

Sprawozdanie z zajęć laboratoryjnych

Algorytmy grafowe – przeszukiwanie BFS i DFS

1. Wstęp:

Celem ćwiczenia było zaimplementowanie grafu oraz metod jego przeszukiwania: przeszukiwanie w głąb („DFS”) i wszerz („BFS”).

Jako metodę reprezentacji grafu wybrano macierz sąsiedztwa ze względu na złożoność sprawdzenia istnienia krawędzi $O(1)$, oraz prostotę implementacji. Wadą tej implementacji jest duża złożoność pamięciowa $O(n^2)$, gdzie n to liczba wierzchołków. Złożoność przejścia wszystkich krawędzi to $O(V^2)$.

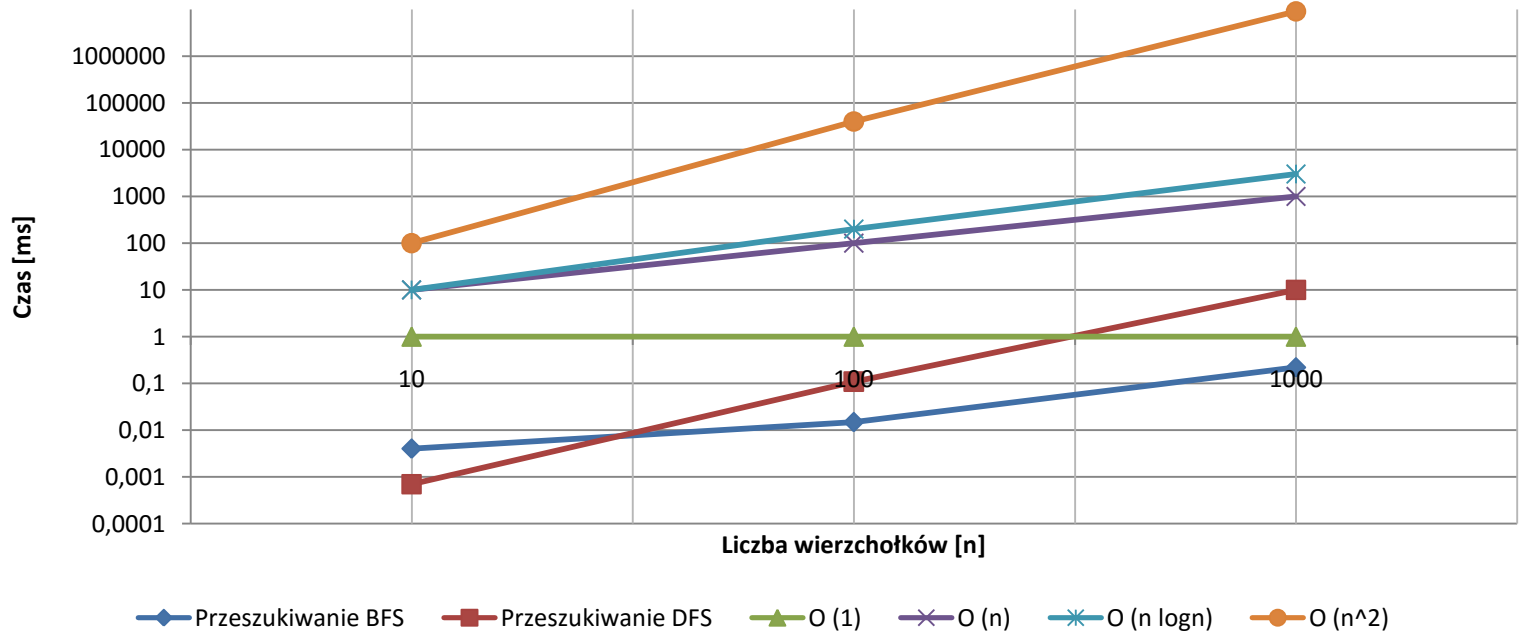
Dla każdej liczby n wierzchołków, stworzono $n+0.3n$ krawędzi losowo rozmieszczonych między istniejącymi wierzchołkami.

2. Wyniki:

| BFS: | | | | DFS | | | |
|----------|-------|--------|--------|----------|--------|--------|--------|
| n: | 10 | 100 | 1000 | n: | 10 | 100 | 1000 |
| | 0,004 | 0,015 | 0,218 | | 0,001 | 0,108 | 10,038 |
| | 0,004 | 0,014 | 0,218 | | 0 | 0,109 | 9,903 |
| | 0,004 | 0,015 | 0,218 | | 0,001 | 0,108 | 9,901 |
| | 0,004 | 0,015 | 0,218 | | 0 | 0,109 | 9,843 |
| | 0,004 | 0,015 | 0,218 | | 0,001 | 0,109 | 9,902 |
| | 0,004 | 0,015 | 0,224 | | 0 | 0,109 | 9,907 |
| | 0,004 | 0,014 | 0,219 | | 0,001 | 0,109 | 9,943 |
| | 0,004 | 0,014 | 0,218 | | 0,001 | 0,109 | 9,902 |
| | 0,004 | 0,016 | 0,218 | | 0,001 | 0,108 | 9,971 |
| | 0,004 | 0,015 | 0,224 | | 0,001 | 0,109 | 9,908 |
| Średnia: | 0,004 | 0,0148 | 0,2193 | Średnia: | 0,0007 | 0,1087 | 9,9218 |

3. Wykresy:

Przeszukiwanie grafu algorytmami BFS i DFS



Rys. 1. Złożoność obliczeniowa dla algorytmów przeszukiwania grafu dla $n = 10, 100, 1000$.

4. Wnioski:

Jak widać z wykresu, algorytm BFS jest szybszy dla małych ilości danych ($\sim O(1)$), natomiast dla dużej liczby danych, algorytmy BFS i DFS mają zbliżoną złożoność $O(n^2)$, przy czym BFS, mimo korzystania z pomocniczej struktury danych (kolejki) ma lepsze wyniki. Wyniki się zgadzają z teoretycznym przeszukiwaniem grafu wyrażonego macierzą sąsiedztwa ($O(n^2)$).

Nie udało się zmierzyć czasu dla 10^6 wierzchołków z powodu braku pamięci na maszynie wirtualnej (debugowano narzędziem Valgrind, który nie wykrył przecieków pamięci).