Implementacja Algorytmu Floyda-Warshalla

Bartłomiej Stachów

Wstęp

Program zawiera implementację algorytmu Floyda-Warshalla, który służy do znajdowania najkrótszych ścieżek pomiędzy wierzchołkami w grafie ważonym.

Opis algorytmu

Algorytm tworzy macierz odległości distance o rozmiarze n×n (n oznacza liczbę wierzchołków w grafie). Element distance[i, j] tej macierzy będzie oznaczał najniższy koszt dojścia od wierzchołka i-tego do j-tego. Po utworzeniu macierzy algorytm wypełniana ją największymi wartościami dodatnimi (plus nieskończoność). Następnie elementy distance[i, i] (na diagonali), dla i = 0, 1, ..., n - 1 ustawia na 0. Oznacza to, że koszt dojścia z wierzchołka i-tego do niego samego jest zerowy. Teraz dla każdego wierzchołka grafu k i każdej pary wierzchołków i, j badamy, czy koszt dojścia distance[i, j] jest większy od sumy kosztów dojść distance[i, k] = distance[k, j]. Jeśli tak, to za koszt dojścia distance[i, j] przyjmujemy wartość tej sumy. Operacja ta wymaga trzech zagnieżdżonych pętli dla k, i, j. Z tego powodu algorytm Floyda-Warshalla ma klasę złożoności obliczeniowej równą $O(n^3)$.

Program

Program składa się z plików:

- *graph.py*, który zawiera klasę *Graph*. czyli implementację grafu ważonego skierowanego z użyciem słownika oraz listy. Metody klasy *Graph*:
  - *add_node(node)* – dodaje nowy wierzchołek *node* do grafu.
  - *add_edge_directed(source, target, weight)* – dodaje skierowaną krawędź z wierzchołka *source* do wierzchołka *target* z wagą *weight.*
  - *list_nodes()* - zwraca listę wierzchołków grafu.
  - *list_edges()* - zwraca listę krawędzi grafu.
  - *print_graph()* - wypisuje postać grafu w formie:

    ['wierzchołek1'] : ['wierzchołek2'](waga_1-2) ['wierzchołek3'](waga_1-3)

    ['wierzchołek2'] : ['wierzchołek3'](waga_2-3) ['wierzchołek4'](waga_2-4)

    …

- read_csvgraph(csvfile) – wczytuje graf z pliku csv.
- *floyd_warshall() - czyli implementację algorytmu.*
    1. Tworzę macierz odległości distance (indeksy macierzy zależą od indeksu danego elementu w słowniku), która jest wypełniona wartościami INF.
    2. Aktualizuję wartości distance[i][i] = 0.
    3. Aktualizuję wartości wynikające z bezpośrednich krawędzi w grafie.
    4. Wykonuję trzy zagnieżdżone pętle i aktualizuję wartości krawędzi.
    5. Tworzę i zwracam nowy graf utworzony w oparciu o macierz odległości.

- *main.py,* w którym używając metody read_csvgraph wczytuje się graf oraz wykorzystując metodę *floyd_warshall* tworzy się nowy graf z najkrótszymi ścieżkami początkowego grafu.
- 10.csv oraz graph.csv zawierające zapis grafu w formie:

  wierzchołek_początkowy, wierzchołek_docelowy, waga_krawędzi

  …

Sposób uruchomienia

Aby uruchomić program należy wywołać komendę *python main.py*