

**1) Mitä algoritmeja ja tietorakenteita toteutat työssäsi:**

Aion toteuttaa Dijkstran algoritmin, A\*-algoritmin sekä ajan puitteissa myös Bellman-Ford algoritmin. Työssä tulen luultavasti käyttämään ainakin pino ja minimikeko tietorakenteita.

**2) Mitä ongelmaa ratkaiset ja miksi valitsit kyseiset algoritmit/tietorakenteet:**

Tarkoituksena on kehittää ohjelma jolla pystyy vertailemaan eri reitinhaku algoritmeja erilaisissa verkoissa. Ohjelma etsii lyhyimmän reitin verkon kahden solmun välillä.

Valitsin aiheen koska haluan oppia toteuttamaan lyhimmän polun etsiviä algoritmeja käytännössä. Reitinhauille löytyy myös monia käytännön sovelluksia kuten esimerkiksi reitinhakuohjelmat ja navigaattorit.

**3) Mitä syötteitä ohjelma saa ja miten näitä käytetään:**

Ohjelmalle annetaan syötteeksi kaksiulotteinen taulukko jonka se lukee tekstitiedostosta. Tekstitiedosto koostuu merkeistä jotka ovat joko esteitä (#) tai vapaata(.). Liikkuminen toimii niin että yhdestä ruudusta voi liikkua kaikkiin sen vapaisiin vierusruutuihin eli heuristiikan mukaan joko neljään tai kahdeksaan ruutuun, jos diagonaalinen liikkuminen sallitaan. Aloitusruutu ja maaliruutu annetaan ohjelmalle koordinaatteina.

**4) Tavoitteena olevat aika- ja tilavaativuudet (m.m. O-analyysi):**

E = Kaarten joukko. V = Solmujen joukko.

Aika:

A\*:  $O((|E|+|V|) * \log |V|)$

Dijkstra:  $O((|E|+|V|) * \log |V|)$

Bellman-Ford:  $O(|V| |E|)$

Tila:

A\*:  $O(|V|)$

Dijkstra:  $O(|V|)$

Bellman-Ford:  $O(|V|)$

**5) Lähteet:**

Patrik Floréen kevät 2012 luentomateriaali.

[http://en.wikipedia.org/wiki/A\\*\\_search\\_algorithm](http://en.wikipedia.org/wiki/A*_search_algorithm)

[http://en.wikipedia.org/wiki/Dijkstra\\_algorithm](http://en.wikipedia.org/wiki/Dijkstra_algorithm)

[http://en.wikipedia.org/wiki/Bellman\\_Ford\\_Algorithm](http://en.wikipedia.org/wiki/Bellman_Ford_Algorithm)