

Määrittelydokumentti

Toteutan työssäni Dijkstran algoritmin ja A* algoritmin, sekä niihin liittyvät tietorakenteet. Dijkstran algoritmin avulla etsitään lyhin mahdollinen reitti pisteestä x kaikkiin muihin pisteisiin suunnatussa verkossa. Pahimmillaan algoritmi vie aikaa $O(|V| + |E| \log |V|)$. Algoritmin avuksi toteutan ainakin yhden minimikekotyyppin, pinon ja listan. Dijkstran lisäksi toteutan A* algoritmin, jolla myös etsitään lyhintä polkua, mutta painojen lisäksi otetaan huomioon jäljellä oleva matkan.

Ongelma ja valitut tietorakenteet

Ongelmana on labyrintin ratkominen kuvasta. Työkaluksi ongelman ratkomiseen valitsin Dijkstran algoritmin, ja muutaman tietorakenteen. Tärkein tietorakenne on minimikeko, jota tarvitsen toteuttaessani Dijkstraa tehokkaasti. Toteutan myös pinon, koska tarvitsen sitä ellen halua erikseen kääntää reittilistaa. Ongelmaa voisi lähteä ratkaisemaan myös muilla algoritmeilla, mutta valitsin Dijkstran, koska saatan haluta lisätä myöhemmin muitakin esteitä kun seinät. Esim vesi, joka vaan lisää nodejen välistä painoa. A Star on tunnetuin reitinetsintä algoritmi ja sillä pitäisi saada tulokset erittäin nopeasti. Tämän takia ajattelin että myös se on hyvä toteuttaa, vaikka ongelman saakin jo pelkällä Dijkstralla ratkaistua.

Ohjelman syöte

Lisäksi toteutan listan jossa voin säilyttää nodejen naapurinodeja. Ohjelma saa syötteenä kuvatiedoston, joka esittää labyrinttiä. Kuvatiedostossa seinät on merkattu mustalla, lähtöpiste punaisella ja maali sinisellä. Kuva muunnetaan kaksiulotteiseksi taulukoksi, joka sisältää nodeja. Node on olio, joka tietää sijaintinsa, painonsa ja naapurinodensa. Lopuksi kun reitti on saatu selville, piirretään se labyrinttiin.

Tavoitteenani on saada Dijkstra toimimaan sen tavallisen aikavaatimuksen, eli pahimmillaan $O(|V| + |E| \log |V|)$ nopeudella.

Lähteet:

<http://www.cs.helsinki.fi/u/floreen/tira2013syksy/tira.pdf> Helsingin yliopiston tietorakenteet ja algoritmit kurssin materiaali.

<http://sange.fi/~atehwa/cgi-bin/piki.cgi/a-t%E4hti-algoritmi>

http://en.wikipedia.org/wiki/A*_search_algorithm