

# Izolacja

## Dokumentacja projektu

### 1. Opis rozwiązania problemu

- a. Wczytanie danych
- b. Konwersja: wejście -> graf

Tworzę graf, który składa się z wierzchołków tylu ile dróg na mapie.

Każdy punkt może posiadać z początku 4 krawędzie, do wszystkich punktów pomiędzy, którymi odległość wynosi 1. Reszta zostanie uzupełniona/wyliczona przy pomocy algorytmu floydaWarshala.

### c. Alorytm Floyd-Warshall

- i. użycie najprostszej wersji algorytmu, bez żadnych usprawnień. O złożoności  $O(n^3)$ .

```
for (k = 0; k < numberRoad; k++) {
    for (i = 0; i < numberRoad; i++) {
        for (j = 0; j < numberRoad; j++) {
            if (getGraph(i, k) == -1 || getGraph(k, j) == -1) {
                continue;
            }

            weight = getGraph(i, k) + getGraph(k, j);

            if (getGraph(i, j) > weight || getGraph(i, j) == -1) {
                setGraph(i, j, weight);
            }
        }
    }
}
```

### d. Algorytm Bron-Kerbosch

W pierwszej kolejności przechodzę po wszystkich parach i edytuję wagę w 2 przypadkach:

- jeśli waga = -1 ustal max\_int (dowolna liczba > L)
  - ponieważ znaczy to, że nie ma połączenia pomiędzy wierzchołkami, leżą w grafach niezależnych od siebie
- jeśli waga w jest  $0 < w \leq L$  -> ustal wagę na -1
  - aby wykluczyć połączenia, które są zbyt blisko siebie

W tej chwili nasze zadanie redukuje się do znalezienia grafu pełnego.

Wykorzystałem do tego klasyczną wersję algorytmu Brona-Kerboscha. [Link](#)

Znalezienie grafu pełnego zawierającego ilość, wierzchołków o liczbie min. K jest równoważne ze znalezieniem rozwiązania zadania. Ponieważ graf, który podajemy do algorytmu Bron-Kerbosch'a zawiera tylko i wyłącznie krawędzie których waga jest > L.