

Sprawozdanie - grafy

200439

Graf jest to abstrakcyjny typ danych, składający się z wierzchołków oraz krawędzi. Poszczególne wierzchołki mogą być połączone krawędziami skierowanymi lub nieskierowanymi w taki sposób, że każda krawędź zaczyna się i kończy w którymś z wierzchołków. Ponadto krawędzie grafu mogą mieć różne wagi. Za pomocą takiego grafu ważonego można zobrazować na przykład połączenia drogowie między miastami, wtedy wagi krawędzi mogłyby oznaczać długości tych dróg lub czas potrzebny do ich przebycia. Za pomocą specjalnych algorytmów takich jak Branch and Bound czy A^* można znaleźć najkrótszą drogę między miastami (wierzchołkami grafu). Istnieją także prostsze algorytmy trawersacji (przeszukiwania) grafów takie jak Przeszukiwanie w głąb (DFS) czy Przeszukiwanie wszerz (BFS).

Graf można zaimplementować na kilka sposobów. Wykonałem dwa przykłady jego implementacji - za pomocą listy sąsiedztwa oraz macierzy sąsiedztwa. Niestety dla implementacji macierzowej, rozmiar tablicy dla kilku tysięcy wierzchołków grafu staje się zbyt duży, przez co program zostaje unicestwiony przez system operacyjny. Ograniczone zasoby maszyny wirtualnej, na której pracuję, mogą powodować ten stan rzeczy.

Oprócz implementacji samej struktury grafu, napisano także funkcje jego przeszukiwania:

1 Przeszukiwanie w głąb (DFS - Depth First Search)

Algorytm ten polega na trawersacji wszystkich krawędzi wychodzących z wierzchołka, po czym następuje powrót do danego wierzchołka. Jest to algorytm rekurencyjny, polegający na zasadzie ustawiania i pobierania etykiet wierzchołków i krawędzi (jako nieodwiedzone lub odwiedzone).

2 Przeszukiwanie wszerek (BFS - Breadth First Search)

Algorytm przeszukiwania wszerek rozpoczyna swoje działanie od wybranego wierzchołka początkowego. W drugiej kolejności odwiedzane są węzły połączone z wierzchołkiem początkowym, następnie ich sąsiedzi itd.

3 Przeszukiwanie wykorzystujące heurystykę - A*

Algorytm A-star jest algorytmem heurystycznym, zupełnym i optymalnym, czyli znajduje najkrótszą ścieżkę pomiędzy podanymi wierzchołkami w grafie ważonym. Algorytm zaczynając od wierzchołka początkowego tworzy ścieżkę wybierając za każdym razem wierzchołek x z tzw. listy otwartej wierzchołków, czyli sąsiadujących z poprzednim wierzchołkiem, tak, by zminimalizować funkcję:

$$f(x) = g(x) + h(x)$$

gdzie g to droga pomiędzy wierzchołkiem początkowym, a danym wierzchołkiem x , a h to przewidziana przez heurystykę droga od x do wierzchołka docelowego. W zaimplementowanym algorytmie w programie, wartość h to odległość w sensie euklidesowym wierzchołka x o współrzędnych x i y do wierzchołka docelowego.

W każdym kroku dołączany jest wierzchołek o najniższym współczynniku f . Algorytm kończy działanie gdy natrafia na wierzchołek docelowy.

Algorytm A* jest najbardziej optymalnym spośród podanych wcześniej metod wyznaczania najmniejszej odległości między wierzchołkami (DFS, BFS). Ten algorytm został zaimplementowany dla grafu z macierzą sąsiedztwa.

A* może być używany na przykład w programach obliczających najlepszy sposób przemieszczania się z miejsca do miejsca w systemach gps czy wyznaczaniu optymalnej trasy, którą trzeba przebyć komunikacją miejską (np. jakdojade.pl). Algorytm ten stosowany jest także w dziedzinie sztucznej inteligencji, imitując inteligentne zachowania i podejmowanie decyzji przez komputer.