

Projektowanie algorytmów i metody sztucznej inteligencji

Dawid Marszałkiewicz 218665

8 maja 2016

1 Zadanie

Zaimplementować BFS oraz DFS, a następnie zmierzyć czas przeszukiwania grafu wypełnionego N elementami, gdzie $N = 10, 100, 1000, 10000$.

2 Analiza zadania

2.1 Graf - implementacja

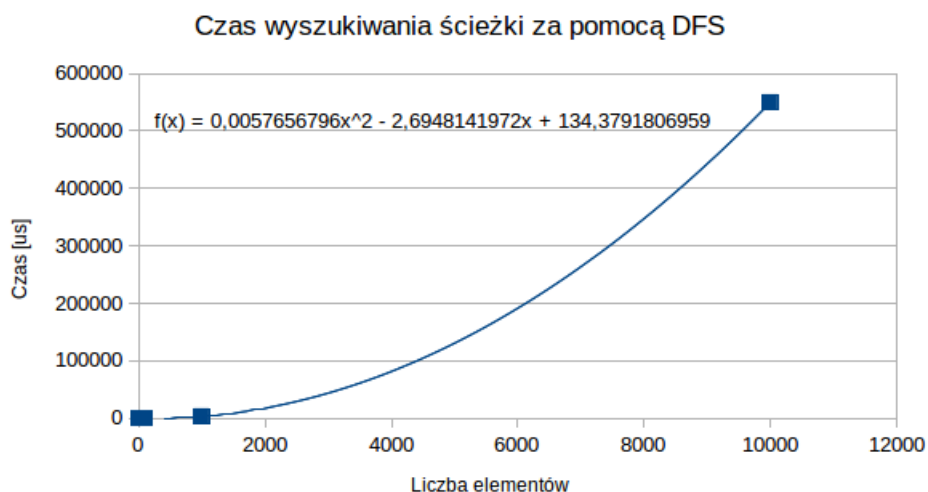
Graf został zaimplementowany na macierzy sąsiedztwa, ponieważ w zamian za dużą złożoność pamięciową $O(n^2)$ otrzymujemy stały czas potrzebny do: dodania nowej krawędzi, usunięcia nowej krawędzi oraz sprawdzenia czy krawędź istnieje.

2.2 Depth First Search

2.2.1 Tabela ze średnimi wynikami przeszukiwania

| l. zapisanych elementów | czas |
|-------------------------|-------------|
| | [μ s] |
| 10 | 1,3 |
| 100 | 41,0 |
| 1000 | 3193,4 |
| 10000 | 549754,3 |

2.2.2 Wykres czasu przeszukiwania dla n elementów

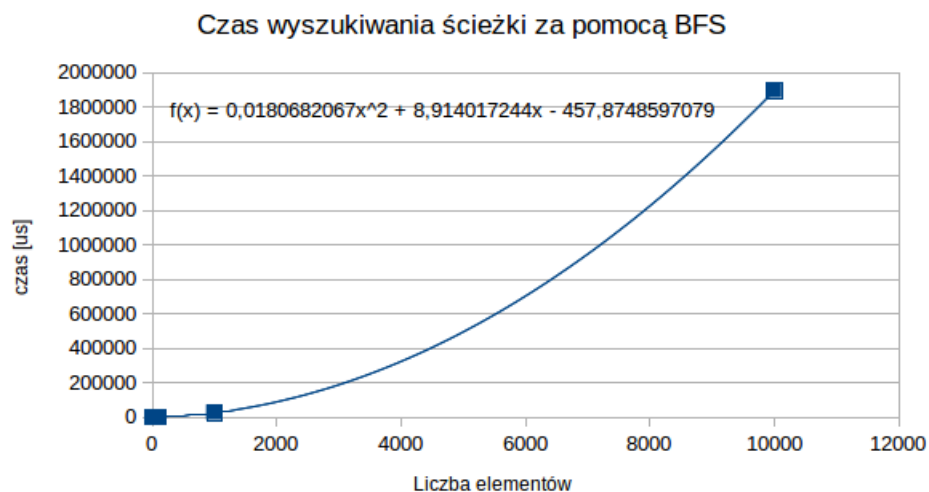


2.3 Breadth First Search

2.3.1 Tabela ze średnimi wynikami przeszukiwania

| l. zapisanych elementów | czas |
|-------------------------|------------|
| | [μ s] |
| 10 | 2,9 |
| 100 | 203,7 |
| 1000 | 26565,4 |
| 10000 | 1895502,6 |

2.3.2 Wykres czasu przeszukiwania dla n elementów



3 Wnioski

- Zgodnie z wykresem 2.3.2 złożoność obliczeniowa algorytmu przeszukiwania wszerz wynosi $O(n^2)$, natomiast złożoność obliczeniowa algorytmu przeszukiwania włąb zgodnie z 2.2.2 wynosi $O(n^2)$. Złożoność ta jest zdeterminowana poprzez implementację grafu - w najgorszym przypadku, należy sprawdzić n^2 krawędzi.
- Zakładając, że wszystkie krawędzie mają jednakową wagę to algorytm przeszukiwania wszerz (BFS), znajduje najkrótszą ścieżkę pomiędzy wierzchołkami.
- Ze względu na dużą złożoność pamięciową macierzy sąsiedztwa testy zostały ograniczone z 10^6 do 10^4 .