Sprawozdanie z laboratorium - PAiMSI. Implementacja grafu.

Patryk Jędrzejko 200406

June 1, 2014

1 Wprowadzenie

W danym ćwiczeniu należało zaimplementować strukturę grafu.

Graf - jest to struktura, która zbudowana jest z wierzchołków, czyli punktów, które połączone są przez krawędzie. Położenie takich wierzchołków na danej płaszczyźnie jest dowolne. Krawędzie pełnią role drogi, po której możemy się poruszać w takim grafie.

Na zamieszczonym obrazku grafu (**znajdującego się na drugiej stronie**) widzimy elementy takie jak V oraz E. Gdzie odpowiednio V - to wierzchołek, a E - krawędź takiego grafu.

2 Działanie programu

Na starcie programu wyświetlane jest menu, w którym użytkownik może:

- Dodawac węzły do grafu
- Usuwać stworzone węzły
- Dodać krawędzie
- Usuwać stworzone krawędzie
- Wyświetlić utworzony graf

W moim przypadku graf został zaimplementowały za pomocą listy sąsiedztwa. Cała konstrukcja programu opiera się na klasie Graf, w której zawarłem klasę Wezel oraz Krawedz. Implementacja takiego grafu jest podobna do reprezentacji grafu za pomocą macierzy sąsiedztwa, czyli mamy tablicę, która zawiera elementy o pewnej liczbie wierzchołków w grafie. W moim przypadku elementy tablicy są listami, które zawierają numery wierzchołków w grafie, do których prowadzi z danego wierzchołka krawędź.

Dzięki takiej implementacji grafu oszczędzamy na pamięci komputera, ponieważ nie rezerwujemy owej pamięci dla nieistniejących krawędzi, jak w przypadku macierzy sąsiedztwa. Korzystamy z pamięci wtedy, jeśli taka krawędź zostanie utworzona.
Kolejnym plusem zastosowania takiej reprezentacji grafu jest łatwy dostęp do każdego z elementów, czyli w tym wypadku do sąsiadów wierzchołka. Zatem możemy szybciej wyszukać dany element, bądź też mieć do niego dostęp.

3 Wnioski:

- Implementacja grafu za pomocą listy sąsiedztwa jest najefektywniejszą reprezentacją grafu, w stosunku do macierzy sąsiedztwa.

- Jeśli chodzi o wymagania pamięciowe to dla listy sąsiedztwa wynoszą one O(|E|), gdzie E - to liczba utworzonych krawędzi. Dla dodania nowej krawędzi mamy czas stały, który wynosi **0.003 ms**. Dla sprawdzenia czy dana krawędź istnieje mamy O(|E|), jak również dla sprawdzanie stopnia danego wierzchołka oraz usunięcia krawędzi. Zatem widać, że w porównaniu do macierzy sąsiedztwa implementacja listą lepiej wypada dla złożoności pamięciowej oraz czasowej.

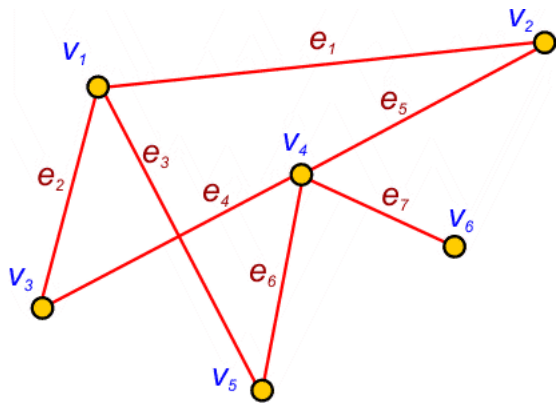

Figure 1: Ilustracja grafu.

Powyższa ilustracja pokazuje nam jak wygląda przykładowy graf, gdzie mamy ponumerowane wierzchołki oraz krawędzie.