

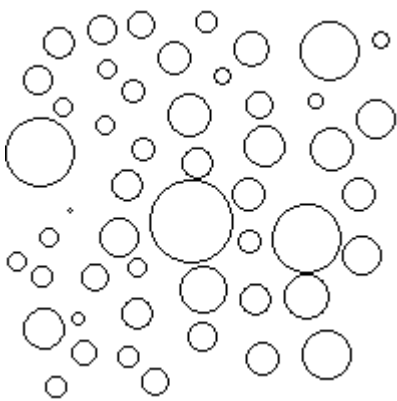
# Testausdokumentti

## Testaus

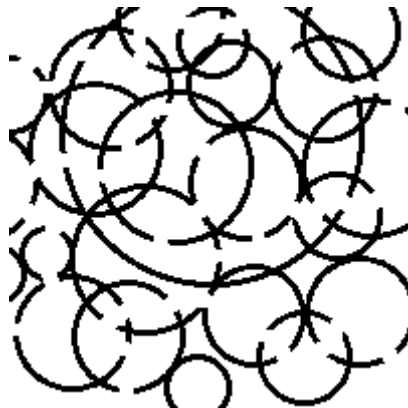
Testasin ohjelman eri algoritmien suoritusnopeuksia useilla syötekuvilla. Suoritin yhden algoritmin samalle syötteelle 10-300 kertaa syötteen koosta riippuen ja laskin keskiarvon siihen kuluneista ajoista. Toistin tämän joka algoritmille kaikilla syötekuvilla, ja lisäsin tulokset diagrammeihin.

## Käytetyt syötteen

Syöteinä testeissä käytettiin seuraavia kuvia:



