

Spetifikatsioon – Proovireisija

Eesmärk

Eesmärk on ära lahendada proovireisija ülesanne. Kasutaja annab programmile info, kus ta tahab ära käia ning rakendus väljastab kasutajale sellise punktide läbimise järjekorra, mis tagaks lühima teekonna.

Tööriistad

- Tekstiredaktor [Visual Studio Code](#)
- C programmeerimiskeel
- Git versioonihaldus [GitHubis](#)
- GNU C kompilaator + GNU Debugger [MSYS2](#) baasil

Funktsionaalsed nõuded

- Programmile tuleb anda info teedevõrgustiku kohta:
 - Ristmike ristkoordinaadid ning ristmike vastavad identifikaatorid
 - Info, millised ristmikud on omavahel ühendatud
- Programmile tuleb anda info kasutaja poolt tahetavate peatuspunktide kohta:
 - Algus- ja lõpp-punkt – jäävad alati paika
 - Kuni 10 ristkoordinaati, mis tähistavad soovitud peatusi
 - Peatuspunktidel on ka vastavad identifikaatorid
- Programm väljastab lühima peatuste läbimise järjestuse.

Mittefunktsionaalsed nõuded

- Erinevad peatuste järgnevuse esitusviisid:
 - Programm väljastab teekonna graafiliselt, SVG-formaadis väljundfaili või graafilise kasutajaliidesega ekraanile.
 - Programm väljastab teekonna konsooliaknas punktide identifikaatoritena.
- Programm võiks kuvada ka teekonna pikkust ning kaugust peatuste vahel.

- Programm võiks arvutada oodatava ajakulu reisile, arvestab, et linnas on keskmine liikumiskiirus autoga $30 - 40 \frac{km}{h}$.

Piirangud

- Maksimaalne peatuste arv on esialgu ainult 10, sest lühima peatuste läbimise järjekorra arvutamine on keerukusega $O(N!)$, kus N on soovitud peatuste arv. Seetõttu läheb ajakulu ebamõistlikult suureks (suurusjärk minutites), kui soovitud peatuste arv on suurem kui ~ 10 .
- Kui toetada rohkem vahepeatuseid kui 10, siis peab tegema eraldi algoritmi, mis leiab piiratud iteratsioonidearvuga võimalikult mõistliku teekonna läbimise kombinatsiooni. Ei ole garanteeritud, et saadud teekond oleks optimaalne.
- Kasutaja peab tõenäoliselt ise soovitud teedevõrgustiku koostama.
- Kaardiandmestiku kasutamine on keeruline, selleks tuleks eraldi teek kasutada/ise kirjutada. Osad kaartide vektorandmestiku formaadid on „kinnised“, pole ette nähtud kolmanda osapoolte rakendustele kasutamiseks.
- Suurte teedevõrgustike korral (terve linn) tuleb piirata raadius, kus otsingualgoritm töötada tohib, muidu läheb ajakulu ebamõistlikult suureks (suurusjärk minutites).

Mõõdikud

- Esialgu prooviks ülesande ära lahendada graafiteoreetiliselt.
 - Kuni 10 peatust, algus- ja lõpp-punkt alati fikseeritud.
 - Programm leiab, millised teed on lähimad antud punktidele, teedele lisatakse peatused uute vahepunktidenä/graafi tippudena.
 - Programm leiab kasutades [Dijkstra otsingualgoritmi](#) lühimad kaugused soovitud punktide kõikvõimalike kombinatsioonide vahel.
 - Dijkstra algoritmi tulemusena saadud punktidevahelisi kaugusi kombineeritakse, et rakendus väljastaks lühima punktide läbimise järjekorra.
-
- Programm väljastab arvutatud tee ka graafiliselt, kas SVG-faili või kuvab ekraanile graafilise kasutajaliidesega.