

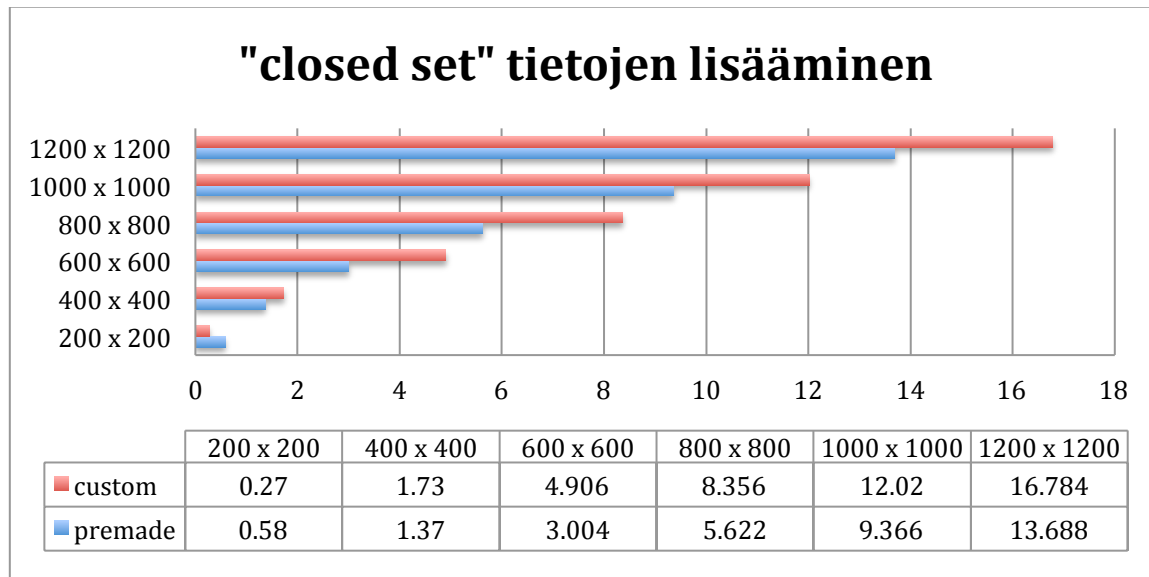
Suorituskykymittauksia

Käytin työssä ensin Javan valmiita tietorakenteita ArrayList ja PriorityQueue. Korvasin ne sittemmin omilla toteutuksilla niin, että ArrayListin korvasi simppele boolean 2D-taulukkoon pohjaava rakenne (joka toimii loistavasti juuri tässä käytössä) ja PriorityQueuen korvasi oma binäärihekkoni.

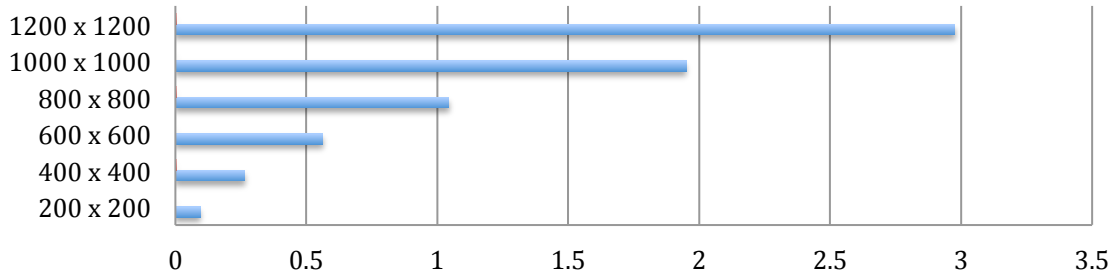
Mittasin sitten aikoja, mitä eri rakenteilla meni suorittaa tiettyjä operaatioita. Ajoin testit kohtalaisen suurilla iteraatiomäärillä ja laskin niiden mittaustuloksista keskiarvot.

Tietorakenteiden testit

Ensimmäinen testi testaa koko ruudukon sisällön lisäämistä "closed set" -rakenteisiin. Toinen testi tarkistaa, onko closed setissä kahta tiettyä gridCelliä (joista toinen on setissä ja toista ei ole). Kolmas testi puolestaan lisää koko ruudukon sisällön "open set" -rakenteeseen. Solujen arvot, joiden mukaan PQ/hekkoni sijoittaa jokaisen solun oikeaan paikkaansa, on arvottu randomilla ohjelman suorituksen alussa. Arvot ovat molemmille toteutuksille samat.

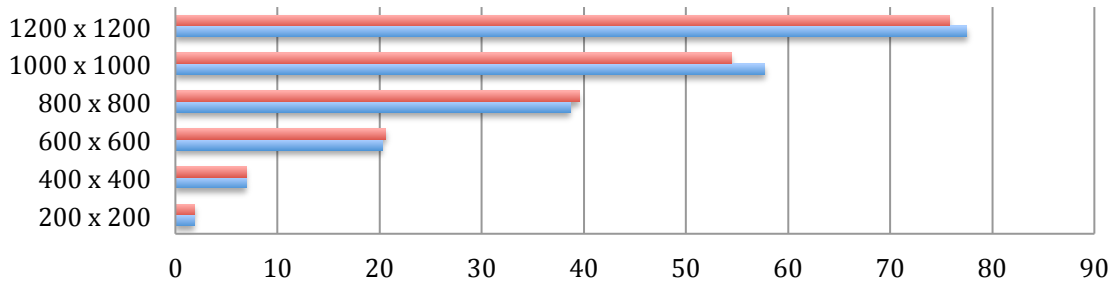


"closed set" tietojen etsiminen



	200 x 200	400 x 400	600 x 600	800 x 800	1000 x 1000	1200 x 1200
custom	0	0.002	0	0.002	0	0.002
premade	0.096	0.264	0.562	1.044	1.95	2.976

"open set" tietojen lisääminen



	200 x 200	400 x 400	600 x 600	800 x 800	1000 x 1000	1200 x 1200
custom	1.882	6.926	20.616	39.548	54.452	75.758
premade	1.832	6.952	20.256	38.678	57.69	77.516