Dokumentacja projektowa

Ogólne informacje

Algorytm Dijkstry, opracowany przez holenderskiego informatyka Edsgera Dijkstrę, służy do znajdowania najkrótszej ścieżki z pojedynczego wierzchołka w grafie o nieujemnych wagach krawędzi. Mając dany graf z wyróżnionym wierzchołkiem (źródłem) algorytm znajduje odległości od źródła do wszystkich pozostałych wierzchołków i wyróżnia najmniej kosztowne ścieżki.

Projekt

Katalog dijkstra składa się z plików:

- main.cpp,
- graph.h,
- minheap.h,
- dijkstra.h
- Makefile

Aby skompilować program należy użyć standardowo polecenia „make”, a żeby uruchomić „./main.exe”.

Plik graph.h zawiera imlementacje grafu opartą na liście sąsiedztwa. Każdy wierzchołek ma swoją własną listę sąsiedztwa, w której on sam jest elementem początkowym. Wszystkie listy wierzchołków są indeksowane w tablicy w klasie „class Graph”. Wierzchołki dodaje się do grafu za pomocą funkcji addEdge(int src, int dest, int weight), której dwa pierwsze parametry to wierzchołek początkowy i wierzchołek, który chcemy dołączyć, a trzeci to waga krawędzi.

Plik minheap.h zawiera implementacje kopca, który podczas wykonywania algorytmu będzie służył do przechowywania wierzchołków, których najkrótsza odległość nie została jeszcze znaleziona. Klasa „class MinHeap” zawieraz funkcje:

- minHeapify(int index) - służy do ustalania prawidłowej struktury kopca
- isEmpty() – sprawdza, czy kopiec jest pusty
- extractMin() – wyciąga najmniejszy element z kopca
- decreaseKey(int v, int dist) – aktualizuje dystans między wierzchołkami

- isinMinHeap(int v) – sprawdza czy wierzchołek v jest w kopcu

Plik Dijkstra.h zawiera funkcję wykonującą algorytm – dijkstra(struct Graph* graph, int src, int wyniki[]), funkcja:

1. Tworzy kopiec zawierający wierzchołki grafu.
2. Inicjuje odległość od wierzchołka startowego do niego samego na 0.
3. Inicjuje pozostałe odległości na nieskończoność.
4. Dopóki kopiec nie jest pusty wyodrębnia się wierzchołek A z najmniejszą odległością i sprawdza, czy jego wierzchołek B jest w kopcu. Jeśli tak i jeśli jego odległość od wierzchołka źródłowego jest większa niż odległość od A do B + odległość od wierzchołka źródłowego do A to odległość jest aktualizowana.

Złożoność obliczeniowa algorytmu wynosi O(E log V) (V to liczba wierzchołków, a E liczba krawędzi).

Plik zawiera też funkcję wyświetlającą wyniki algorytmu – printArr(int dist[], int n).

Plik main.cpp zawiera funkcje main, która tworzy graf oraz demonstruje działanie algorytmu.