Spis treści

1 Wstęp **2**

2 Krótki opis programu **2**

3 DFS vs BFS **3**
3.1 Wyniki : 3
3.2 Uwagi : 4

4 Algorytm A* **4**

5 Wnioski: **5**

Sprawozdanie 8

Grafy v1.5

Paweł Żurek 200404

14.0.2014

1 Wstęp

Prosty program, w którym użyte są podstawowe funkcje grafów.

2 Krótki opis programu

Program po uruchomieniu pyta się z ilu wierzchołków ma stworzyć graf a następnie udostępnia proste menu :

- Ustawienie połączeń między wierzchołkami
- Dodanie wierzchołka
- Usunięcie wierzchołka
- Dodanie krawędzi
- Usunięcie krawędzi
- Wyświetlenie sąsiadów wierzchołka
- Wyświetlenie grafu (za pomocą macierzy połączeń)
- Podanie ilości wierzchołków
- Wyświetlenie aktualnych wierzchołków
- Algorytm Depth First Search
- Algorytm Breadth First Search
- Losowanie połączeń wraz z ich wagami
- Algorytm A* (A star)

Jako, że zaimplementowałem strukturę grafu za pomocą macierzy połączeń, najważniejszym elementem mojego programu jest :

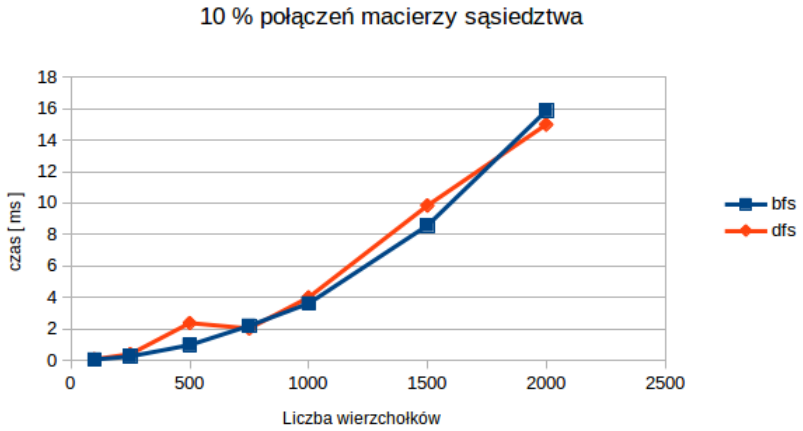
- Macierz typu int (tablica dynamiczna dwu wymiarowa) przechowująca aktualne połączenia między wierzchołkami

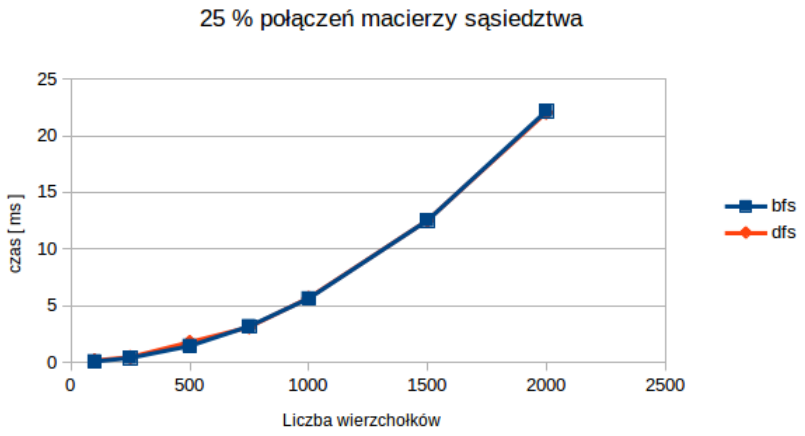
W tej wersji programu uwagę skupiłem na algorytmach **dfs**, **bfs** i **A***. Dlatego też tym algorytmom będzie poświęcona dalsza część sprawozdania.