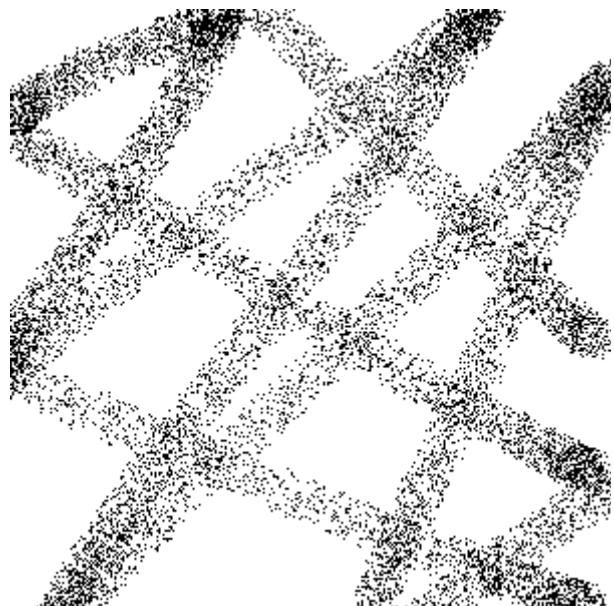
*circles.png*



*circles2.png*



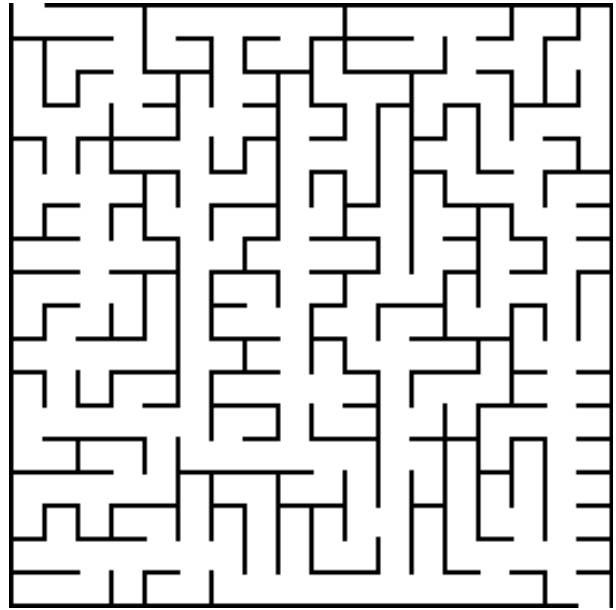
*lines.png*



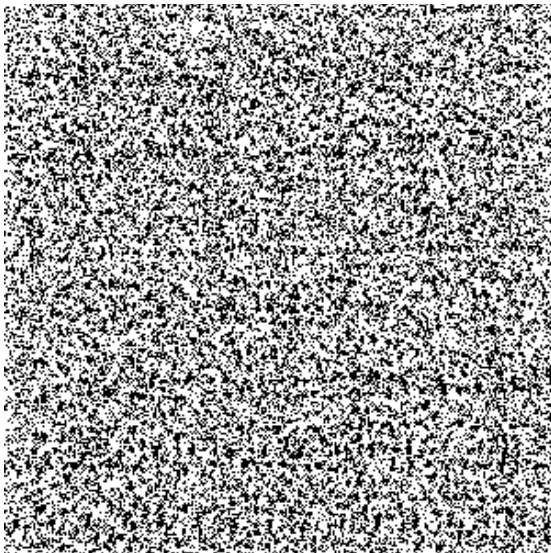
*spary.png*



*horizontal.png*



*maze.png*



*noise.png*

sekä kaksi muuta kuvaa, *no\_obstacles.png* ja *unsolvable.png*, joista ensimmäinen on täysin valkoinen ja kooltaan 600x100 px, ja toinen on täysin valkoinen 1500x1500 px, mutta sen oikean alakulman pikseli on ympäröity mustalla, joten reitinetsintä ei löydä sitä.

Jokaisella syötekuvalla aloituspisteeksi määriteltiin vasen yläkulma ja lopetuspisteeksi oikea alakulma, paitsi kuvalle *horizontal.png*, jolla aloituspiste oli (250, 20) ja lopetuspiste (250, 480) sekä kuvalle *no\_obstacles.png*, jolla aloituspiste oli (0, 50) ja lopetuspiste (599, 50).

# Testien toistaminen

Testien suorittamiseen käytettiin seuraavaa komentoa:

