

Tietorakenteet ja Algoritmit

2. Harjoitustyö

Julia Harttunen 291740

Harjoitustyön toisen vaiheen takia kaupunkien structeihin on lisätty vektori `neighbour_town`, joka sisältää kaupungit, joihin vie tie kyseisestä kaupungista. Lisäksi structiin on lisätty `int d`, `int de`, `TownID path_back` ja `Colour colour`. `D`:llä kuvataan linnuntien matkaa, `de` kertoo kuljetun matkan + linnuntien määränpäähän, `path_back` kertoo, mistä on tullut ja `colour` kertoo, kaupungin käsittelyn tilan. Lisäksi `hh`-tiedostoon on lisätty enum `Colour`, joka sisältää värit valkoinen, harmaa ja musta. Valkoinen tarkoittaa, että kaupunkia ei ole käsitelty, harmaa tarkoittaa, että on käsitelty osittain ja musta tarkoittaa, että kokonaan käsitelty. Näitä käytetään hyödyksi 2.vaiheen tehtävissä.

`Hh`-tiedostoon on lisätty vektori `roads_vec`, joka sisältää teiden päätekaupungit. Harjoitustyössä käytetään paljon vektoreita, koska useiden funktioiden täytyy palauttaa vektori. Tallennetun vektorin ansiosta funktion `all_roads` asymptoottiseksi tehokkuudeksi saadaan vakioaikainen. Vektorien kokoja määritellään `reserve`-toiminnolla, jotta `push_back` olisi varmasti asymptoottiselta tehokkuudeltaan vakioaikainen.

Funktiossa `trim_road_network` on käytetty `unordered_map`:iä, koska tietorakennetta ei tarvitse pitää järjestyksessä ja täten saadaan asymptoottista tehokkuutta parannettua lisäämisessä. Funktion `BFS` jonotietorakenne ja funktion `DFS` pinotietorakenne on toteutettu `deque`:na. Olisin voinut käyttää myös valmiita `stack`:iä ja `queue`:ta, mutta tarvittavien toimintojen asymptoottiset tehokkuudet ovat kuitenkin samat. Lisäksi funktiossa `Astar` on prioriteettijono, joka on toteutettu valmiin `priority_queue`:n avulla. Valmis prioriteettijono pitää jonon suuruusjärjestyksessä etäisyyden mukaan. Etäisyydet ovat negatiivisi, jotta saadaan pienimmät luvut ensin.

Harjoitustyön ensimmäisessä vaiheessa en käyttänyt osoittimia, minkä takia en käytä toisessakaan vaiheessa. Osoittimien käyttäminen olisi ollut järkevää, koska ilman niitä jouduin `find`:n avulla etsimään useaan kertaan kaupunkeja. Ainoastaan ensimmäiseen vaiheeseen on tehty muutoksia funktioon `remove_town` ja `clear_all`. Toisen vaiheen takia tarvitsee poistaa myös poistettavasta kaupungista lähtevät tiet ja `clear_all`:ssa täytyy tyhjentää myös `roads_vec`.