

Wstęp do Algorytmów

Kierunek: Inżynieria Systemów

Semestr Letni – 2024/2025

Lista 5 – Grafy

O grafach również możemy poczytać w Cormenie (to taka Biblia programistów, algorytmików – polecam, Popek).

Pomocnicze wskazówki i fragmenty kodu do implementacji grafów są w materiałach na Teamsie.

Istnieje pokusa, by graf zapisywać jako macierz, ale zachęcam do prób skorzystania z list sąsiedztwa albo z dedykowanych klas.

Czasami chcemy z grafu (spójnego) nieskierowanego ważonego usunąć zbędne krawędzie w taki sposób, aby pomiędzy dowolnymi dwoma wierzchołkami istniała ścieżka, ale żeby jednocześnie suma wag wszystkich pozostawionych krawędzi była możliwie najmniejsza. Powstaje w ten sposób Minimalne Drzewo Rozpinające (Minimal Spanning Tree).

Dwa najbardziej popularne algorytmy to Algorytm Kruskala (https://pl.wikipedia.org/wiki/Algorytm_Kruskala) i Algorytm Prima (https://pl.wikipedia.org/wiki/Algorytm_Prima).

- Większość algorytmów grafowych działa dla grafu spójnego, w którym pomiędzy dowolnymi dwoma wierzchołkami istnieje ścieżka. Dlatego często pierwszym krokiem podczas aplikacji algorytmów grafowych jest sprawdzenie, czy graf jest spójny. Jeśli okazuje się, że nie jest, dzielimy go na składowe spójne. Napisz metodę losującą graf nieskierowany (na przykład losując symetryczną macierz binarną lub losowo dodając połączenia). Dla wylosowanego grafu nieskierowanego zaimplementuj i uruchom zaproponowaną przez siebie metodę podziału grafu na składowe spójne. Zaproponuj wizualizację wyników (możesz skorzystać z istniejących bibliotek do wyświetlania grafu lub zaproponować rozwiązanie własne, na przykład rozmieszczając wierzchołki losowo na płaszczyźnie).
 - Opis problemu podziału grafu na składowe spójne: https://eduinf.waw.pl/inf/alg/001_search/0129.php
 - Wyświetl przypisanie wierzchołków do poszczególnych składowych.
- Dla zadanego grafu nieskierowanego (możesz go wylosować) zaimplementuj algorytm Dijkstry dla wyszukiwania najkrótszej ścieżki pomiędzy dwoma losowymi węzłami tego grafu. Zwróć uwagę, że może ona nie istnieć.)
 - Wyświetl odpowiedź jako ciąg wierzchołków i odległości między nimi, aby można było przynajmniej częściowo zweryfikować poprawność implementacji.
 - Zmodyfikuj algorytm Dijkstry, aby obsługiwał więcej niż jeden punkt początkowy. Znajdź długość najkrótszej ścieżki do najbliższego z punktów początkowych.
 - Zwróć uwagę, że suma (w sensie sumy zbiorów krawędzi) ścieżek z danego wierzchołka wyznaczonego przez algorytm Dijkstry tworzy pewne drzewo rozpinające. Czy potrafisz określić jego własność?
- Zaimplementuj algorytm Kruskala i algorytm Prima oraz przetestuj je na wybranym przykładzie.