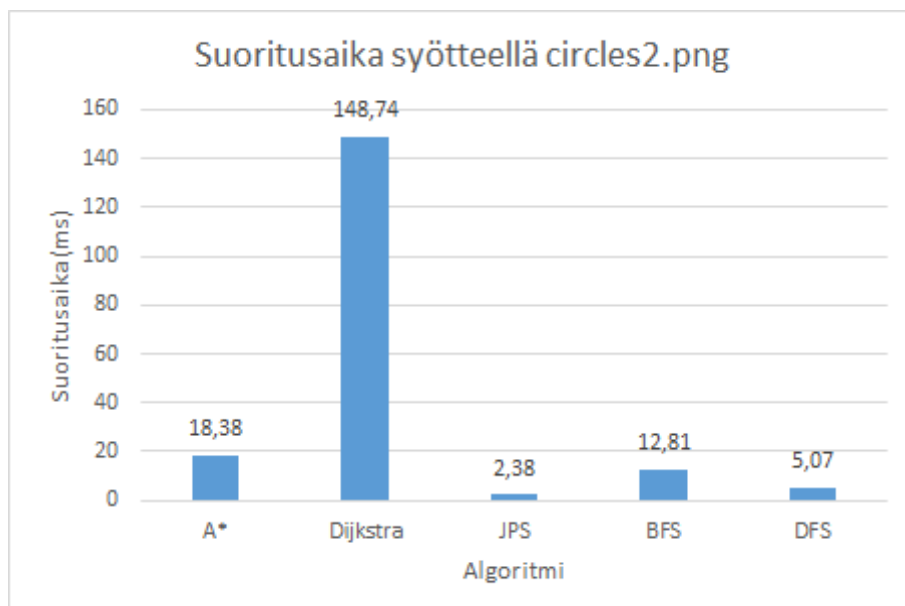
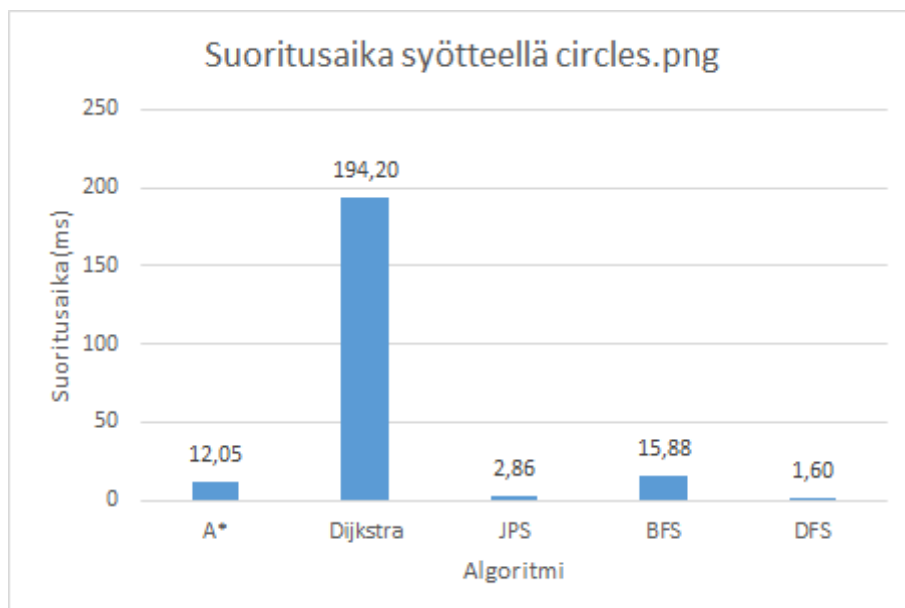
```
java -jar Tiralabra.jar -i syöte -s lähtö -g maali -a ALGORIMI -c toistot --less
```

