

Ohjelma koostuu viidestä luokasta. Pääohjelmaluokka on ohjelman käyttöliittymä ja se luo instanssit kartanrakentajasta ja reitinhausta. Kartanrakentajaluokalla rakennetaan käyttöliittymän kautta haluttu kartta. Reitinhakuluokassa on ohjelman päätoiminnallisuus, ja reitinhaku luo tarpeellisen määrän solmuolioita ja prioriteettijonon, sekä lisää ja poistaa solmuja prioriteettijonosta. Solmuluokka sisältää solmun tarpeelliset tiedot. Prioriteettijono on säiliö solmuolioille. Reitinhaku käyttää avoimena listana prioriteettijonoa.

Prioriteettijonon lisäämis- ja otto-operaation aikavaativuus on $O(\log n)$. Tilavaativuus $O(n)$. Pienimmän palautus toimii vakioajassa.

Dijkstran aikavaativuus on $O(|E| + |V| \log |V|)$, missä E on kaarien lukumäärä ja V solmujen lukumäärä. Tilavaativuus on $O(n)$.

A*-algoritmin toteutuksen aikavaativuus on teoriassa $O(\log n)$, mutta jostakin virheestä johtuen suorituskykyyn voi tulla piikkejä tietyillä syötteillä. Tilavaativuus on $O(n)$.

Suorituskykyongelmista huolimatta A* toimii huomattavasti nopeammin kuin Dijkstran algoritmi.

Käsiteltyjen solmujen listaukset olisi voinut toteuttaa varmaan jotenkin järkevämmiin, kuin käyttämällä boolean-taulukkoita, vaikka ovatkin aikavaativuudeltaan vakioaikaisia. Käyttöliittymää olisi voinut parantaa esimerkiksi mahdollistamalla tiedostojen lukemisen.

Lähteet:

<http://pages.cs.wisc.edu/~vernon/cs367/notes/11.PRIORITY-Q.html>
<http://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html>
<http://www.policyalmanac.org/games/aStarTutorial.htm>
http://en.wikipedia.org/wiki/A*
<http://algs4.cs.princeton.edu/24pq/>
http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm