

Määrittelydokumentti

Työssäni käytän luokkia Dijkstra (algoritmi, jolla haetaan lyhin reitti ulos labyrintistä), Solmu (yksikkö, joista labyrintti koostuu, voi olla tyhjä tai täytetty), Taulukko (sisältää monta solmua), Keko (apuna Dijkstran toteutuksessa), Kekosolmu (apuna keon toteutuksessa), Bellman-Ford (toinen algoritmi, joka löytää labyrintista ulos) sekä pääluokkaa Labyrintti.

Tarkoitus on löytää kaksi-ulotteisesta labyrintistä lyhin ulospääsy Dijkstran sekä Bellman-Fordin algoritmeilla ja vertailla saatuja tuloksia. Dijkstra toimii ajassa $O((|V| + |E|)\log|V|)$ eli tässä tiheän verkon tapauksessa tapauksessa $O(n^2)$. Syötteenä algoritmi saa siis taulukon, joka koostuu Solmu-olioista sekä lähtöpisteestä. Labyrintin koko on solmujen määrä eikä Dijkstra sen enempää tilaa vaadi, algoritmin tilavaativuus on $O(n)$. Labyrintti on toteutettu ascii-merkeillä, Y viittaa pääsyyn ja E esteeseen. Myös Bellman-Ford toimii ajassa $O(|V||E|)$ eli tiheässä verkossa $O(n^2)$ ja vaatii tilaa myös sen $O(n)$.

Lähteet:

TiRan kurssimateriaali

<http://codereview.stackexchange.com/questions/71536/minheap-implementation-in-java> (5.8.2016)