Antti Kaikkonen

Testausdokumentti

Suorituskykytestaus

Testiolosuhteet:

Vertasin Dijkstran, A-tähden ja Bellman-Fordin suorituskykyjä niin että mittasin eri olosuhteissa kuinka monta kokonaista kertaa algoritmi ehti suoriutua 10 sekunnin aikana. Ruudukon esteet arvottiin uudelleen ennen joka suoritusta, mutta jokaisella ruudukolla ajettiin kukin algoritmi kerran. Laskennassa otettiin aikaa vain itse algoritmin suorituksesta, eikä esimerkiksi ruudukon uudelleen arpomisesta.

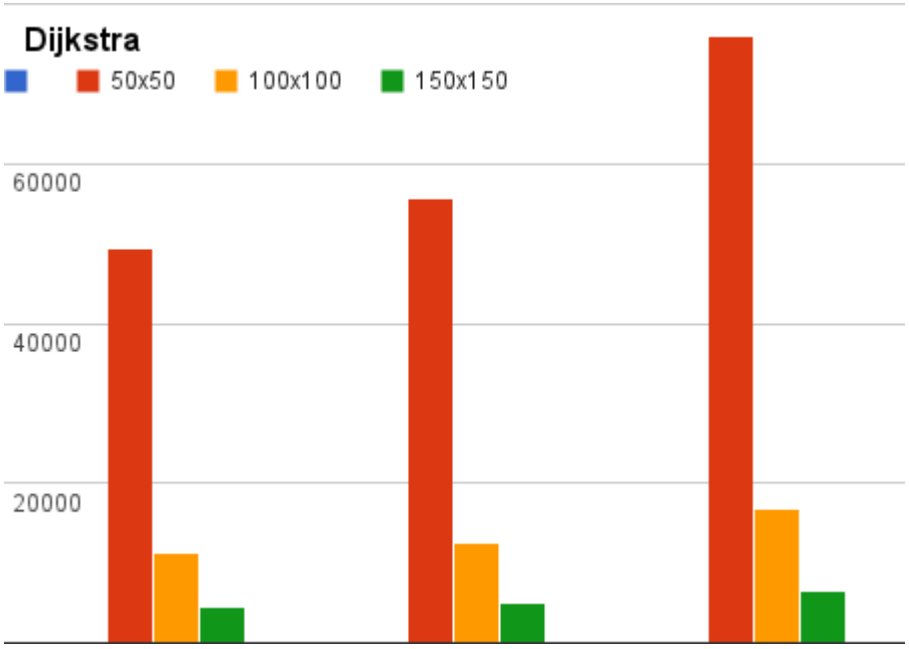
Algoritmin aloitus- ja maalisolmut asetettiin toistuvasti satunnaisesti lähelle ruudukon vastakkaiskulmia, kunnes molemmat olivat ruudussa joka ei ollut este. Mittauksessa ei ole hylätty testejä joissa reittiä lähdön ja maalin väliltä ole löytynyt. Testit on suoritettu tietokoneella jossa oli käytännön syistä myös muita ohjelmia samanaikaisesti käynnissä, joka saattaa aiheuttaa pieniä muutoksia tuloksissa. Testejä suorittaneen prosessorin kellotaajus oli 3.30GHz.

Diagrammien tulkitseminen

Suorituskykytestauksen diagrammeissa y-akselilla on algoritmien suorituskertojen määrä 10 sekunnin aikana. Korkea pylväs tarkoittaa siis, että algoritmi on toiminut nopeasti. X-akselilla puolestaan on eri tapauksia. Punaiset pylväät kuvaavat 50x50 ruudukossa tehtyjä testejä, keltaiset 100x100 ruudukossa tehtyjä testejä ja vihreät 150x150 ruudukossa tehtyjä testejä.