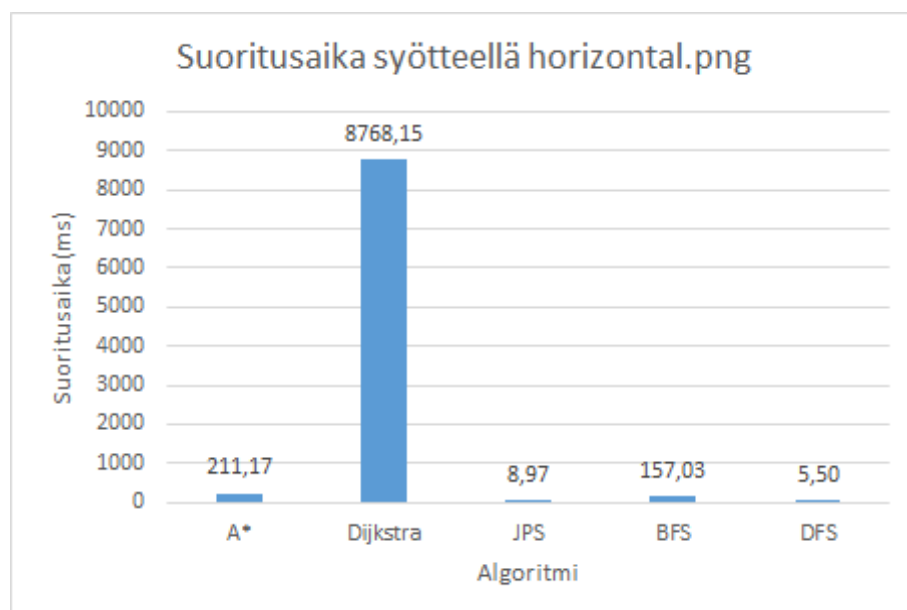
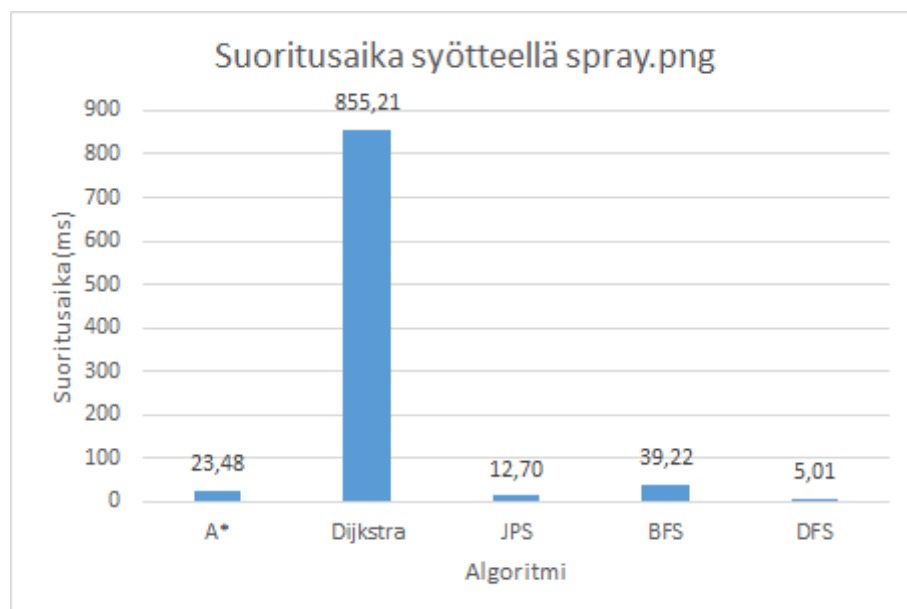
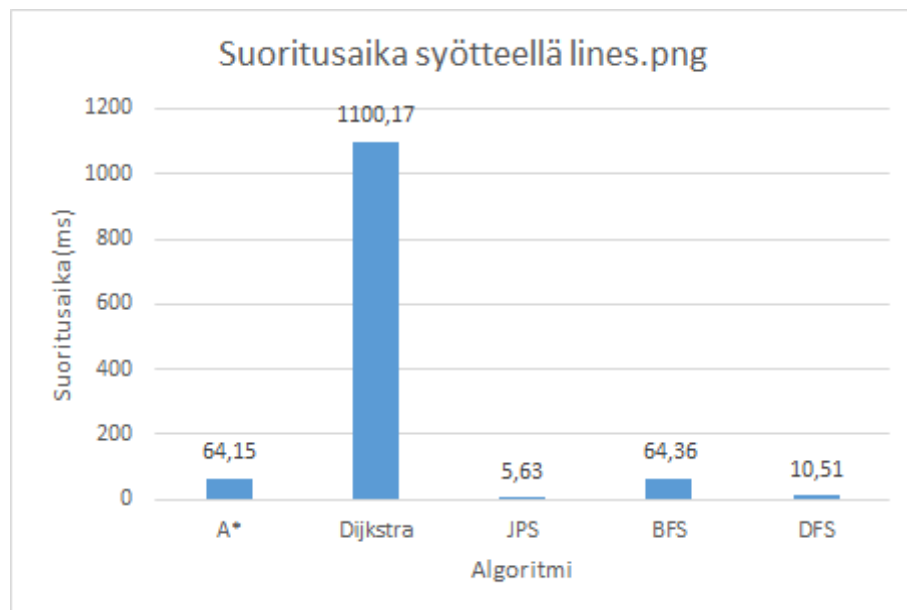
Eli esimerkiksi A\*:n ajamiseen circles2.png -kuvalla 100 kertaa vasemmasta yläkulmasta oikeaan alakulmaan tulostaen vain kuluneet ajat onnistuu komennolla:

