Aineopintojen harjoitustyö: Tietorakenteet ja algoritmit
Vuodenvaihde 2017-2018
Toteutusdokumentti / AltitudeRoutes
16.1.2018
Mikko Kotola

Ohjelman lyhyt kuvaus
AltitudeRoutes on sovellus lyhimmän reitin etsimiseen maaston korkeusmallissa. Sovellus käyttää reitinhakuun A*- ja Dijkstran algoritmeja ja vertailee niiden suorituskykyä reitinhaussa.

Sovellus käyttää lähdeaineistona Maanmittauslaitoksen avoimena datana tarjoamia 2 metrin resoluutiolla toteutettuja korkeusmalleja, jotka sisältävä 3000 x 3000 -resoluutioisen korkeusmallin 6 km x 6 km -maastoalueesta (Maanmittauslaitoksen Maastotietokannan 12/2017 aineistoa). Sovellus tukee myös MML:n 10 m -korkeusmallitiedostoja (2400x1200 ruutua).

Ohjelman yleisrakenne
Ohjelma koostuu seuraavista pakkauksista:

- *altitudeMap*: Kartta
- *controller*: Ohjelman suorituksen ohjaustoiminnot
- *dataStructures*: Itse toteutetut tietorakenteet
- *graph*: Kartasta luotava verkko
- *io*: IO-työkaluja, ascii-kartanlukija ja kuvanpiirtäjä visualisointia varten
- *movementModel:* Etenemismalli, jota käytetään verkon muodostamisessa
- *performanceTesting*: Algoritmien suorituskykytestaus
- *searchAlgo*: Reitinhakualgoritmit
- *ui*: Käyttöliittymät

Huomioita toteutuksesta
Toteutin ohjelman seuraten melko läheisesti alkuperäistä määrittelyä. Aikaa toteutukseen kului kuitenkin sen verran, että jouduin karsimaan optioksi kirjoitetun D*Liten toteutuksen pois. Toisaalta toteutin ohjelman suunniteltua laajemmin mm. sisällyttämällä tuen 2 m -korkeusmallien lisäksi myös 10 m -korkeusmalleille.

Ohjelma tukee pisteiden lähtö- ja maalikoordinaattien syöttämistä joko karttasidonnaisina (yleensä x ja y 1-3000) tai tasokoordinaattijärjestelmän etrs-tm35fin (https://fi.wikipedia.org/wiki/ETRS-TM35FIN) mukaisina.

Ohjelmassa on yksinkertainen konsolikäyttöliittymä. Ohjelma tulostaa kuvia reittihauista ja kirjoittaa reitinhaun tietoja osaksi kuvaa. Reitin solmulistan voi halutessaan tulostaa käyttöliittymässä.

A* on toteutettu perimällä Dijkstran algoritmi ja täydentämällä sitä etäisyysarvioinnilla maalisolmuun. Molemmat algoritmit käyttävät samoja verkkoja ja solmuja.

Saavutetut aika- ja tilavaativuudet (m.m. O-analyysit pseudokoodista)
Dijkstran ja A*:n aikavaativuus on teoreettisesti O((|E| + |V|) log |V|) ja tilavaativuus O(|V|) kun algoritmi toteutetaan käyttäen minimikekoa. Tässä ohjelmassa mallinnan kartan 4 suuntaan yhdistettynä verkkona, jossa jokainen (x,y)-piste on solmu ja sillä on enintään 4 lähtevää kaarta. Verkon mallista johtuen verkossa on kaaria |E| vähemmän kuin neljä kertaa solmujen määrä (|E| < 4*|V|). Näin ollen asymptoottinen aikavaativuus yksinkertaistuu muotoon O(|V|log |V|).

Algoritmien toteutus seuraa Tietorakenteet ja algoritmit -kurssin kurssimateriaalien toteutusmallia. Huomionarvoista on, että molemmat algoritmit kutsuvat osana initialiseSingleSource-rutiinia verkon samaa metodia Graph.resetGraph() (aikavaativuus O(|V|), joka resetoi solmujen etäisyysarviot lähtöön ja maaliin, polut ja kekoviittaukset. Näin ollen tässä osassa algoritmia suoritusaika on molemmilla algoritmeilla sama. A* kuitenkin lisäksi estimoi kaikkien solmujen maalietäisyydet. A*:n implementointia olisi mahdollista tehostaa tältä osin. Asymptoottiseen aikavaativuuteen tämä ei vaikuttaisi. InitialiseSingleSourcen aikavaativuus on molemmilla algoritmeilla O(|V|).

Kukin solmu poistetaan keosta korkeintaan kerran, joten ohjelman pääsilmukka suoritetaan korkeintaan |V| kertaa. Kaikkien pääsilmukan sisällä suoritettavien keko-operaatioiden vaativuus on O(log|V|). Relax suoritetaan ajassa O(1), sillä kullakin solmulla on enintään 4 lähtevää kaarta. Näin ollen algoritmien pääsilmukoiden ja samalla koko algoritmien aikavaativuus on O(|V|log|V|).

Algoritmit käyttävät aputietorakenteena minimikekoa. Jokainen solmu lisätään minimikekoon korkeintaan kerran, joten minimikeko vaatii tilaa O(|V|). Käytännössä verkon mallista johtuen kaikki solmut eivät voi olla samaan aikaan minimikeossa, mutta tilavaativuusluokka on silti sama.

Suorituskyky- ja O-analyysivertailu
Suorituskykyanalyysi on kirjoitettu testausdokumenttiin.

Työn mahdolliset puutteet ja parannusehdotukset
- Käyttöliittymä on hyvin yksinkertainen konsolikäyttöliittymä. Ohjelmaan voisi jatkokehityksessä rakentaa graafisen käyttöliittymän haun määrittelyä varten.
- Etenemismallinnusta olisi mahdollista jatkokehittää hienompisyiseksi.
- A*:n alustusta olisi mahdollista tehostaa luomalla erillinen verkon resetointirutiini (nyt maalietäisyydet ensin nollataan ja sitten lasketaan).

Lisenssit ja oikeudet
Maanmittauslaitoksen korkeusmallit on lisensoitu Creative Commons Nimeä 4.0 Kansainvälinen -lisenssillä. Niitä käytettäessä tulee käyttää mainintaa “Maanmittauslaitoksen Maastotietokannan 12/2017 aineistoa”. Oma lähdekoodini on lisensoitu MIT-lisenssillä.

Lähteet

Wikipedia (2017): A* Search Algorithm. https://en.wikipedia.org/wiki/A*_search_algorithm

Helsingin yliopiston Tietorakenteet ja algoritmit -kurssin luentomateriaali (syksy 2017) / Jyrki Kivinen

Aineistolähde: Maanmittauslaitoksen avoimien aineistojen tiedostopalvelu. http://www.maanmittauslaitos.fi/asioi-verkossa/avoimien-aineistojen-tiedostopalvelu