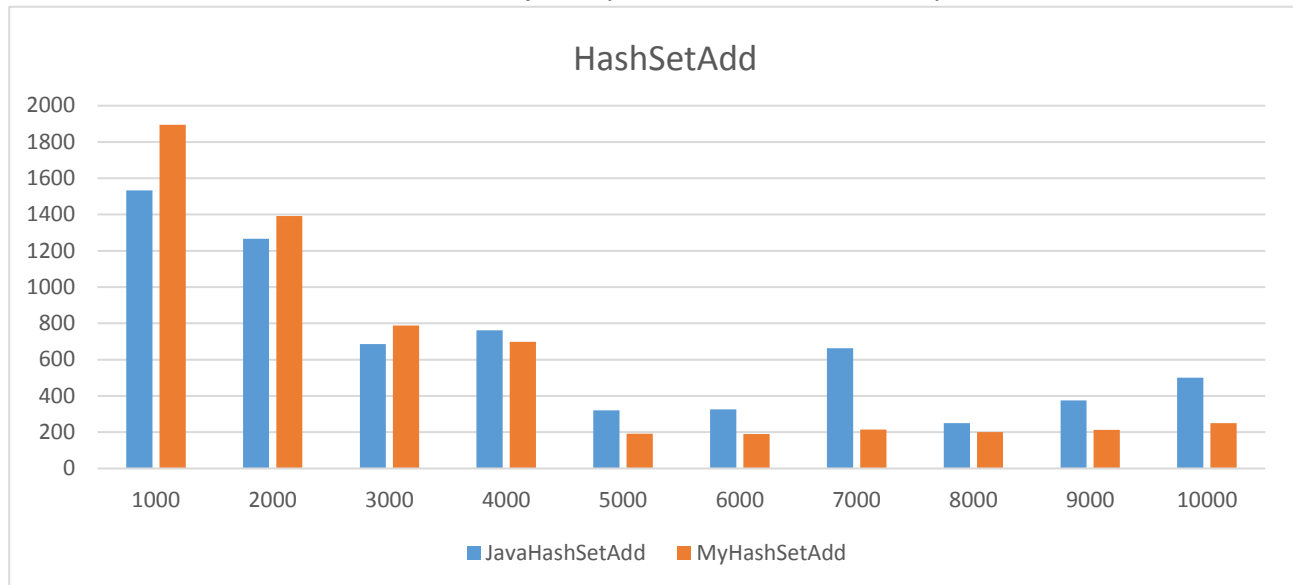
Kaikki muut tietorakenteet paitsi linkitetty lista ovat suunnilleen yhtä nopeita kuin valmiit versiot. Linkitetyn listan tapauksessa add()-metodi on hitaampi kuin Javan versio. Syynä tähän on luultavasti se, että Javan LinkedList-luokka on kahteen suuntaan linkitetty, kun itse tekemäni on vain yhteen suuntaan linkitetty.

Tietorakenteita testattiin eri kokoisilla joukoilla. Pienin oli 1000 ja suurin 10000. Testauksessa syötettiin 1000 kerrallaan sekä javan että omiin tietorakenteisiin, ja tämän jälkeen sieltä haettiin jokin alkio. Linkitettyssä listassa käytettiin keskeltä otettua alkioita. Testauksessa selvisi, että jos haettiin lopusta niin vaikka Javan kahteen suuntaan linkitetty lista ei välitä siitä, niin itse toteutettu kuluttaa noin 20000ns lisää per 1000 alkioita. Muiden kanssa nopeudet ovat suunnilleen samoja.

