

## Tietorakenteet ja algoritmit 1 – harjoitustyö osa 1

### Käytetyt tietorakenteet

Yksittäisen kaupungin tiedot olen tallentanut "Town"-nimiseen structiin. Valitsin kyseisen tietorakenteen, koska se mahdollistaa tallennettujen arvojen hakemisen avainsanalla. Samaan structiin olen tallentanut kaupunkia koskevat solmutiedot projektin toisen vaiheen funktioita varten.

Kaikkien kaupunkien tiedot olen tallentanut unordered\_mapiin, koska sinne alkion lisääminen ja poistaminen on tehokasta. Lisäksi map-rakenne mahdollistaa yksittäisen kaupungin hakemisen tehokkaasti kaupungin id:llä, kun olen tallentanut avaimeksi id:n ja arvoksi structin. Unordered\_mapista tietojen hakeminen on keskimäärin vakioaikaista, joten olen käyttänyt sitä myös disjoint\_settiä trim\_network-funktiossa.

Projektin toisen osan funktioissa olen käyttänyt stack, queue ja priority\_queue tietotyyppejä funktion tarpeista riippuen. Priority\_queueessa lisäsin aina tupla-alkioita, jos arvoja piti muuttaa, koska se oli tehokkaampaa ja helpompaa kuin muuttaa yksittäisen alkion arvoja.

Muuten olen hyödyntänyt työssä vectoreita, joiden alkioina on kaupunkien id:tä tai tiepareja funktiosta riippuen. Pääasiassa siksi, että vectoriin lisääminen ja poistaminen on vakioaikaista vektorin päästä ja useassa funktiossa piti paluuarvo olla kyseistä muotoa.

### Tehokkuuden lisääminen

Tehokkuuden takaamiseksi olen pyrkinyt valitsemaan asympotoottisesti tehokkaista algoritmeja ja näitä tukevia tietorakenteita. Funktioiden toteutuksissa olen pyrkinyt välttämään samojen toimintojen toistuvuutta, jolloin samaa asiaa ei yhdessä funktiossa tehtäisi turhan montaa kertaa. Kiinnitin huomiota myös siihen, että vältin sisäkkäisiä for- ja while-loop rakenteita.

Vector-rakenteista käytettäessä olen käyttänyt reserve-funktiota, jolla voidaan varata tarvittavat muistialueet. Tällöin voidaan varmistua siitä, että vektorille kutsuttaessa push\_back on tehokkuudeltaan aina vakio. Asympotoottista tehokkuutta ajatellen olen pyrkinyt välttämään rekursiivisten funktioiden ja silmukoiden sisällä muita kuin asympotoottiselta tehokkuudelta vakioita funktioita.

### Muutoksia ja kehitysehdotuksia

Ensimmäisen vaiheen koodiin ei tehnyt muita muutoksia kuin kaupungin poiston yhteyteen, jolloin poistetaan myös kaupunkiin kulkevat tiet. Samoin clear\_all funktioon lisäsin funktiokutsun teiden poistamiseen.

Toista vaihetta tehtäessä huomasin, että osoittimien käyttö olisi voinut olla aiheellista työssä. Ilman osoittimia toteutukseen jouduin laittamaan moneen kertaan find-funktioita. Vaikka valitsemani tietorakenteet tekivät findin käytöstä asympotoottisen tehokkuuden näkökulmasta tehokasta, osoittimet olisivat ajaneet saman asian tehokkaammin.