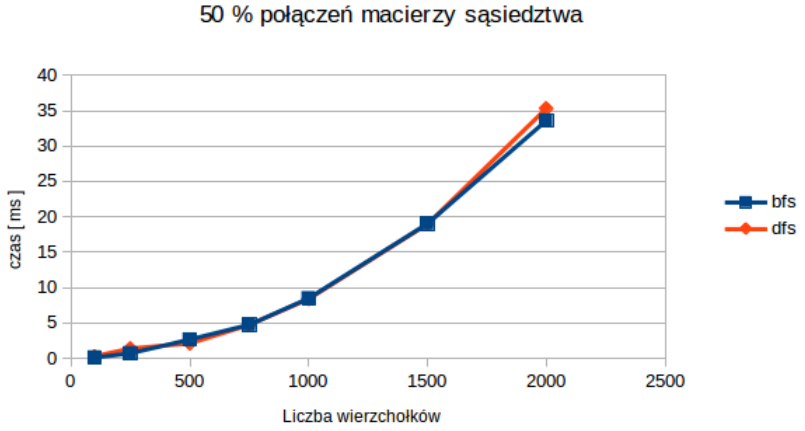
3 DFS vs BFS

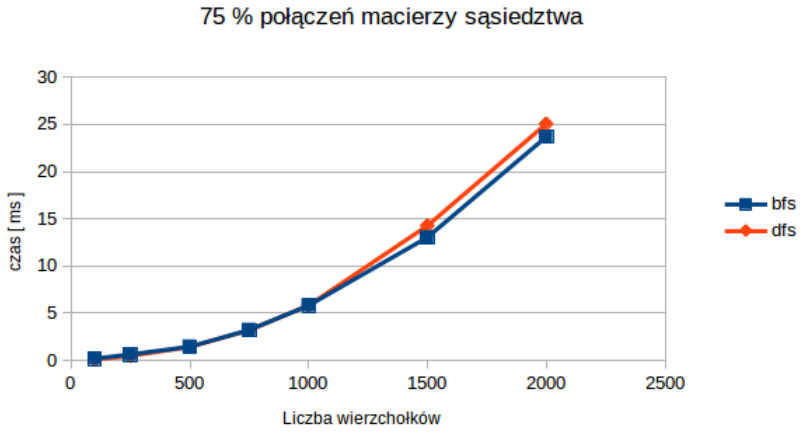
Algorytm Depth First Search działa poprawnie. Jako argument, funkcja przyjmuje dowolny wierzchołek, od którego zacznie przeszukiwanie. Wynikiem funkcji jest ciąg wierzchołków połączonych w pośredni lub bezpośredni sposób z wierzchołkiem przyjętym jako argument funkcji.
Algorytm Breadth First Search również działa poprawnie. Mimo tego, że oba algorytmy posiadają inną metodę działania, działanie funkcji wywołującej oba algorytmy jest dokładnie taki sam.

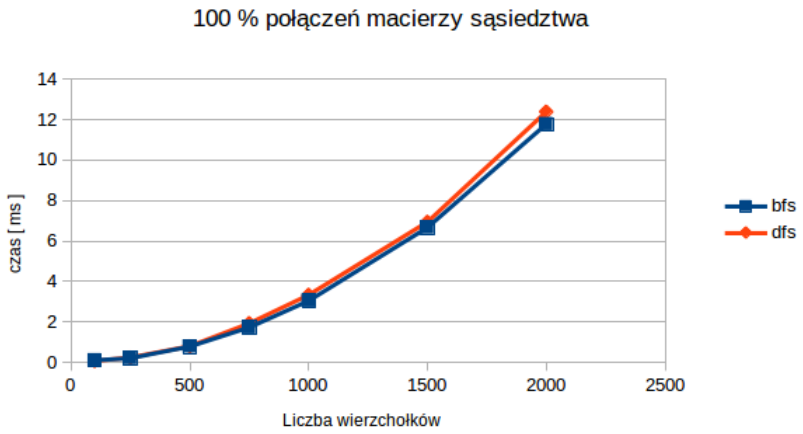
3.1 Wyniki :




25 % połączeń macierzy sąsiedztwa
czas [ms]
Liczba wierzchołków
bfs
dfs


50 % połączeń macierzy sąsiedztwa
czas [ms]
Liczba wierzchołków
bfs
dfs


75 % połączeń macierzy sąsiedztwa
czas [ms]
Liczba wierzchołków
bfs
dfs



3.2 Uwagi :

- Zarówno algorytm **bfs** jak i algorymt **dfs** uruchamiałem z tym samym argumentem : wierzchołkiem o numerze 1.

4 Algorytm A*

Na dzień dzisiejszy (14.05.2014) algorytm ten nie działa poprawnie. Jest to głównie spowodowane trudnościami w przejściu z struktur opisanych w materiałach znalezionych w sieci lub literaturze na strukturę, którą operuję. We wszystkich spotkanych mi materiałach, algorytm ten jest stosowany, dla struktur dwu wymiarowych i szukania na nich połączenia. W moim programie, ciężko przedstawić w taki sposób zbiór wierzchołków.
Algorytm jest zupełny i optymalny, w tym sensie, że znajduje ścieżkę, jeśli tylko taka istnieje, i przy tym jest to ścieżka najkrótsza. Stosowany głównie w dziedzinie sztucznej inteligencji do rozwiązywania problemów i w grach komputerowych do imitowania inteligentnego zachowania.

5 Wnioski:

- Algorytm **dfs** jest znacznie prostszy w implementacji niż **bfs**. Przynajmniej w mojej wersji grafu
- Czasy wykonywania algorytmów **bfs** i **dfs** są bardzo zbliżone (na tyle, że nie jestem pewien czy program dobrze działa). Częściej mniejsze czasy są dla **dfs**, lecz tak małe różnice (na poziomie milisekund) mogą być spowodowane obsługą innych procesów w tle procesora.

- algorytm **A*** jest prawdopodobnie najefektywniejszym algorytmem tego typu. Dzieje się tak głównie poprzez fakt, iż algorytm na bieżąco "przewiduje", która ścieżka jest optymalna.

Dokładne wyniki programu są zamieszczone w pliku (obecny folder) wykresy.xls. Podobnie wszystkie wykresy są dostępne w osobnych plikach (format png).