

Graph

W tym projekcie realizowano:

- Utworzenie listy krawędzi na podstawie macierzy sąsiedztwa
- Budowy drzewa opinającego za pomocy algorytmu Kruskala
- Utworzenie macierzy sąsiedztwa na podstawie listy krawędzi
- Wyszukiwanie najtańszej(najkrótszej) drogi pomiędzy dwoma miastami za pomocy algorytmu Dijkstry

Wejściowe dane:

- Plik miasta.txt w którym mamy dane w formacie
X1 Y1 miasto1
X2 Y2 miasto2
- Plik matrix.txt z macierze sąsiedztwa o rozmiarze (ilość miast) x (ilość miast)

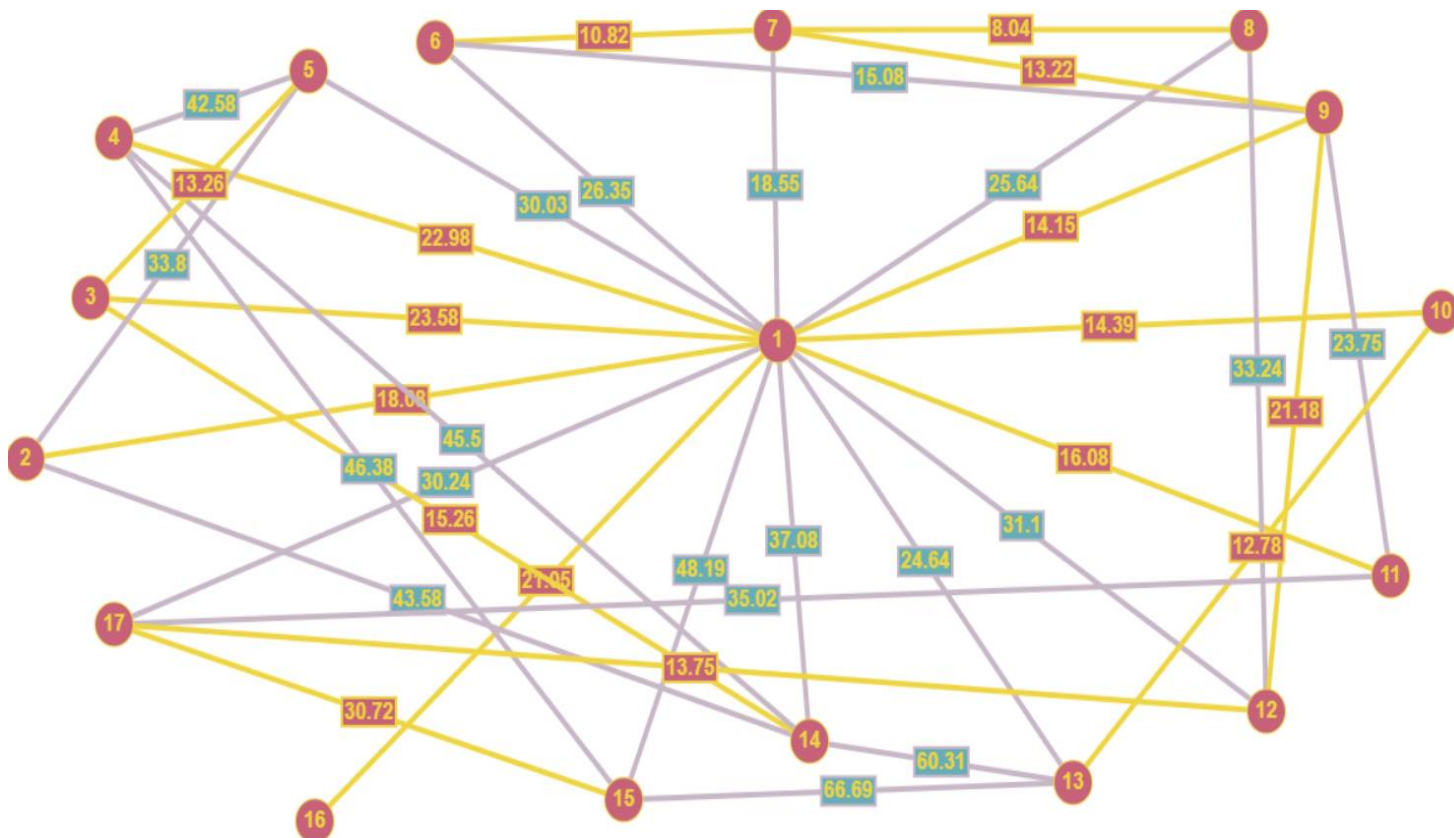
Za pomocy plików miasta.txt i matrix.txt otrzymali listu wszystkich krawędzi z wagami wynik zapisany w pliku AllAdjacencies.txt, poniżej te związki są wizualizowane



Teraz chcemy utworzyć drzewo opinające, dla tego będzie wykorzystany algorytm Kruskala:

Na początku bieżący zestaw krawędzi jest pusty. Następnie, o ile jest to możliwe, przeprowadzana jest następująca operacja: ze wszystkich krawędzi, których dodanie do istniejącego zestawu nie powoduje pojawienia się w nim cyklu, krawędź masy minimalnej jest wybierana i dodawana do istniejącego zestawu. Gdy nie ma już takich krawędzi, algorytm jest zakończony. a zestaw znalezionych krawędzi to drzewo opinające o minimalnej wadze.

Na wyjściu mamy plik MST_Adjacencies.txt poniżej te krawędzi są wizualizowane



Po usunięciu niepotrzebnych krawędzi i uporządkowaniu ich mam taki graph





Dla wyszukiwania najtańszej(najkrótszej) drogi pomiędzy dwoma miastami użyjmy algorytmu Dijkstry

Musimy zachować odległość ścieżki każdego wierzchołka. Możemy przechowywać to w tablicy o rozmiarze v , gdzie v jest liczbą wierzchołków. Chcemy również uzyskać najkrótszą ścieżkę, nie tylko znać długość najkrótszej ścieżki. W tym celu mapujemy każdy wierzchołek na wierzchołek, który jako ostatni zaktualizował długość ścieżki. Po zakończeniu algorytmu możemy cofnąć się od wierzchołka docelowego do wierzchołka źródłowego, aby znaleźć ścieżkę. Kolejka o minimalnym priorytecie może być użyta do efektywnego odbioru wierzchołka przy najmniejszej odległości ścieżki.

Przykład, znalezienie drogi pomiędzy Szczecin a Gdańsk

D:\STUDY\ЛыюЁшЕь\Graphs\Graphs\bin\Debug\Graphs.exe

```

Distance from Szczecin to Gdansk = 113
Szczecin Wroclaw Poznan Lodz Warszawa Bydgoszcz Gdansk

Process returned 0 (0x0)   execution time : 0.038 s
Press any key to continue.

```

W kodzie kolejny krok zrobiony w osobnym pliku dla widoczności, oczywiście ze to wszystko można zrobić porządniej, ale wtedy nie można będzie zobaczyć kolejny krok działania