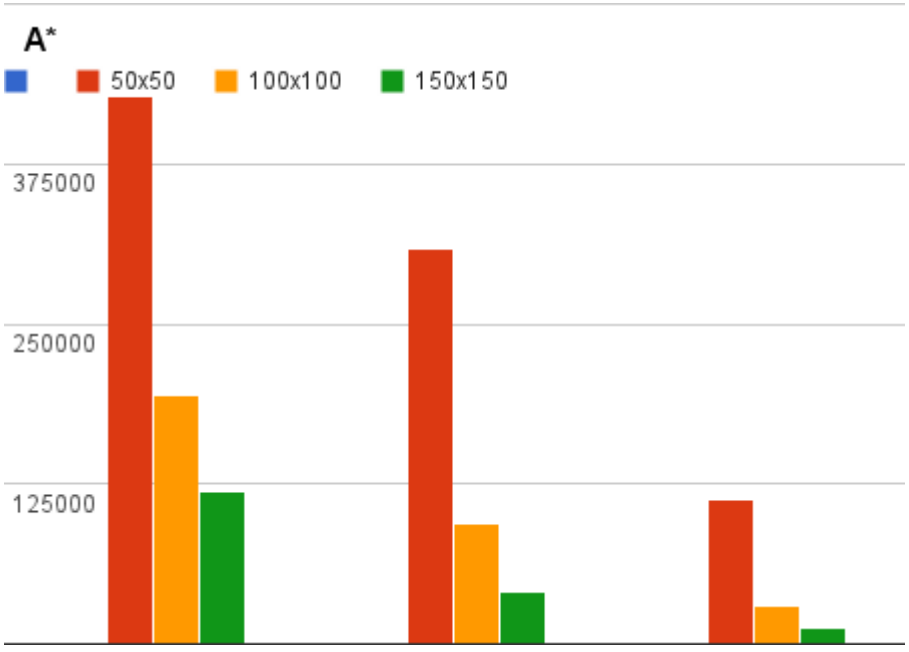
Vasemmanpuoleinen kolmen pylvään ryhmä kuvaa testejä joissa ruutuihin on arvottu esteet 0% todennäköisyydellä. Vastaavasti keskimmäisessä ryhmässä esteet on arvottu 15% todennäköisyydellä ja oikeanpuoleisessa 30% todennäköisyydellä. On suositeltavaa ottaa huomioon että 30% todennäköisyydellä varsin monessa tapauksessa reittiä lähdön ja maalin välillä ei ole.

Dijkstra



Kuten diagrammista huomaamme, niin Dijksta:n aikavaativuus kasvaa melko jyrkästi syötteen kasvaessa. Diagrammi vaikuttaa uskottavalta, koska ruudukon leveyden ja korkeuden kaksinkertaistuessa sen pinta-ala, eli syötteen koko nelinkertaistuu. Jos nyt tekisin testauksen uudestaan niin valitsisin kyllä pienempiä eroja syötteen koossa.

