

Testausdokumentti

JUnit-testeillä testasin käyttöliittymä- ja rajapintaluokkia lukuunottamatta kaikkien luokkien perustoimintaa. Testien kattavuuden arvioinnissa auttoi pit-mutaatiotesti, joka antoi testien rivikattavuudeksi 100% ja mutaatiokattavuudeksi 95%.

Algoritmien testaus

Luokkaa SolvabilityDeterminer testasin etsimällä sellaisia 15- ja 8-pelejä, joista tiesi, ovatko ne mahdollisia tai mahdottomia ratkaista. Valitsin kolme ratkaisematonta peliä ja kolme sellaista, joihin on ratkaisu, ja katsoin osaako algoritmi määrittää niiden ratkaistavuuden oikein.

Heuristisista funktioista testasin pääasiassa sen, antavatko ne sellaisen arvion jäljellä olevien siirtojen määrästä kuten niiden pitäisi. Valitsin muutaman satunnaisen 15- ja 8-pelin, laskin millaisen arvion pitäisi olla ja vertaisin sitä algoritmin antamaan. Lisäksi testasin, että kunkin algoritmin update()-funktio toimii eli että algoritmi antaa uuden arvion oikein yhden siirron jälkeen.

IDA*-algoritmista testasin, että algoritmi antaa siirtojen määrän oikein etsimällä pelejä, joiden minimisiirtojen määrä oli tiedossa. Haastavampaa oli tutkia, onko algoritmin antamat siirrot oikeita. Testasin sitä valitsemalla yksinkertaisen pelin, josta oikeat siirrot oli helppo nähdä. Kokeilin myös päätyykö peli tavoitetilään, jos pelin laattoja siirretään algoritmin antamien siirtojen mukaisesti.

Tietorakenteiden testaus

Kesken

Suorituskykytestaus

IDAStar Manhattan-etäisyys heuristiikan kanssa

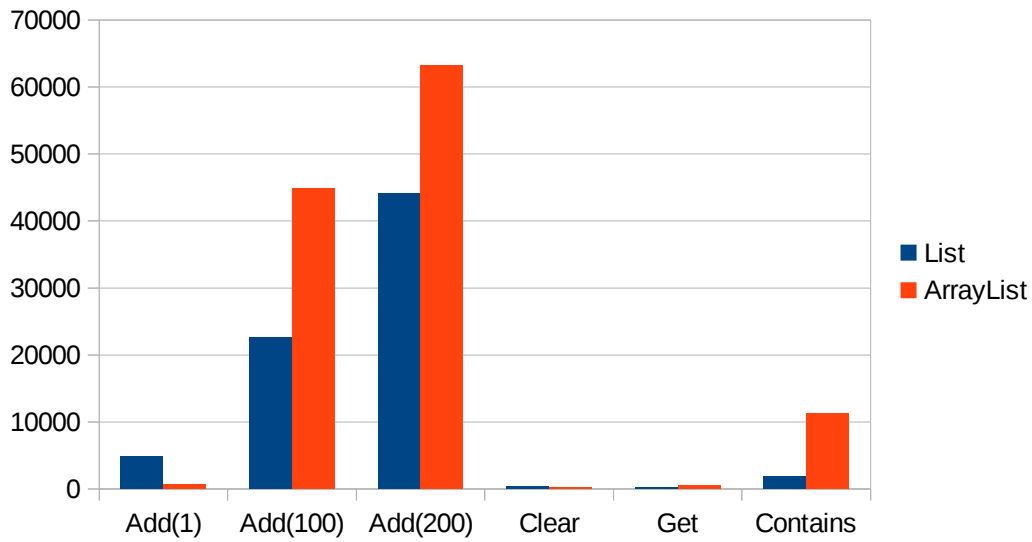
****8-peli****

****15-peli****

Tietorakenteiden vertaus Javan vastaaviin

Vertasin tekemääni List-tietorakennetta Javan valmiiseen ArrayList-tietorakenteeseen. Tein testit kokonaisluvuilla. Testasin yhden, 100 ja 200 järjestyksessä olevan luvun lisäystä ja lisäksi niiden poistoa. Testasin myös get- ja contains-metodeja, joissa hain satunnaista listoilta löytyvää lukua. Kaikki testit toistettiin 100 kertaa ja laskettiin niistä keskiarvot.

	List	ArrayList
Add(1)	4880ns	586ns
Add(100)	22579ns	44826ns
Add(200)	44071ns	63167ns
Clear	360ns	175ns
Get	253ns	523ns
Contains	1898ns	11233ns



Yksinkertainen toteutus lista-tietorakenteesta näyttää pärjäävän nopeudessa Javan valmiille ainakin näiden metodien osalta ja näin pienillä lukumäärillä. List-tietorakenteestani puuttuu kuitenkin aika paljon metodeja, mitä ArrayListillä on, kuten `remove()`, koska en tarvinnut niitä ohjelmassani.