

Algorytmy grafowe [2 tygodnie lub wybór zadań]

wszelkie prawa zastrzeżone
zakaz kopiowania, publikowania i przechowywania
all rights reserved
no copying, publishing or storing

Maciej Hojda

Uwaga: Słowa „dany”, „zadany”, „podany”, „wybrany” itd. w kontekście parametrów (zmiennych) oznacza parametr zadany przez użytkownika (a nie na stałe, przez programistę), a implementacja wykorzystująca taki parametr powinna obsługiwać jego różne wartości.

Uwaga: Grafy wyświetlaj korzystając z biblioteki `matplotlib` (lub innej graficznej).

1 Zadanie nr 1 – przeszukiwanie w głąb

Dla danego grafu wczytywanego z pliku zaimplementuj algorytm przeszukiwania w głąb.

Procedurę przedstaw graficznie, krok po kroku.

Wejście: plik z grafem, początkowy wierzchołek.

Wyjście: wizualizacja.

2 Zadanie nr 2 – sortowanie topologiczne

Dla danego grafu wczytywanego z pliku zaimplementuj algorytm sortowania topologicznego.

Procedurę przedstaw graficznie, krok po kroku.

Wejście: plik z grafem.

Wyjście: wizualizacja.

3 Zadanie nr 3 – minimalne drzewo rozpinające

Dla danego grafu wczytywanego z pliku zaimplementuj algorytm Kruskala (generacji minimalnego drzewa rozpinającego).

Procedurę przedstaw graficznie, krok po kroku.

Wejście: plik z grafem, początkowy wierzchołek.

Wyjście: wizualizacja.

4 Zadanie nr 4 – najkrótsza ścieżka

Dla zadanego grafu wczytywanego z pliku zaimplementuj algorytm Dijkstry (poszukiwania najkrótszej ścieżki między parą wierzchołków).

Procedurę przedstaw graficznie, krok po kroku.

Wejście: plik z grafem, para wierzchołków.

Wyjście: wizualizacja.

5 Zadanie nr 5 – maksymalny przepływ

Dla zadanego grafu wczytywanego z pliku zaimplementuj algorytm Edmondsa-Karpa (poszukiwania maksymalnego przepływu).

Procedurę przedstaw graficznie, krok po kroku.

Wejście: plik z grafem, para wierzchołków.

Wyjście: wizualizacja.