Seuraavaksi tulee esitys tietorakenteista ja selitys miksi juuri ne ovat valittu

Harjoitustyön toiseen vaiheeseen on luotu kaksi tietorakennetta – toinen kulkuväylille ja toinen risteyksille. Molemmat tietorakenteet tallennetaan *unordered_map*:iin. Kulkuväylien avaimena toimii *WayID* ja risteyksien avaimena sen koordinaatti. Molemmat avaimet ovat uniikkeja, joten multimappia ei tarvita. Mappien valueina on omat erilliset struct-rakenteet: *struct Way* ja *struct Crossroad*. Valitsin unordered_mapin, koska tietorakenteen ei tarvitse olla koko ajan järjestyksessä ja järjestyksen luominen aiheuttaisi tehonkulutusta. Unordered_map on keskiarvoltaan vakio, kun sieltä etsitään tiettyjä alkioita ja työssä tarvitsee kaikkein eniten etsiä tiettyjä alkioita tietorakenteesta.

Risteyksen struct tietorakenne on hieman monimutkaisempi, koska siellä on risteyksen omien tietojen lisäksi tieto sinne johtavista kulkuväylistä. Nämä kulkuväylät on tallennettu vectoriin, joka on muotoa *std::vector<std::pair<Crossroad*, Way*>>*, vectorista löytyy myös väylä toisessa päässä olevat risteykset, eli kyseisen risteyksen ”naapuriristeykset”. Risteyksen structissa on myös algoritmien vaatimia muuttujia, jotka nollataan aina ennen, kun uusi algoritmi ajetaan.

Tehokkuuteen vaikuttaa tällä kertaa eniten käytetyt algoritmit, mutta unordered_map on ollut mielestäni tehokkain tapa varastoida ohjelman tietoja.