

Määrittelydokumentti

Toteutetaan sovellus jolla on mahdollista vertailla Dijkstran –algoritmin ja A* -algoritmin suorituskykyä.

Toteutettavat tietorakenteet & algoritmit:

- minimikeko (tarvitaan kummassakin algoritmissa pitämään läpikäytävät solut järjestyksessä)
- A* -algoritmi
- Dijkstran algoritmi (toteutetaan A* -algoritmin erikoistapauksena, jossa heuristiikan laskema etäisyys on aina nolla.)

Syötteet:

Ohjelmalle syötetään tekstimuotoisia verkkoja:

```
.....  
.X.....  
.....##  
.....##  
.....O..  
.....
```

. = tyhjä tila verkossa
X = haun lähtöpiste
O = haun päätepiste
= este

Lisäksi käyttäjän on voitava valita käytettävä algoritmi, sekä mikäli valittu algoritmi on A*, myös käytettävä heuristiikka on oltava mahdollista antaa.

Ohjelmaa voitava ajaa myös 'batch' moodissa vertailuja varten, jolloin sovellus generoi satunnaisia verkkoja, käy ne läpi kummallakin algoritmilla ja raportoi tuloksista. Tällöin parametrina täytyy voida antaa kuinka monta verkkoa generoidaan.

Ennakko-odotukset:

Koska reitinhaku on toteuttajalle uusi asia, ei ennakko-odotuksia juurikaan ole. A* -algoritmin pitäisi olla tehokkaampi kuin Dijkstran, varsinkin kun läpikäytävän verkon koko kasvaa. Lisäksi A* -heuristiikkojen eroista ei ole selkeää ennakko-odotusta, mutta oletettavasti Euklidisella heuristiikalla päästään tarkempiin tuloksiin kuin Manhattan –heuristiikalla. Manhattan –heuristiikan etu on lähtötilanteessa tuntematon.

Lähteet:

Cormen

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

http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

<http://memoization.com/2008/11/30/a-star-algorithm-in-java/>

http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Clustering_Parameters/Manhattan_Distance_Metric.htm

<http://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html>

<http://www.policyalmanac.org/games/aStarTutorial.htm>