

Jakub Heczko - Projekt Numer 3 Labirynt

- A więc problemem zadania było wygenerowanie losowo labiryntu za pomocą algorytmu kruskala. Jedyną różnicą było to, aby użyć losowego wybierania krawędzi, a nie według najmniejszej wagi. Aby taki labirynt wygenerować, należy stworzyć graf, który będzie topologicznie wyglądał jak siatka, a następnie z niego trzeba wygenerować drzewo rozpinające go. Można to zrobić między innymi algorytmem kruskala, który modyfikujemy wybierając losową krawędź. Dostajemy wtedy graf, który jest podgrafem naszej siatki, ponieważ graf rozpinający przechodzi przez wszystkie wierzchołki więc labirynt zawsze będzie mieć wyjście. Oto pseudokod algorytmu kruskala, ale dla losowej krawędzi:

```
Funkcja kruskal(){  
    Utwórz nowy graf g;  
    Utwórz nową pustą tablicę arr;
```

```
    Dla v, które idzie po wszystkich wierzchołkach:  
        wstaw v do grafu;
```

```
    Dla e idącego po wszystkich krawędziach:  
        wstaw e do tablicy arr;
```

```
    Stwórz podział Partition dla każdego wierzchołka z grafu g;
```

```
    Dopóki rozmiar tablicy arr nie jest równy 0 i Partition.count > 1:
```

```
        e = losowo wybrana krawędź z arr;  
        v0 = pierwszy wierzchołek e;  
        v1 = drugi wierzchołek e;  
        s = wyszukaj w Partition zbiory zawierający v0;  
        t = wyszukaj w Partition zbiory zawierający v1;  
        jeśli(s!=t):  
            złącz te dwa zbiory s i t w Partition;  
            wstaw krawędź e do grafu g;
```

```
    Zwróć graf g;  
}
```

- Użyte struktury to graf, zbiór oraz tablica
- Operacje wstawiania, wyszukania czy krawędź istnieje, wyciągnięcia krawędzi w grafie są w czasie stałym $O(1)$, ale operację przejrzenia, czyli iterator po krawędziach jest w czasie $O(n^2)$. W zbiorze wszystko jest stałe oprócz operatorów i iteratorów które są w czasie liniowym.
- Budowanie grafu na początku to złożoność czasowa $O(n^2)$, tak samo pamięciowa. Zmodyfikowany algorytm kruskala ma złożoność czasowa równą $O(n^2)$, pamięciowa złożoność wynosi $O(3*n^2)$. Algorytm generowania wizualizacji, również jest $O(n^2)$, złożoność pamięciowa jest stała.
- Uruchamianie poprzez skompilowanie main.cpp, można wprowadzić dwa argumenty poprzez argumenty programu, ale można to też zrobić za pomocą programu