Totetutusdokumentti

Ohjelman yleisrakenne:

Ohjelma toimii siten että se ensiksi luo 64x64 Levelin (Taulukko Tilejä) sitten kutsuu levelin apply metodilla Java8 Consumer rajapinnan toteuttavaa MazeGenerator luokkaa. MazeGenerator luo Leveliin täydellisen labyrintin. Sitten ohjelma tulostaa levelin merkkijonoesityksen ja sen jälkeen luo png kuvan levelistä. Sitten ohjelma luo uuden Level olion ja Tekee samat asiat Levelille paitsi että MazeGeneratorin sijasta apply metodin parametrina on RoomGenerator. RoomGenerator nimensä mukaisesti luo huoneita tasoon. Sitten ohjelma luo vielä kerran uuden Level olion ja tekee samat asiat kuin edellisillä kerroilla mutta applyn parametrinä on DungeonGenerator.

Pino:

Pino on toteutettu taulukolla, jonka pituus kaksinkertaistuu kun se on täynnä.

Pop operaatio on vakioaikainen koska siinä tehdään vain vakioaikaisia operaatioita.

Push operaatio on vakioaikainen kun pinon sisäinen taulukko ei ole täynnä. Operaatio on lineaarinen kun taulukko on täynnä ja pinon kasvatusoperaatiota joudutaan kutsua. Koska kasvatusoperaatio kaksinkertaistaa taulukon koon kaksinkertaistuu myös push kutsujen määrä, jonka jälkeen kasvatusoperaatio kutsutaan. Tämän takia push operaation aikavaativuus on keskimääräisesti vakioaikainen.

HashSet:

Lisäyksen keskimääräinen aikavaativuus on vakio.

Määrittelydokumentissa tasogeneraattorille tekemääni O-analyysiä:

Huoneiden generoinnin aikavaativuus riippuu lineaarisesti yritysmäärän suhteen. Täydellisen labyrintin generoinnin aikavaativuus on syvyyssuuntaisen läpikäynnin aikavaativuus eli O(|V|+|E|) jossa V on kartan korkeus kerrottuna leveydellä ja E on (x – 1) * y + (y – 1) * x jossa x on leveys ja y on korkeus. Eli täydellisen labyrintin generoinnin aikavaativuus riippuu lineaarisesti kartan pinta-alasta. Kartan yhtenäistämisen aikavaativuus riippuu huoneiden määrästä, joka riippuu huoneiden laittamisyritysten lukumäärästä.

Tarkennuksia analyysiin:

Testausdokumentissa tekemieni suorituskykytestien perusteella tekemäni analyysi ei ole ainakaan täysin väärin. Labyrinttigenerointi näyttäisi kasvavan lineaarisesti pinta-alan mukaan

Huonegeneroinnin aikavaativuus ei sen sijaan riippu lineaarisesti yritysmäärän suhteen. Pahin algoritmin tapaus on sellainen, jossa kaikki huoneiden sijoitusyritykset (sijotusyritysten määrä annetaan luokan konstruktorin parametrina) onnistuvat. Eli pahimmassa tapauksessa, jos sijoitusyrityksiä on n määrä niin huoneita luodaan n määrä. Parhaimmassa tapauksessa luodaan vain 1 huone ja loput n-1 huone yritystä epäonnistuvat huone törmäykseen.eli tässä tapauksessa vakio aikaisia huonetörmäystarkistuksia tehdään n-1 määrä eli parhaan tapauksen aikavaativuus on lineaarinen. Pahimmassa tapauksessa jokaiselle huoneelle tehdään törmäystarkistuksia h määrä jossa h on valmiiksi generoitujen huoneiden määrä. Viimeiselle huoneelle n-1 törmäys tarkistusta. Siis koko algoritmissa tehdään pahimmassa tapauksessa 0+1+2+3+4+5+6+7+8+...+n-1 törmäystarkistusta eli törmäystarkistuksia tehdään $(1/2)m^2 + (1/2)m$ kertaa jossa m = n-1. Eli pahimman tapauksen aikavaativuus on: $O(n^2)$

Se että kuinka lähellä pahinta tapausta suoritusaika on riippuu tuurista johon Levelin koko vaikuttaa. Isommalla pinta-alalla on suurempi todennäköisyys siihen, että arvottu huone ei leikkaa aikaisempien huoneiden kanssa.