

Toteutusdokumentti

Ohjelman Verkko luo verkko olion kaksiluokitteisesta kokonaislukutaulukosta. Taulukon arvot vastaavat solmun painoja ja indeksit solmun koordinaatteja. Solmun paino (vähintään 1) kertoo kuinka suuri kustannus solmuun siirtymiselle on. Verkossa voi liikkua vasemmalle tai oikealle, tai alas ja ylös.

Prioriteettijono luokka vastaa prioriteettijonon toiminnasta, joka on taulukkona toteutettu minimikekoja se sisältää verkon solmuja, niin, että minimikeon ensimmäinen alkio on aina se solmu n , jonka sen hetkinen arvo $f(n)=g(n)+h(n)$ on pienin, eli jonka pienin tunnettu kustannus lähtösolmusta plus heuristinen arvio kyseisestä solmusta maalisolmuun on pienin.

Astar luokka vastaa itse haun toiminnasta. Haku palauttaa aina keon pienimmän alkion, kunnes maalisolmu löytyy tai keko on tyhjä. Jos tutkittava solmu ei ole maalisolmu, algoritmi hakee solmun naapurisolmut, ja käy ne läpi, jos polku tutkittavasta solmusta naapurisolmuun on edullisempi kuin ennestään edullisin polku lähtösolmusta naapurisolmuun, päivitetetään naapurisolmun edullisimman polun kustannus lähtösolmusta prioriteettijonossa.

Aikavaativuus on $O((|V|+|E|)\log|V|)$, jossa V on verkon solmujen joukko ja E kaarten joukko. Haku käy pahimmillaan jokaisen solmun ja sen naapurit kertaalleen läpi, päivittäen minimikekoa korkeintaan jokaisen kaaren verran ($\log|V|$ operaatio).

Ohjelman tilavaativuus on $|V|$, ohjelma tallentaa jokaisen verkon solmun kertaalleen minimikekoon ja taulukkona toteutettuun verkko-olioon.

Ohjelman Heuristiikka luokka vastaa solmun etäisyysarvion maalisolmuun laskemisesta. Heuristiikka luokassa on kokonaisluku heuristiikka attribuutti, joka kertoo mitä heuristiikkaa käytetään, 0:lla etäisyysarvio lasketaan solmujen välisenä Manhattan etäisyytenä, 1:llä lasketaan Euklidinen etäisyys, muilla arvoilla etäisyysarvio on aina 0, jolloin algoritmi toimii kuten Dijkstra.

Haettaessa solmun naapurisolmuja käytetään solmujen tallentamiseen linkitettyä listaa.