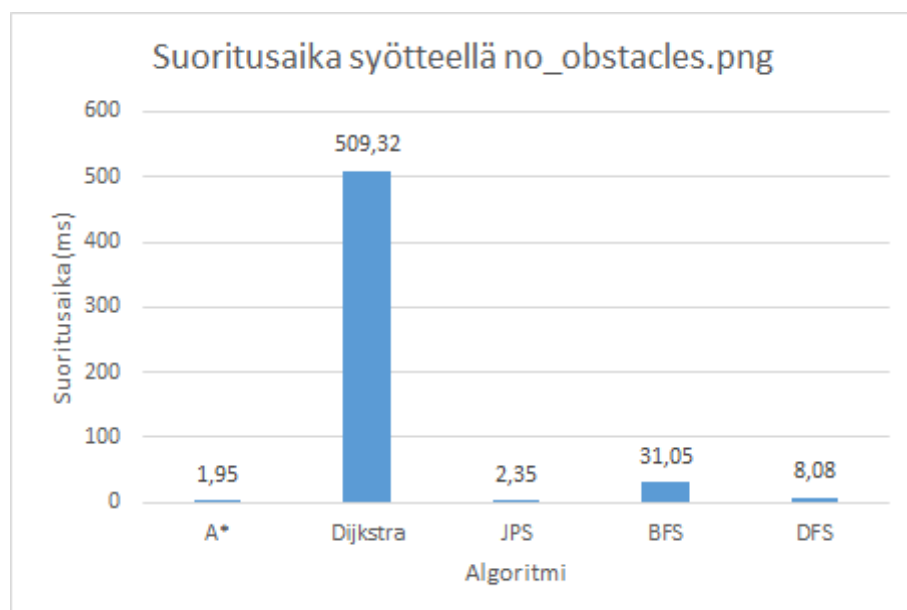
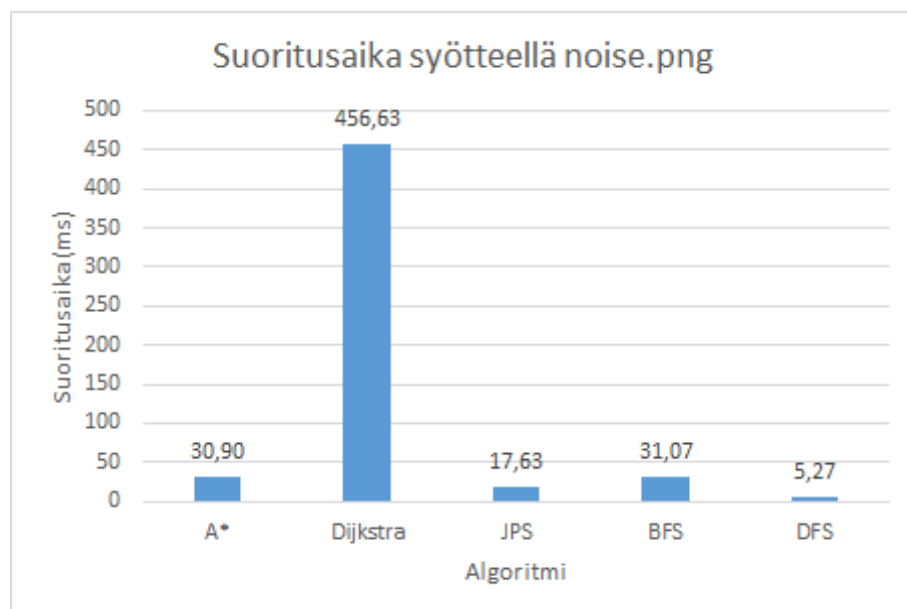
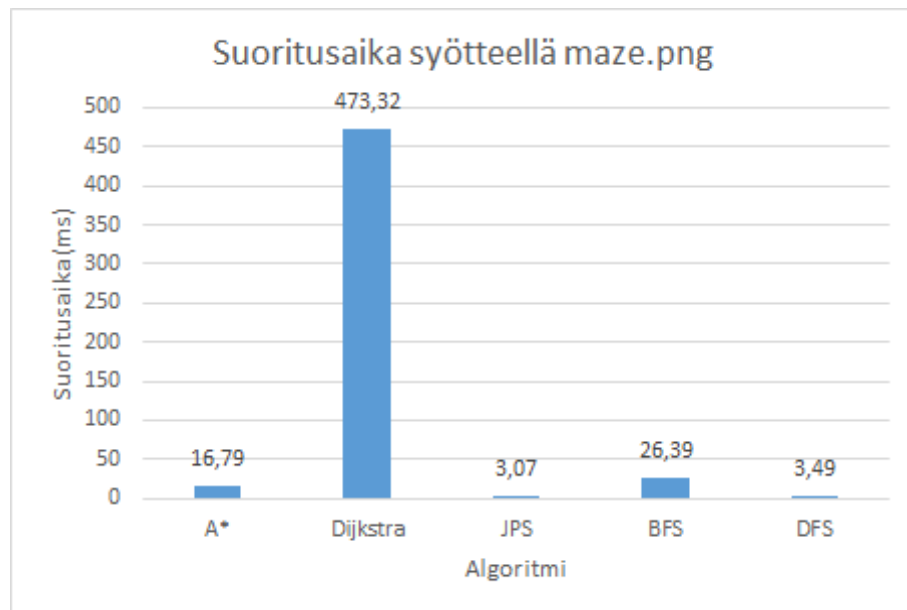
```
java -jar Tiralabra.jar -i circles2.png -s 0,0 -g 199,199 -a A_STAR -c 100 --less
```