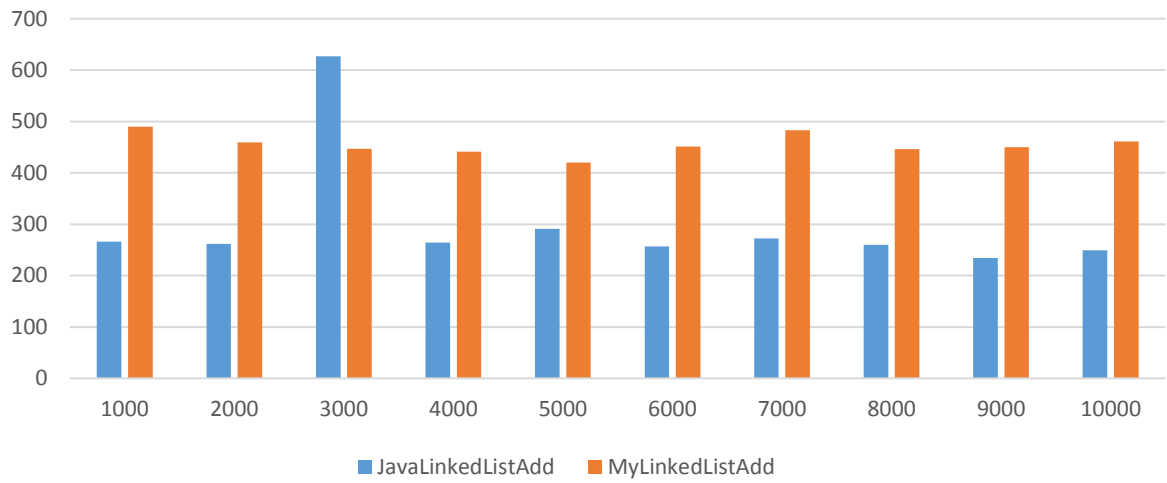
MyHashSet ja MyHashMap luokkien hajautus taulukoiden koko kasvatettiin 20000 tätä testausta varten, sillä A* algoritmilla niiden kooksi on luokiteltu 2x isompi kuin kaikkien solmujen määrä kartalla. MyHashSet:in 8000 syötteen koolla, sen etsiminen jostain syystä muuttui 10x hitaammaksi, ilman mitään löydettyä syytä. Kaikilla muilla syötteillä haku on suunnilleen yhtä nopea kuin Javan HashSet.

A* algoritmin testaus on vaikeaa, koska sen toteuttava metodi vaatii toimiakseen kokonaisen GameState olion. Tätä varten on src/main/test/resources kansiossa kaksi valmista JSON tiedostoa, joissa on kaikki mitä tarvitaan tällaisen olion luomiseksi. A* testi luokka osaa muokata näistä tiedostoista valmiit oliot, joita käytetään testaukseen.

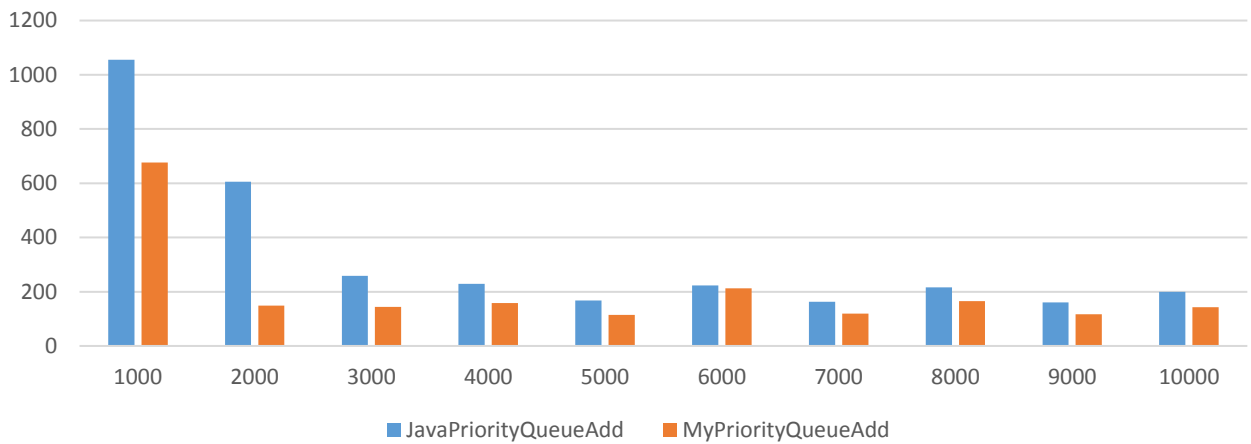
Kaavioita jokaiselle tietorakenteelle. Alhalla on syötteen koko, ja vasemmalla on aika nanosekunteina. Kuvastaa yhtä tapahtumaa, eli Add taulukoissa 1000 syöte kuvastaa syötteiden 1-1000 keskiarvoa. Kaikkien haku metodien ajat kuvastavat yhden arvon hakemista tietorakenteesta kun arvoja on yhteensä 1000-10000 kpl.



LinkedListAdd



PriorityQueueAdd



LinkedListGet

