

Ohjelmointi4: Tietorakenteet ja Algoritmit

Misse Nysse?

Osuus 1 prg1

Valitut tieto rakenteet

```
std::unordered_map<RegionID,Region> regions_;  
std::vector<RegionID> regionIDs_;  
  
std::unordered_map<StopID,Stop> stops_;  
std::vector<StopID> stopIDs_;
```

Käytin toteutuksessani unordered_mappeja, sillä ne mahdollistivat usein kutsuttujen komentojen kuten add_stop, add_region asympotoottisesti tehokkaan toteutuksen. Unordered_mappien lisäys muualle, alkion poisto ja tietyn etsintä ovat kaikki $O(N)$, theta(1) mikä oli syy, että käytin kyseisiä tietorakenteita tallentamaan structit. Vektoreita käytettiin hyväksi operaatioissa joita ei otettu huomioon tehokkuustesteissä, niiden helppokäyttöisyytensä vuoksi.

Projektissa lähtökohtaisesti pyrin juurikin optimoimaan operaatiot joita kutsuttiin usein kuten add_region. Etenkin operaatioissa, joita kutsuttiin harvemmin koodasin toimivan $O(N)$ kompleksisen ratkaisun ja mietin sitten hetken voisiko joillan valmiilla algoritmeilla tehdä saman asian jonka tein for loopissa.

```
struct Stop{  
    StopID StopID;  
    Name Name;  
    Coord Coord;  
    RegionID belongsToRegion = NO_REGION;  
};  
  
struct Region{  
    Name name;  
    RegionID parentRegion = NO_REGION;  
    std::vector<StopID> stops = {};  
    std::vector<RegionID> subRegions = {}; //children  
};
```

Valitut structit, joista löytyy kaikki tarpeellinen ja mahdollistaa parent-children suhteen.