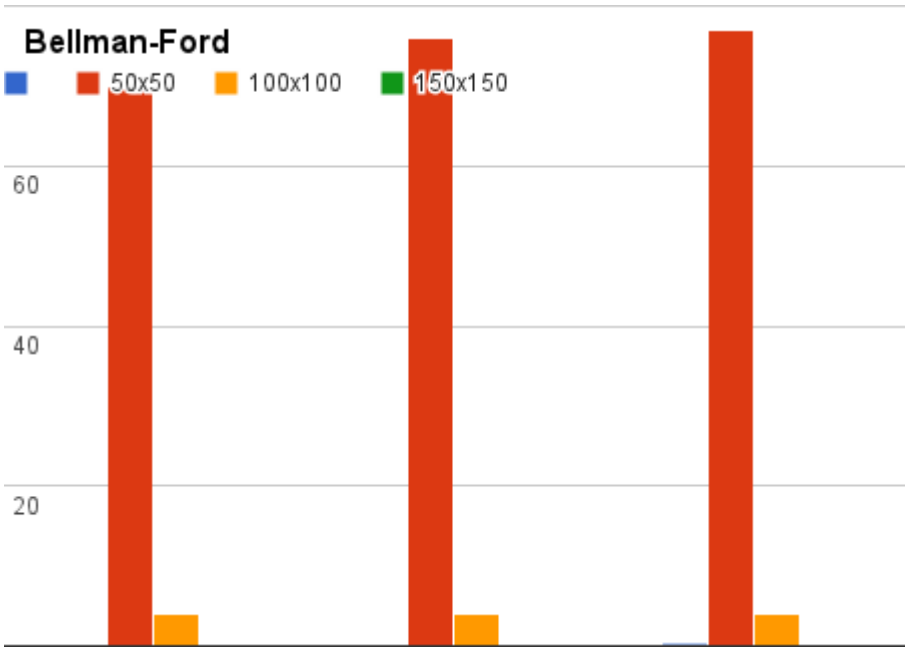
A-tähti



A-tähti algoritmissa aikavaativuus kasvaa syötteen kokoon nähden hitaammin kuin Dijkstan algoritmissa. Huomaamme myös että kasvattaessamme esteiden määrää algoritmin suoritus

hidastuu, toisin kuin Dijkstran algoritmissa jossa se nopeutuu. A-tähti on siis optimaalinen pienellä määrällä esteitä.

Bellman-Ford



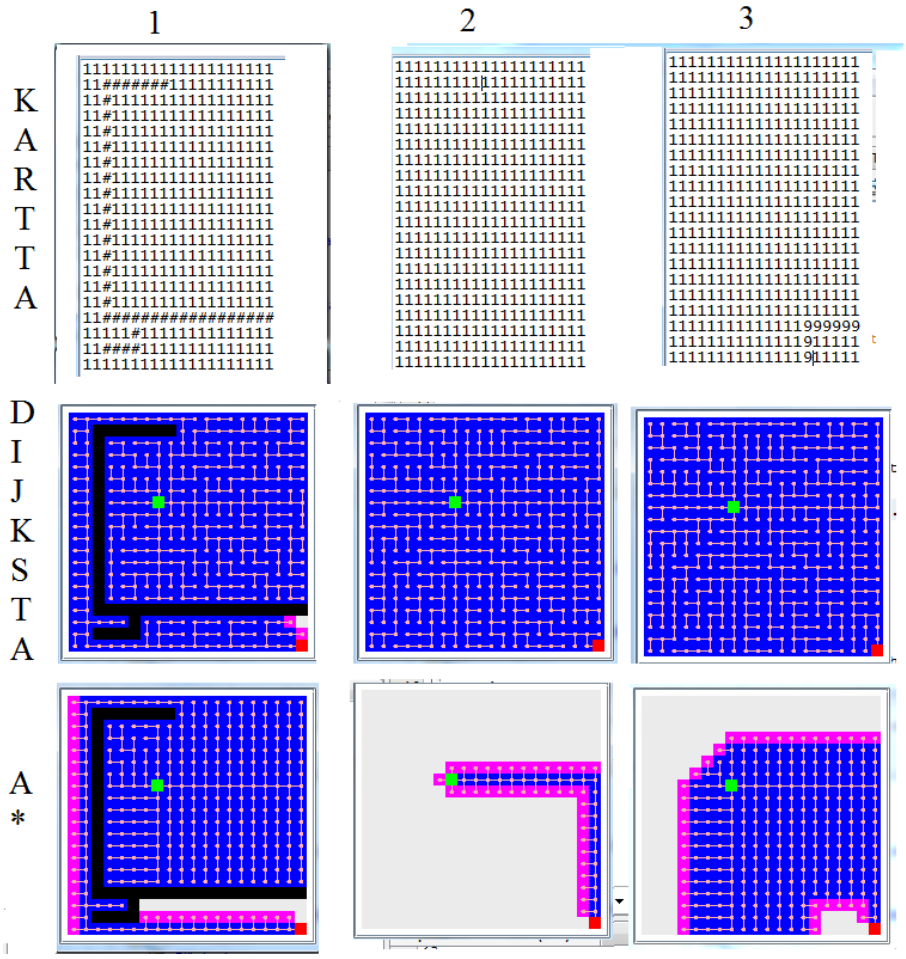
Bellman-Ford on näistä kolmesta hitain, koska sen pitää toimia myös negatiivisilla kaarenpainoilla jos verkosta ei löydy negatiivista sykliä. Diagrammeista huomataan että Bellman-Ford ei ehdi tehdä yhtäkään kokonaista suoritusta 150x150 ruudukossa kymmenen sekunnin aikana. Lisäksi aikavaativuus kasvaa neliöllisesti jolloin pienetkin muutokset syötteen koossa aiheuttavat melko suuria muutoksia aikavaativuudessa.

