Mikko Merioksa
014155394

Tietorakenteiden harjoitustyö – Määrittelydokumentti

1. Käytettävät tietorakenteet ja algoritmit

Tavoitteenani on tutkia erilaisia algoritmeja, jotka etsivät nopeimman mahdollisen reitin annetusta labyrintistä ja vertailla näiden tehokkuuksia. Valitsin aiheeksi reitinhaun, koska aihe tuntuu mielenkiintoiselta, mutta myös arvioni mukaan tämänhetkisille ohjelmointitaidoilleni sopivalta.

Toteutettavat algoritmit ovat A*, Dijkstra ja Bellman-Ford, joista A* ja Dijkstra toimivat samalla periaattella, kun taas Bellman-Fordin algoritmi on lähempänä ”brute force”-lähestymistapaa ongelmaan. Siinä missä Dijkstra ja A* tutkivat kartan (eli verkon) solmuja aina parhaaksi kullakin hetkellä katsotusta lähtien, Bellman-Ford yksinkertaisesti käy kaikki verkon solmut läpi niin monta kertaa, että paras reitti on varmasti löytynyt.

Aputietorakenteina algoritmit käyttävät minimikekoa, josta siis Dijkstra ja A* hakevat aina kulloinkin lyhyimmän arvioidun matkan sisältävän solmun. Keon aikavaativuus on sekä lisättäessä että paras alkio poistettaessa O(log n). Toinen tarvittava tietorakenne on yksinkertainen linkitetty lista, jonka aikavaativuus lisättäessä on O(1) ja poistettaessa sekä alkiota haettaessa O(n).

2. Ongelma

Tutkittava ongelma on siis eri reitinhakualgoritmien tehokkuus. Valitsin aiheen koska reitinhaku on paitsi mielenkiintoinen aihe (esimerkiksi tekoälyn kannalta, josta olen kiinnostunut), myös mielestäni sopiva tämänhetkiselle taitotasolleni.

Valitsin juuri valitsemani algoritmit, koska ne olivat tulleet tutuiksi aiemmin käymälläni Tietorakenteet ja algoritmit -kurssilla, joten ajattelin niiden olevan sopivan tuttuja että voisin itse alkaa niitä toteuttamaan.

3. Syötteet

Ajatukseni on, että ohjelma saa syötteenään labyrintin kaksiulotteisena merkkitaulukkona, jonka merkit kuvaavat seiniä ja lattiaa alla olevan esimerkin mukaisesti:

```
##.##
#...#
#.#.#
#...#
#####
```

. = lattia

= seinä

Labyrintin voi algoritmille syöttää joko antamalla tiedostonimen joka sisältää halutun labyrintin, tai suoraan ohjelmalle syöttämällä rivi kerrallaan. Labyrintin lisäksi ohjelmalle voidaan syöttää halutut lähtö- ja maalikoordinaatit, joiden väliltä reitti etsitään. Jos näitä ei anneta, ovat vakiokoordinaatit kartan vasen ylänurkka ja oikea alanurkka.

4. Algoritmien aikavaativuudet

Aikavaativuustavoite A*-algoritmilla on sama kuin Dijkstran algoritmilla, eli O((E + V) log V),

missä V on labyrintin solmujen (eli lattiaruutujen) lukumäärä ja E niitä yhdistävien kaarien lukumäärä. Tässä tapauksessa siis yhdestä solmusta lähtevien kaarien lukumäärä on muille kuin kartan reunoilla olevilla solmuilla 4. Tilavaativuudeltaan itse algoritmit ovat O(1), joskin oma toteutukseni tallettaa aina viimeksi käytetyn kartan solmujen tiedot, jolloin tilavaativuudeksi saadaan O(n).

Bellman-Fordin algoritmi on selvästi kahta muuta tehottomampi, ja sen aikavaativuus onkin O(V * E). Tilavaativuudeltaan Bellman-Ford vastaa kahta muuta algoritmia.