

Toteutusdokumentti

Tässä dokumentissa käsittelen tuottamani algoritmin tehokkuus- ja tilavaatimuksien toteutumista ohjelman logiikan kannalta. Tämä kattaa labyrintin alustuksen sekä itse reitin etsimisen.

Ohjelma luo maapalat ajassa $O(n)$, sillä jokainen maapala täytyy käydä lävitse luontiprosessissa.

```
for i = 0 : i < koko; i++;  
    for j = 0: j < koko; j++;  
        labyrntti[j][i] = new Maapala(j, i);
```

Näin ollen vaadittu aika on riippuvainen syötteen eli labyrintin koosta: $O(n)$

Tilavaativuus on myöskin riippuvainen syötteen koosta, joten sekin on: $O(n)$

Koodissa oleva muuttuja koko on labyrintin sivun pituus, joten syöte on todellisuudessa

koko x koko, mistä johtuu vaativuus $O(n)$ eikä $O(n^2)$. Jokainen paikka käydään läpi kerran eikä kahta kertaa.

Kun maapalat on luotu, rakennetaan seinät ja asetetaan lähtö- ja loppupisteet.

Nämä operaatiot tapahtuvat vakio ajassa, sillä maapaloilla on omat muuttujat sille, ovatko ne seiiniä, vai ei.

Tämän lisäksi Maapalarekisterille syötetään tiedot alun ja lopun koordinaateista, joten sekin tapahtuu vakio ajassa $O(1)$.

```
maapala.asetaseinaksi()  
    this.seina = true;
```

```
maapalarekisteri.asetalkuX(int luku)  
    this.alkuX = luku;
```

Sijoitus operaatiot ovat vakio aikaisia, joten operaatiot ovat kokonaisuudessaan vakioaikaisia.

Tämän jälkeen maapaloille asetetaan heuristiset arvot sen mukaan, miten kaukana ne ovat loppupisteestä x-koordinaatin ja y-koordinaatin suhteen. Jokainen alkio käydään läpi, joten aikavaatimuksena on $O(n)$, kuten ensimmäisessä kohdassa on selitetty. Heuristisen arvon laskeminen sekä sijoitus kyseiselle maapalalle ovat vakioaikaisia operaatioita, joten metodin vaativuudeksi tulee $O(n)$.

Lopuksi itse lyhimmän reitin etsimisessä käytetyn algoritmin aika- ja tilavaativuudet.

Jokainen alkio (maapala) asetetaan enintään kerran avoimelle listalle. Tämän lisäksi jokaista alkiota avoimella listalla tarkistellaan enintään kerran ennen kuin loppupiste olisi löydetty.

Tästä johtuen pahin tapaus, mikä voisi tapahtua vaatisi, että kaikki pisteet käytäisiin läpi ennen lopun löytymistä eli aikavaativuudeksi tulisi $O(n)$.

Näin ollen:

ohjelman aikavaativuus $O(n)$

ohjelman tilavaativuus $O(n)$