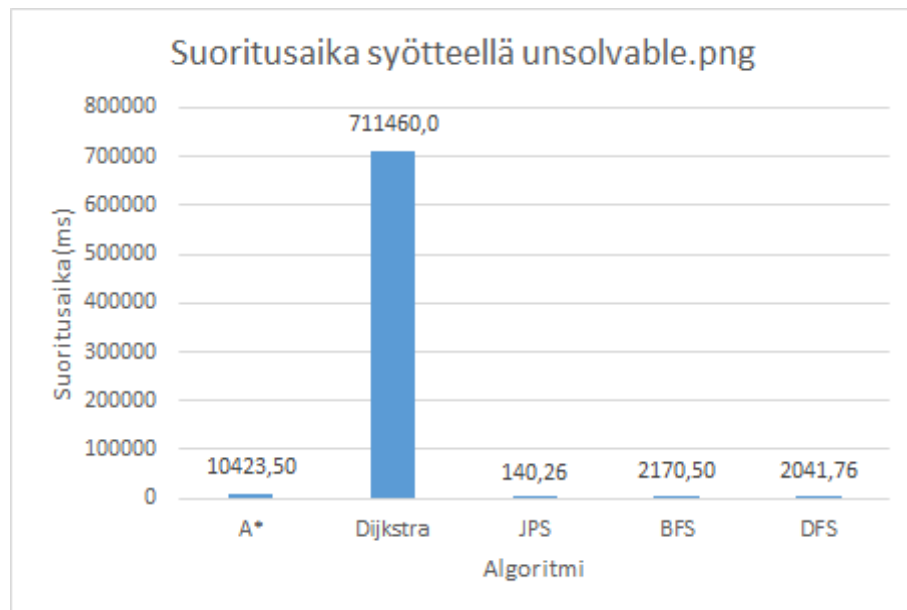
Lisätietoja komentorivioptioiden käytöstä löytyy käyttöohjeesta.

## Tulokset









[insert jotain analysointia datasta here]