Grzegorz Malczewski 218480

Sprawozdanie 7: Grafy, przeszukiwanie dfs i bfs.

Celem siódmych zajęć laboratoryjnych była implementacja realizująca strukturę abstrakcyjnego typu danych jakim jest graf, oraz dwóch algorytmów jego przeszukiwania: w głąb (dfs) i w szerz (bfs). Sama implementacja grafu może zostać zrealizowana na więcej niż jeden sposób. Najbardziej popularne metody implementacji opierają się na liście lub macierzy. W tym przypadku zadanie zostało zrealizowane w oparciu o macierz co ma swoje wady i zalety. Głównym powodem wyboru takiej realizacji była łatwość dostępu do krawędzi każdego z wierzchołków, co jest bardzo wydajne i przyjazne w użytkowaniu w przypadku przeszukiwań. Ceną takiego rozwiązania jest bardzo duża macierz kiedy graf składa się z wielu elementów, jako że ma ona zawsze wymiar NxN, gdzie N – ilość elementów w grafie, bez względu na ilość połączeń.

Przeszukiwanie grafu także popularnie realizuje się na 2 sposoby. Pierwszy sposób to przeszukiwanie w głąb. Opiera się on na sprawdzaniu sąsiadów elementu, następnie spośród nich wybieramy jednego z sąsiadów i ponawiamy proces do momentu, w którym znajdziemy szukany element, lub natrafimy na sytuację, że odwiedziliśmy wszystkich możliwych sąsiadów danego elementu. O ile taki algorytm może szybko znaleźć szukane połączenie to jest dosyć losowy i znalezione połączenie prawdopodobnie nie będzie najkrótsze.
Alternatywna metoda to przeszukiwanie w szerz. Polega ona na sprawdzaniu wszystkich sąsiadów, przed sprawdzeniem sąsiadów sąsiadów. Taka metoda powinna znaleźć krótsze połączenia, lecz w szerszym grafie może mieć większą złożoność.
Poniższy wykres przedstawia średni czas szukania połączenia w grafie, gdzie każdy element poza końcowymi ma 2 sąsiadów.

Jak widać na powyższym wykresie algorytm bfs okazał się zdecydowanie bardziej skuteczny, nawet w przypadku, gdzie teoretycznie dfs powinien mieć najlepszy dla siebie przypadek (graf zdegradowany do listy). Sugerowało by to pewne wady implementacji zrealizowanej przez studenta, których niestety nie udało się znaleźć. Pomimo tego algorytm bfs wydaje się bardziej przekonujący ideowo ze względu na pewną losowość algorytmu dfs.
Złożoność obu algorytmów zdaje się być wielomianowa.