Tietorakenteet ja algoritmit harjoitustyö työselostus

Casimir Saastamoinen

2588397

cajoal.cs@gmail.com

Ongelman ratkaisemiseen käytin Djikstran algoritmia, joka oli kurssin koodimateriaaleissa. Algoritmiin muokkasin hieman löysäämisehtoa, jotta se säilyttäisi kasaantuvan painon sijaan keveimmän polun suurimman arvon jokaisen kaupungin kohdalla. Graafin, jota algoritmi käy läpi, loin kurssin koodimateriaaleista löytyvällä WeightedGraph-funktiolla. Djikstran algoritmi soveltuu tilanteeseen, koska se käy läpi yhteyksiä aloituspisteestä lähtien, käyden aina keveimmät reitit ensin. Tällöin algoritmin päästessä haluttuun pisteeseen, on kevein reitti se reitti, jota pitkin algoritmi on sinne juuri kulkenut.

Mittasin Pythonin time-kirjaston avulla ohjelman nopeutta hetkestä, jolloin tiedoston nimi annetaan hetkeen, jolloin reitti on tulostettu. Tulosten perusteella tein exel-taulukon, josta loin kuvaajan.

Kuvaajan perusteella aikakompleksisuus on suurin piirtein n², mikä tarkoittaa että 15 sekunnissa ohjelma pystyisi käymään läpi n. 5000 kaupungin graafin. Ohjelma pystyy suoriutumaan 1000:n kaupungin graafeista alle yhdessä sekunnissa.

