

Määrittelydokumentti

Kuvaus työstä

Harjoitustyön tarkoituksena on toteuttaa ja jossain määrin vertailla muutamia lyhyimmän polun etsimiseen käytettäviä verkkoalgoritmeja. Harjoitustyössä käytettäviä algoritmeja ovat kurssilla käsitellyt Bellmanin-Fordin algoritmi ja Dijkstran algoritmi. Käytettävä ohjelmointikieli on Java.

Mikäli aikaa riittää, toteutetaan työhön myös Dijkstraan pohjautuva A*-algoritmi ja jonkinlainen interaktiivinen, graafinen käyttöliittymä.

Algoritmit

Bellmanin-Fordin algoritmi

Ensimmäinen toteutettavista algoritmeista on Bellmanin-Fordin algoritmi, joka löytää lyhimmat polut lähtösolmusta muihin solmuihin ajassa $O(|V||E|)$.

Dijkstran algoritmi

Toinen toteuttava algoritmi on Dijkstran algoritmi, joka löytää lyhimmat polut lähtösolmusta muihin solmuihin ajassa $O(|V|^2)$. Dijkstran algoritmi voidaan myös toteuttaa käyttämällä apuna kekoa, jolloin aikavaativuus on $O((|E|+|V|)\log|V|)$. Harjoitustyössä toteutetaan algoritmi keolla.

A*-algoritmi

A*-algoritmi on yksi tunnetuimpia ja eniten käytettyjä algoritmeja lyhyimmän polun etsimiseen. Se perustuu Dijkstran algoritmiin joka käyttää apuna heuristiikkaa.

Tietorakenteet

Linkitetty lista (korvaa LinkedListin)

Tällä hetkellä solmujen ja kaarien tallennukseen käytetään Javan LinkedList-luokkaa. Lopulliseen ohjelmaan tulee itsetoteutettu versio.

Aikavaativuudet ovat search $O(n)$, insert $O(1)$, delete $O(n)$

Minimikeko (Korvaa PriorityQueueen)

Dijkstrassa ja A*:ssa käytetty Javan PriorityQueue korvataan itsetoteutetulla minimikeolla joka on itseasiassa jopa tehokkaampi: PriorityQueueesta puuttuu heap-decrease-key.

Aikavaativuudet ovat heap-insert $O(\log n)$, heap-del-min $O(1)$, heap-decrease-key $O(\log n)$.

Lähteet

Bellmanin-Fordin algoritmi, Dijkstran algoritmi ja A*-algoritmi

Floréen, Patrik. 2013: Tietorakenteet ja algoritmit-kurssin luentomateriaali. Viitattu 18.10.2014.
<http://www.cs.helsinki.fi/u/floreen/tira2013syksy/tira.pdf>

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