Testausdokumentti

Mitä on testattu ja miten
Olen testannut toteutettavia tietorakenteita (jono, lista, pino ja solmu), sekä algoritmeja, jotka näitä hyödyntävät (BFS, DFS ja Dijkstra). Jotain testejä olen tehnyt empiirisesti, kuten satunnaiset verkon konfiguraatiot ja testannut reitin löydettävyyden suoraan ajamalla ohjelman. Alla kerron automatisoitujen testien toteutuksesta ja tämän dokumentin loppuun laitan empiiristen testien tuloksia kuvina. Lopussa myös eri algoritmien hakujen viemät ajat taulukossa.

Jono (Queue)
Jonon testaus on toteutettu kokeilemalla jonoon lisäämistä ja poistamista siten, että poistettujen ja lisättyjen järjestystä on vertailtu keskenään.

Lista (Array)
Listan toimintaa on testattu lisäämällä listaan alkioita ja vertaamalla listan indeksien sisältämiä olioita siihen, missä järjestyksessä listaan on tavaraa lisätty. (esim. Indeksissä 0 pitäisi löytyä ensimmäisenä lisätty alkio ja indeksistä 2 tulisi löytyä kolmantena lisätty alkio). Listan testauksessa on myös huomioitu duplikaattien lisääminen, mitä ei saa tapahtua.

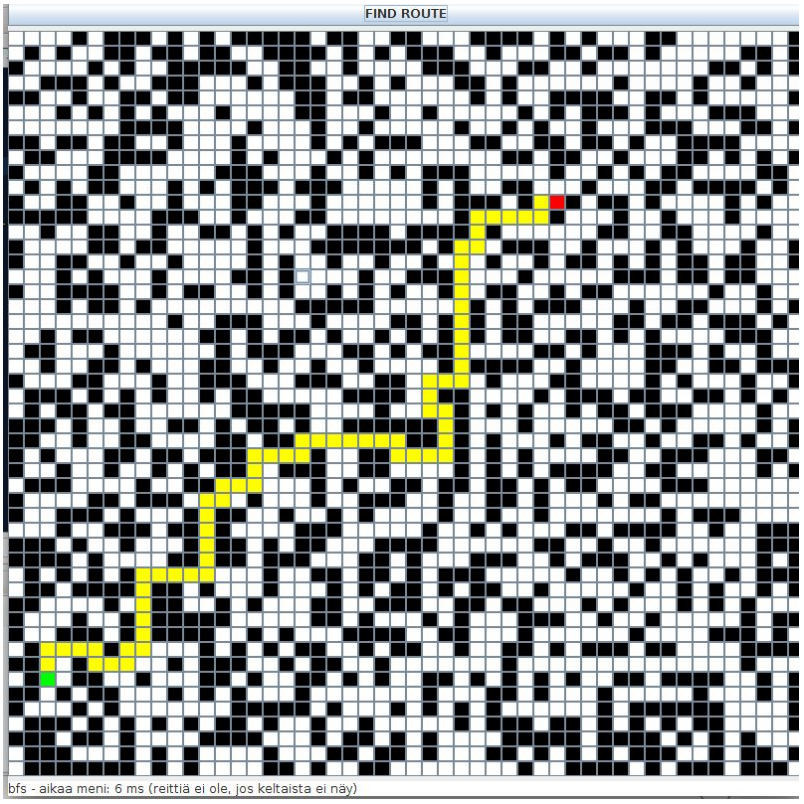
Pino (Heap)
Pinon (tarkemmin sanottuna minimipinon) testauksessa on tarkistettu, että lisätyt solmut löytyvät oikeista indekseistään pinon taulukosta. Solmuilla on eri painoja ja nämä painot vaikuttavat niiden paikkaan pinossa. Tämän järjestyksen täytyy pysyä oikeana myös, kun pinosta poistetaan alkioita.

Solmu (Vertex)
Solmujen testauksessa on katsottu, että konstruktori asettaa arvot oikein. Lisäksi testeissä on tarkistettu, että solmujen yhdistäminen toimii oikein ja solmujen naapurit löytyvät näiden naapurilistalta.

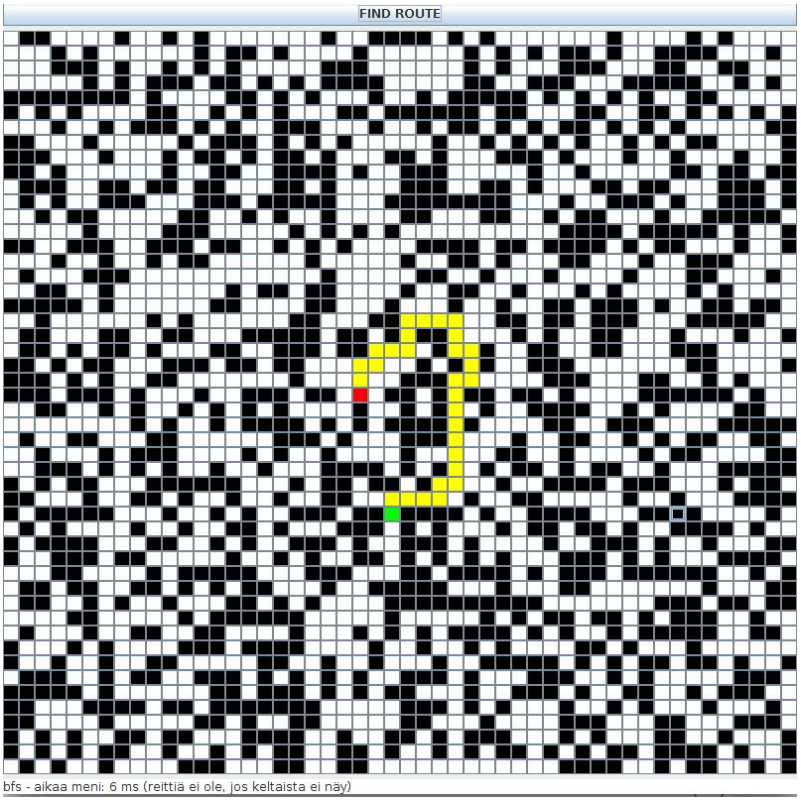
BFS, **DFS ja Dijkstra**
Reittialgoritmien testauksessa on testattu, että algoritmit löytävät reitin verkossa parissa eri skenaariossa (aloitus- ja lopetuspisteet ovat suhteellisen lähellä toisiaan, siten, että näiden välissä on vain yksi este ja siten, että oikean reitin on kuljettava tietyn solmun kautta, koska muuta reittiä ei ole). Testaamisessa on myös tarkistettu, etteivät algoritmit löydä mahdotonta reittiä, kun reittiä ei ole esteiden takia.