Yksikkötestaus (JUnit)

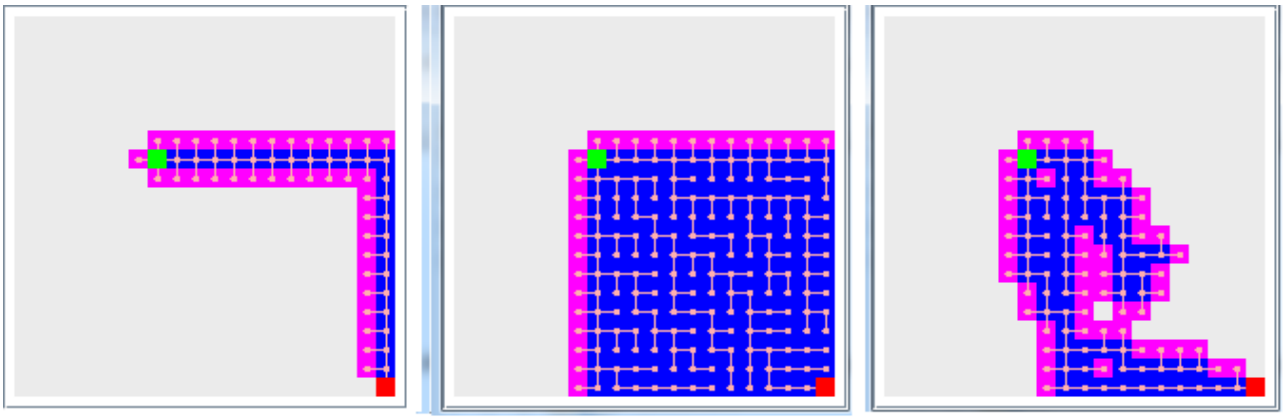
Testasin yksikkötesteillä asioita joista on helppo tietää oikea lopputulos. Testasin muun muassa binäärikeon ja dynaamisen taulukon toimintoja melko laajamittaisesti sekä esimerkiksi algoritmien toimimista sikäli että Dijkstra, A* ja Bellman-Ford kaikki löytävät yhtä lyhyen reitin samassa verkossa.

Visuaalinen testaus

Tein heti alkuvaiheessa työhön graafisen käyttöliittymän joka pystyy mm. animoimaan algoritmin etenemistä. Tästä oli paljon apua sillä monet virheet olisivat muuten voineet jäädä huomaamatta.



Esimerkki visuaalisesta testaamisesta. Siniset ruudut ovat käsiteltyjä ruutuja ja magentan väriset käsittelyssä olevia. Vihreä kuvaa lähtöä ja punainen maalia.



Toinen esimerkki visuaalisesta testaamisesta. Kuvassa A* algoritmi on ajettu erilaisilla tavoilla priorisoida saman g arvon saavia ruutuja. Kaikki toimivat mutta toiset ovat keskimäärin muita nopeampia.