

Tiralabra määrittelydokumentti

Harjoitustyöni aiheena on toteuttaa kaksi reitinhakualgoritmia: Leveyssuuntainen haku, sekä a-tähti algoritmi. Nämä algoritmithan löytävät lyhimmän polun annetusta verkon lähtösolmusta annettuun maalisolmuun. Tavoitteena on toteuttaa algoritmit niin, että ne toimivat oikein (löytävät lyhimmän polun, jos sellainen on olemassa, sekä käyvät algoritmia ajettaessa juuri niissä solmuissa, joissa niiden algoritmien perusidean mukaan kuuluukin käydä), sekä tehokkaasti (tässä keskitytään erityisesti aikavaatimukseen). Toteutukseen kuuluu myös yksinkertainen graafinen käyttöliittymä, jonka avulla voidaan nähdä että algoritmi löytää oikean polun tietynlaisessa verkossa.

Algoritmien toteutuksen lisäksi toteutetaan niiden käyttämät tietorakenteet (hajautustaulu, minimikeko, pino, linkitetty lista sekä arraylist). Nämä tehdään käyttäen pitkälti samaa nimeämistä kuin javan valmiissa kirjastoissa. Tietorakenteet toteutetaan kuitenkin nimenomaan toteutettavien reitinhakualgoritmien tarpeita silmälläpitäen, joten niihin ei välttämättä toteuteta kaikkia javan työkalujen sisältämiä ominaisuuksia ja toisaalta niihin voidaan toteuttaa joitain hyödylliseksi havaittuja ominaisuuksia joita javan valmiista työkaluista ei löydy.

Tavoitteena on siis päästä algoritmien tunnettuihin aikavaativuuksiin, jotka ovat leveyssuuntaiselle haulle $O(|V| + |E|)$ ja a-tähdelle $O((|V| + |E|) \log |V|)$ (ks. esim <http://www.cs.helsinki.fi/u/floreen/tira2012/tira.pdf>). Tässä $|V|$ siis tarkoittaa verkossa olevien solmujen määrää ja $|E|$ kaarien määrää. Myös tilavaativuuksiin otetaan jonkin verran kantaa. Tilavaativuus BFS algoritmille on: $O(|V|)$, sillä pahimmassa tapauksessa aloitussolmusta on kaari kaikkiin muihin solmuihin. Omassa toteutuksessani kaaria on jokaisesta solmusta korkeintaan 4 kappaletta, mutta algoritmi käyttää kuitenkin aputaulukkoja, joiden tilavaativuus on $O(|V|)$. Myös a-tähti algoritmilla avoimet listan pahimman tapauksen koko on verrannollinen verkossa olevien solmujen määrään, joten senkin tilavaativuus on $O(|V|)$. Oma toteutukseni käyttää lisäksi aputaulukkoja jotka ovat myös $O(|V|)$.