

# Projektowanie Algorytmów i Metody Sztucznej Inteligencji

## Laboratorium 7 i 8

### UWAGA:

1. Zadanie należy rozpocząć od implementacji grafu przechowującego elementy określonego typu, oraz podstawowych operacji na grafie, zgodnie z informacjami podanymi na wykładzie (2 wersje: macierz sąsiedztwa i lista sąsiedztwa).
2. Na ocenę bardzo dobrą program musi zostać napisany obiektowo z wykorzystaniem dziedziczenia. Na ocenę dostateczną wystarczy przedstawić implementację grafu wraz z procedurami zapisu i odczytu grafów z pliku, a także implementację jednego z algorytmów: Kruskala lub Prima.

### Opis zadania

Bieżąca lista jest poświęcona badaniu efektywności algorytmów Kruskala i Prima w zależności od metody reprezentacji grafu. Należy uwzględnić reprezentacje grafu w postaci macierzy sąsiedztwa oraz listy sąsiedztwa. Badania należy wykonać dla 5 różnych liczb wierzchołków w grafie  $V$  (np. 10, 50, 100, 500 i 1000), oraz następujących gęstości grafu: 25%, 50%, 75% oraz dla grafu pełnego. Dla każdego zestawu parametrów: algorytm, reprezentacja grafu, liczba wierzchołków i gęstość grafu należy wygenerować po 100 losowych instancji, natomiast w sprawozdaniu umieścić wyniki uśrednione.

Przed implementacją algorytmów Kruskala i Prima należy opracować procedury (metody) losowego generowania grafów (przy zadanych parametrach: liczba wierzchołków i gęstość), zapamiętania ich struktury w zależności od przyjętej metody reprezentacji oraz wczytania/ zapisania struktury grafu z/do pliku.

### Sprawozdanie

Sprawozdanie powinno zawierać:

- krótkie wprowadzenie,
- omówienie planu i przebiegu eksperymentów, oraz przedstawienie uzyskanych wyników (w postaci tabel i wykresów),
- podsumowanie i wnioski (w przypadku niezgodności uzyskanych wyników z przewidywaniami spróbować wyjaśnić przyczyny),
- literaturę (materiały wykorzystane do wykonania ćwiczenia, w tym strony internetowe).

### Literatura

1. Cormen T., Leiserson C.E., Rivest R.L., Stein C., Wprowadzenie do algorytmów, WNT
2. Drozdak A., C++. Algorytmy i struktury danych, Helion
3. <http://www.algorytm.org/algorytmy-grafowe/algorytm-prima.html>
4. <http://www.algorytm.org/algorytmy-grafowe/algorytm-kruskala.html>
5. <http://users.v-lo.krakow.pl/~toma/algorytmy/Algorytmy%20grafowe.pdf>

### Krótki opis algorytmów

Dany jest spójny, nieskierowany graf  $G=(V, E)$ , w którym  $V$  to zbiór wierzchołków, zaś  $E$  to zbiór krawędzi. Z każdą krawędzią  $(u, v) \in E$  związana jest waga  $w(u, v)$ . Wyznaczenie minimalnego drzewa rozpinającego polega na znalezieniu acyklicznego podzbioru  $T \subseteq E$ , który łączy wszystkie wierzchołki i którego łączna waga  $\sum_{(u,v) \in T} w(u, v)$  jest najmniejsza. Graf  $T$  jest acykliczny i zawiera wszystkie wierzchołki, wobec tego jest drzewem. Drzewo to rozpina graf  $G$ , dlatego nazywamy je drzewem rozpinającym. Dwa najpopularniejsze algorytmy wyznaczania minimalnego drzewa rozpinającego to algorytm Kruskala oraz algorytm Prima. Złożoność obliczeniowa obu algorytmów to  $O(E \log V)$  w przypadku implementacji z wykorzystaniem zwykłego kopca binarnego, zaś wykorzystując kopce Fibonacciego można zredukować złożoność algorytmu Prima do  $O(E + V \log V)$ . Dokładny opis algorytmów można znaleźć między innymi [1] oraz [2].