Requirements Documentation

Author: Henrik Harjula
Course: Aineopintojen harjoitustyö: Tietorakenteet ja algoritmit (loppukesä 2021)
Study program: Mathematical Sciences (Matemaattiset tieteet)
Programming language for project: Python

Topic

Topic of this project is to make a pathfinding application. Different pathfinding algorithms are used to compare their effectiveness (in terms of runtime) in finding the quickest path from one Finnish city to another Finnish city using Finnish highways (valtatie in Finnish). Speed limits of different highways are taken into account. Purpose is to find the quickest path(s) (not necessarily shortest) when driving with car. Application is used from command line – when a shortest path is queried, then a popup window visualizing resulting shortest path(s) is opened.

Algorithms and data structures implemented

Purpose is to implement three pathfinding algorithms in finding quickest path(s) between two Finnish cities using Finnish highways (valtatie): (1) classic **Dijkstra's algorithm**, (2) **JPS** (Jump Point Search) **algorithm**, and (3) **IDA*** (Iterative Deepening A*) **algorithm**. If time permits, fourth (more challenging) Fringe Search algorithm will be implemented.

For all the algorithms, data of Finnish cities (coordinates) and data of distances between Finnish cities using Finnish highways is stored in graph format. For Dijkstra's algorithm, priority queue or heap will be used as additional data structure. JPS algorithm doesn't require any additional data structures. IDA* algorithm will use tree data structure in addition.

Problem solved & Reasoning for algorithms and data structures

Problem to be solved is how to find quickest path between two points (Finnish cities in this project) using edges between points (Finnish highways in this project). Different edges have different weights (distance in kilometres + speed limit when driving with car).

Dijkstra's algorithm is the most classic shortest path finding algorithm, so it is chosen as the benchmark. In this context, no edges have negative weights (since it's not possible to travel between two cities in negative time), so Dijkstra's algorithm can be used. Then, two more efficient A* based algorithms (JPS and IDA*) are used in trying to achieve quicker runtimes in finding shortest path than Dijkstra's. Data structures chosen when implementing the algorithms are the most usual data structures proposed by academic literature when one implements the algorithms in practice.

Input for program & How input is used

Input for the program beforehand consists of necessary geographic data in order to calculate quickest paths between Finnish cities. Data consists of coordinates of Finnish cities (in order to visualize location of

different cities), data of all Finnish highways (valtatie in Finnish – there are 28 highways in Finland currently), data of distances between cities using these highways, and finally data of highway type (motorway or regular highway), as different highways have different speed limits which affect travel time between the cities.

Input the user gives is simply the starting city and ending city – the application then calculates the quickest path(s) and visualizes the path. It also gives information about the performance of different pathfinding algorithms.

Program uses user input in calculating the quickest path between the cities input based on base data input by the programmer (described above).

Time and space complexity targets

Time complexity target of Dijkstra's algorithm is O(|E| + |V|log|V|). Target for space complexity is O(E log V).

Time complexity target for JPS algorithm is O(VV). Target for space complexity is O(1) since the algorithm does not use any additional data structures.

Time complexity target for IDA* algorithm is O(b^d) and space complexity target is O(d) where b is branching factor and d is depth of first solution.

Sources

Sources for data

https://fi.wikipedia.org/wiki/Valtatiet_Suomessa
https://fi.wikipedia.org/wiki/Suomen_tieverkko
https://fi.wikipedia.org/wiki/Suomen_moottoritieverkko
https://fi.wikipedia.org/wiki/Luettelo_Suomen_kuntien_koordinaateista

Sources for algorithms (more to come)

https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm
https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/
https://en.wikipedia.org/wiki/Jump_point_search
https://www.gamedev.net/tutorials/programming/artificial-intelligence/jump-point-search-fast-a-pathfinding-for-uniform-cost-grids-r4220/
https://en.wikipedia.org/wiki/Iterative_deepening_A*
https://algorithmsinsight.wordpress.com/graph-theory-2/ida-star-algorithm-in-general/

Language

Everything related to the project will be written in English (including e.g. documentation, project code, comments). Programming language of the project is Python.