Empiiriset testit

BFS

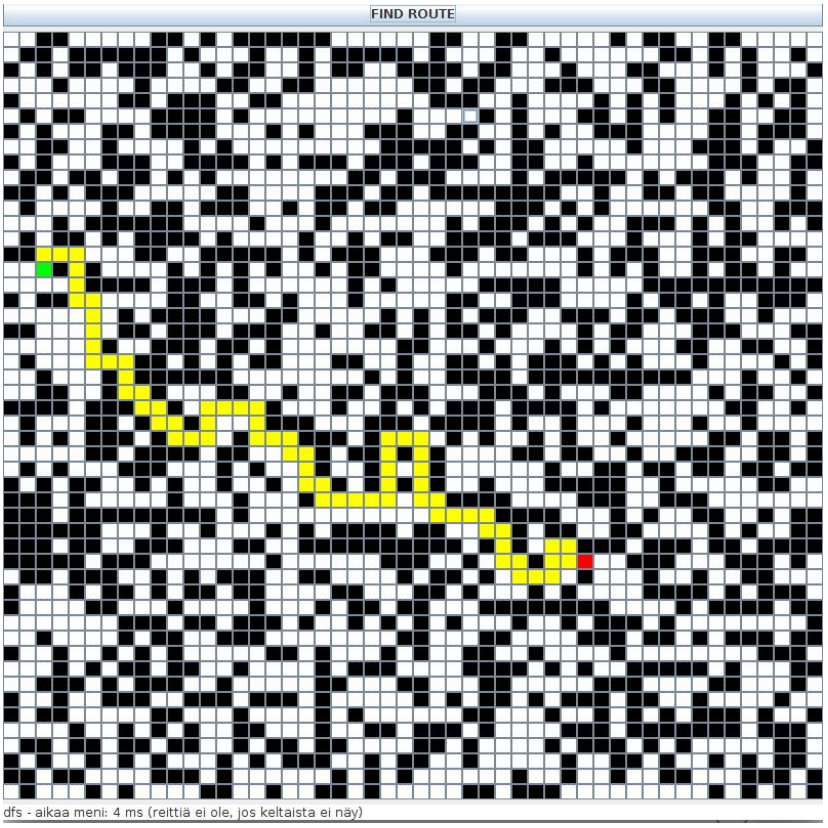


Kuva satunnaisesta BFS-algoritmin reittihausta



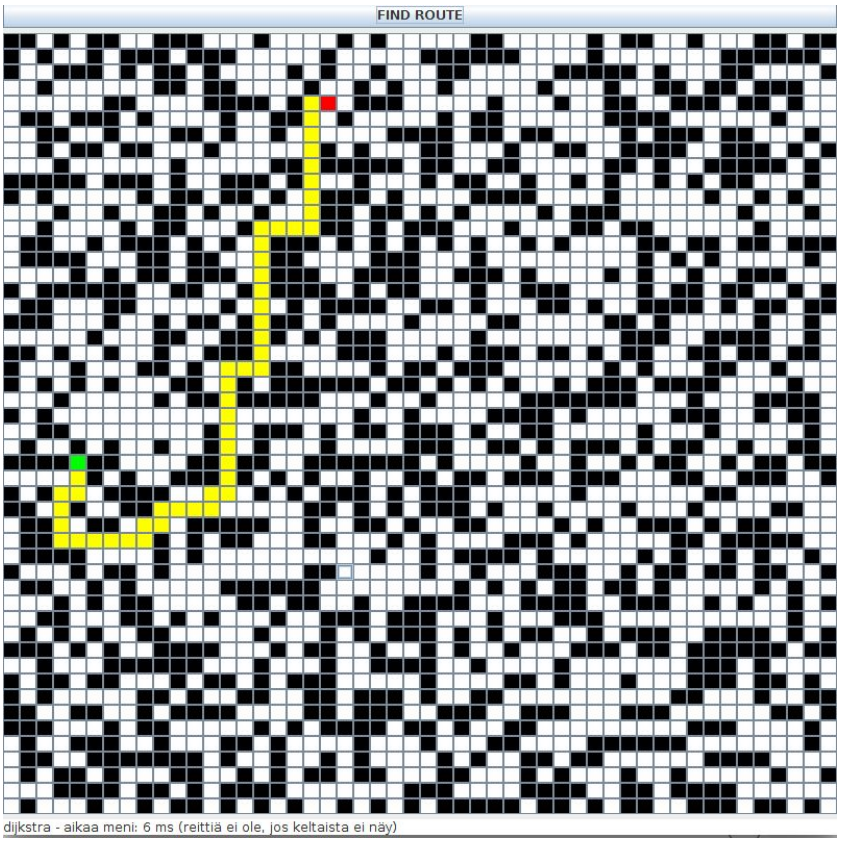
Kuva toisesta satunnaisesta BFS-algoritmin reittihausta

DFS

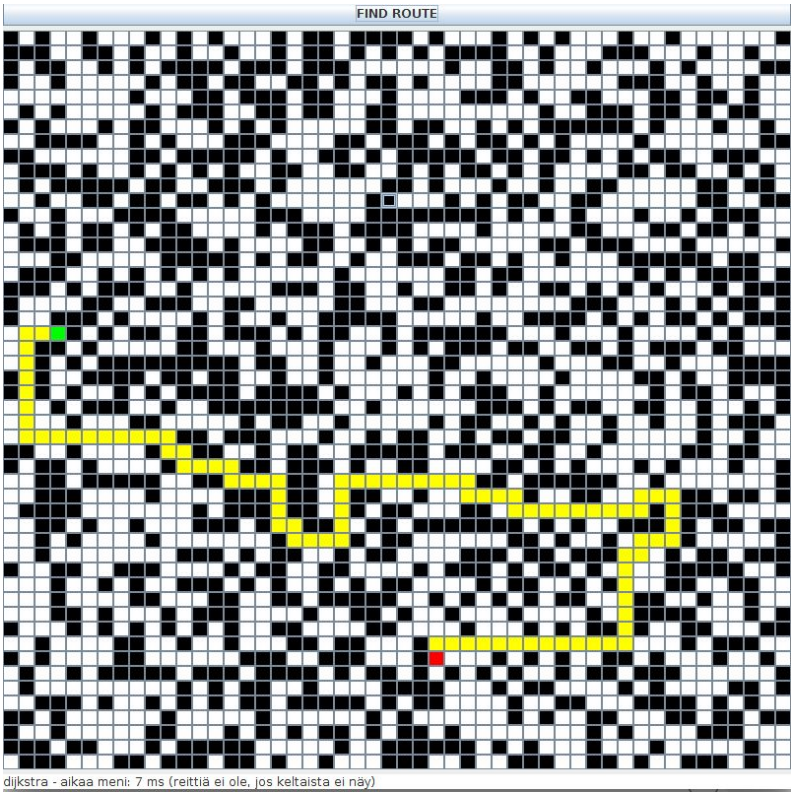


Kuva satunnaisesta DFS-algoritmin reittihausta

Kuva toisesta satunnaisesta DFS-algoritmin reittihausta



Kuva satunnaisesta Dijkstra-algoritmin reittihausta



Kuva toisesta satunnaisesta Dijkstran-algoritmin reittihausta

Eri aikoja millisekunteina algoritmien reittien hauille, kun verkko on jokaiselle algoritmille sama jokaisella kierroksella ja verkko muuttuu kierrosten välissä.

Alg.	**1.**	**2.**	**3.**	**4.**	**5.**	**6.**	**7.**	**8.**	**9.**	**10.**
BFS	5	7	5	3	5	6	2	5	2	4
DFS	4	3	4	2	4	5	3	4	3	4
Dijks.	8	3	4	3	9	6	7	4	4	3

BFS aikojen keskiarvo: **4,4 ms**
DFS aikojen keskiarvo: **3,6 ms**
Dijkstran aikojen keskiarvo: **5,1 ms**

Otin algoritmien reittihauille useampiakin aikamittauksia ja tulokset ovat aina olleet saman kaltaisia yllä olevien kanssa; Dijkstra on yleensä hitain, BFS on vähän nopeampi ja DFS on nopein.