

Implementacja grafu oraz algorytmów przeszukiwania - Depth First Search oraz Breadth First Search

Monika Litwin 200586

19 maja 2014

Sposób implementacji grafu

W programie graf jest przedstawiany za pomocą listy sąsiedztwa. Polega to na tym, że tworzona jest tablica n-elementowa (n oznacza liczbę wierzchołków w grafie). Każde pole tej tablicy skojarzone jest z jednym wierzchołkiem - numer wiersza jest numerem wierzchołka. Elementy tablicy są vectorami z szablonu STL, zawierającymi wierzchołki grafu, z którym dany wierzchołek połączony jest krawędzią.

Za reprezentację elementów grafu odpowiadają specjalne struktury - Wierzchołek oraz Krawędź. Krawędzie definiowane są przez wierzchołki, które łączą (posiadają wskaźniki na nie) oraz wagę. Wierzchołki natomiast przez numer indeksu, posiadają również 2 pola typu bool, określające, czy wierzchołek istnieje oraz, czy został już odwiedzony (wykorzystywane w algorytmie DFS/BFS).

Reprezentowany typ grafu

Program obsługuje grafy nieskierowane. Możliwe są wielokrotne krawędzie między wierzchołkami (mogą się różnić wagą) oraz pętle. Wierzchołki dodawane są pokolei, dopiero w przypadku, gdy któryś z nich zostanie usunięty, inny może zostać dodany na jego miejsce.

Przeszukiwanie DFS

Algorytm wykorzystuje stos z szablonu stack. Odkłada na niego wierzchołki do odwiedzenia, zaczynając od tego, który został zadany jako początkowy. Zdejmuje go ze stosu, sprawdza czy nie został już wcześniej odwiedzony, jeśli nie to oznacza go jako odwiedzony i przeszukuje jego wektor sąsiedztwa. Dodaje na stos sąsiadów i w następnym przejściu pętli pobiera ze stosu i przetwarza ostatniego z dodanych sąsiadów. Przechodzenie jest kontynuowane dalej (w głąb grafu) aż zostanie osiągnięty wierzchołek, który nie posiada nieodwiedzonych sąsiadów. Wówczas procedura odwiedza kolejnego ze stosu sąsiada poprzedniego wierzchołka i przeszukuje dalej wzdłuż dostępnej krawędzi, aż przejdzie cały graf.

Przeszukiwanie BFS

Algorytm wykorzystuje kolejkę z szablonu queue. Odkłada na jej koniec wierzchołki do odwiedzenia, zaczynając od tego, który został zadany jako początkowy. Zdejmuje go z początku kolejki, sprawdza czy nie został już wcześniej odwiedzony, jeśli nie to oznacza go jako odwiedzony i przeszukuje jego wektor sąsiedztwa. Dodaje do kolejki sąsiadów i w następnym przejściu pętli pobiera z kolejki i przetwarza pierwszego z dodanych sąsiadów. Przechodzenie jest kontynuowane dalej (w szerz grafu), aż zostaną odwiedzone wszystkie wierzchołki połączone krawędzią z początkowym. Wtedy procedura przechodzi do pierwszego sąsiada, drugiego odwiedzonego wierzchołka i tak dalej, aż przejdzie cały graf.