private:

```
struct Town {
        TownID town_id = NO_TOWNID;
        Name town_name = NO_NAME;
        Coord coords;
        int tax = NO_VALUE;

        Town *master = nullptr;
        std::vector<Town*> vassals;
    };

std::unordered_map<TownID,Town> town_db;
```

Struct Towneja säilytetään unordered_mapissa siksi, että ei ollut tarvetta pitää kaupunkeja järjestyksessä id:n perusteella, joka kuitenkin ymmärtääkseni oli ainoa yksilöivä tekijä kaupungeissa ja täten ainoa järkevä avain mihin tahansa tietorakenteeseen. Unordered_map tarjoaa myös nopeat lisäykset, haut, viittaukset ja poistot.

Olisi ollut mahdollista lisätä tähän valmiiksi tietorakenteita, jonne päivitetään kaupunkeja aakkos- ja etäisyysjärjestykseen aina, kun kaupunki lisätään (tai poistetaan). Sellainen ratkaisu minulla jo olikin, jotta sain aakkos- ja etäisyysjärjestyksiä (sekä min ja max) käsitteleville funktioille O(1) uhraamalla add_townin nopeutta. Luin kuitenkin Mattermostista kurssin keskustelua, enkä tiedä oliko vika minun ymmärryksessäni vai muutaman avainhenkilön keskenään ristiriitaisilta vaikuttavista sanomisista, mutta tulkitsin tilannetta niin, että tällaisia valmiita tietorakenteita ei ehkä katsottaisi suopeasti, joten päädyin poistamaan ne.

public:

```
TownID furthest_town(TownID id, int temp = 0, int distance = 1, TownID id_return
= NO_TOWNID);
```

Lisäsin vain yhden funktion, jonka avulla saan tietää tietystä kaupungista kauimpana olevan vasallikaupungin. Tätä kutsutaan longest_vassal_pathissa.