

# Testausdokumentti

## Ohjelman testaus

Miinaharavanratkojaa on testattu JUnit testeillä ja testien kattavuus tulokset löytyvät pit-rapotti kansioista. Pit mutaatiotestauksen rivikattavuus algoritmille on tällä hetkellä 88% ja mutaatiokattavuus 51%.

Ratkaisijan testaaminen oli siinä mielessä haastavaa, että pystyin vain laittamaan testit suorittamaan miinaharavoja, jotka sen pitäisi pystyä ratkaisemaan ja tarkistamaan että lauta on ratkaistu. Testauksessa olevat kenttien koko pitää olla myös vähintään 8 x 8, ettei peliä voita vain yhdellä klikkauksella. Poikkeuksena on 3 x 4 kokoiset testipelilaudat 11-, 121- ja 1221-tilanteiden testauksessa.

ArrayListillä ja ArrayDequella on melkein 100% kattavuus pit-testaus tuloksissa. Testaus pääasiassa tietojen syöttämistä ja hakemista tietorakenteista eli onko niissä Ruudut listattuna mitä siellä pitäisikin olla.

## Algoritmin suorituskyky ja nopeus

Suorituskykyyn testausta varten on erikseen luokka nimeltä SuorituskykyTestaaaja, sillä graafinen käyttöliittymä veisi huomattavan suuren osan suoritusajasta suhteessa algoritmin suoritusajaan.

Ohessa tuloksia algoritmin nopeudesta pelilaudan kasvaessa. Aika on laskettu monen suoritusajon keskiarvona.

Pelilaudan koko	Aika
10 x 10	0.00041
20 x 20	0.00418
30 x 30	0.02312
40 x 40	0.07993
50 x 50	0.22079
70 x 70	0.82081
80 x 80	1.54572
100 x 100	3.62081
150 x 150	17.6208

Algoritmin  $O(n^3)$  aikavaativuuden vuoksi ei ole yllättävää että suoritus aika alkaa kasvamaan eksponentiaalisesti hyvin nopeasti. Jo 80x80 pelilaudalla laskuajat ovat reilusti yli sekunnin.

## Algoritmin todennäköisyys ratkaista peli kokonaan

SuorituskykyTestaaaja luo 100kpl pelilautoja joita ratkaisija ratkaisee niin pitkälle kuin pystyy ja lopuksi tulostaa kokonaan ratkaistujen pelilautojen määrän. Taulukossa näkyy paras ja huonoin prosenttiarvo ratkaistuista peleistä kyseisellä pelilaudan koolla jotka onnistuin saamaan monella eri ajokerralla.

Ratkaistujen pelien määrät vaihtelevat hyvin suuresti riippuen miten vaikeita pelilautoja randomisti sattuu tulemaan.

Pelilaudan koko	Ratkaistut pelit %
10 x 10	45 - 71
15 x 15	35 - 65
20 x 20	35 - 54
30 x 30	25 - 47
40 x 40	21 - 39
50 x 50	18 - 38
60 x 60	14 - 25
70 x 70	9 - 19

Taulukon arvoista on pääteltävissä että, mitä isompi pelilauta, sitä todennäköisempää on että pelilaudalle tulee tilanteita joita algortimi ei osaa ratkaista joten ratkaisutodennäköisyys vähenee lineaarisesti suhteessa pelilaudan kokoon. Käsien testatessa havaitsin että pelilaudan koosta riippuen pelilaudalle jää yleensä 1-3 ratkaisematonta tilannetta, joissa voisi päästä eteenpäin arvaamalla. Pahimmillaan noin 90% pelilaudasta voi jäädä ratkaisematta jos ensimmäinen klikkaus avaa mahdollisimman pienen neliönmuotoisen alueen, jonka ympäriltä ei saa avattua yhtään varmasti turvallista ruutua.

## Havaitut bugit

- Kaikk liputukset eivät näy jostain syystä aina heti, vaikka ratkaisijan tiedoissa miina on löydetty ja merkattu muistiin. Liputus tehdään siis samalla hetkellä kun miina löydetään.
- Jos viimeinen avaamaton ruutu on yksin nurkassa, ratkaisija ei osaa avata sitä.

## Käyttöliittymän testaus

Graafista käyttöliittymää on hyvin yksinkertaista testata. Siinä on kolme nappulaa: "Uusi peli", "Ratkaise siirto" ja "Ratkaise peliä". Uusi peli luo aina uuden pelin käyttöliittymään. Ratkaise siirto ratkaisee pelilaudalta ensimmäisenä ratkaisijan jonossa olevan siirron. Ratkaise peliä käynnistää

MiinaharavanRatkaisijan missä tahansa vaiheessa peliä ja ratkaisijan jäädessään jumiin pelaamista voi jatkaa itse ja käynnistää ratkaisijan jälleen uudestaan. RatkaisijanKomentaja välittää ratkaisijan jonoon laittamat siirrot käyttöliittymälle. Ratkaisijan tekemä liputtaminen on vain extra lisäinformaatiota ja havainnollistaa selkeämmin missä miinat ovat, mutta se ei ole välttämätöntä toiminnan kannalta.

Graafisen käyttöliittymän rajoitteena on, että sen piirtämien ruutujen koko on 24x24 näytöllä, joten korkein mahdollinen pelilauta yleisimmillä tietokoneen näytöillä on 40 ruutua.