Piotr Stopyra 218552 Termin zajęć: PT 13:15

Przeszukiwanie grafu – algorytmy DFS i BFS

Opis zadania:

Należało zaimplementować graf nieskierowany, nieważony oraz dwa algorytmy przeszukiwania grafów: BFS i DFS, a następnie przeprowadzić pomiar czasu przeszukania grafu dla obu algorytmów oraz różnej jego wielkości, poczynając od 10, kończąc na 10^6 elementów.

Implementacja:

Kluczowym elementem programu są dwie klasy, Graph oraz Vertex. Klasa Vertex posiada atrybuty takie jak: kolor wierzchołka, jego indeks, odległość, numer indeksu poprzedniego wierzchołka (używany do przeszukania grafu) oraz lista sąsiedztwa zawierająca indeksy sąsiadujących wierzchołków. Głównym elementem klasy Graph jest dynamiczna tablica wskaźników do obiektu typu Vertex. Rozwiązanie to pozwala na odwoływanie się do wierzchołków używając ich indeksów. Jako sposób reprezentacji grafu wybrano listę sąsiedztwa, ponieważ złożoność pamięciowa tej metody wynosi O(V+E), co jest korzystniejsze od metody macierzowej, której złożoność pamięciowa wynosi $O(V^2)$. Szczególnie istotne jest to przy grafach rzadkich, czyli takich, które posiadają mało krawędzi w stosunku do wierzchołków. Ponadto użycie list sąsiedztwa daje możliwość łatwego dodania atrybutu takiego jak np. waga. W programie każdy wierzchołek posiada co najmniej dwie krawędzie.

Wyniki pomiarów:

Tabela 1 Czasy przeszukania grafu za pomocą algorytmu DFS w zależności od ilości elementów grafu podane w sekundach

Ilość elementów grafu:						
Lp.	10	10^2	10^3	10^4	10^5	10^6
1	0,000009	0,000057	0,000146	0,001931	0,032853	0,453270
2	0,000007	0,000053	0,000154	0,002610	0,038550	0,452175
3	0,000006	0,000047	0,000147	0,002314	0,038908	0,452885
4	0,000006	0,000048	0,000145	0,001571	0,036994	0,446825
5	0,000006	0,000047	0,000156	0,001547	0,036582	0,445289
6	0,000005	0,000049	0,000143	0,001622	0,038518	0,451249
7	0,000006	0,000047	0,000139	0,002143	0,039156	0,453419
8	0,000005	0,000048	0,000144	0,002049	0,036529	0,448622
9	0,000006	0,000047	0,000140	0,002046	0,036460	0,446525
10	0,000005	0,000048	0,000138	0,002037	0,038493	0,451509
Średnia:	0,0000061	0,0000491	0,0001452	0,001987	0,0373043	0,4501768

Wykres 1 Czas przeszukania grafu za pomocą algorytmu DFS w zależności od ilości elementów

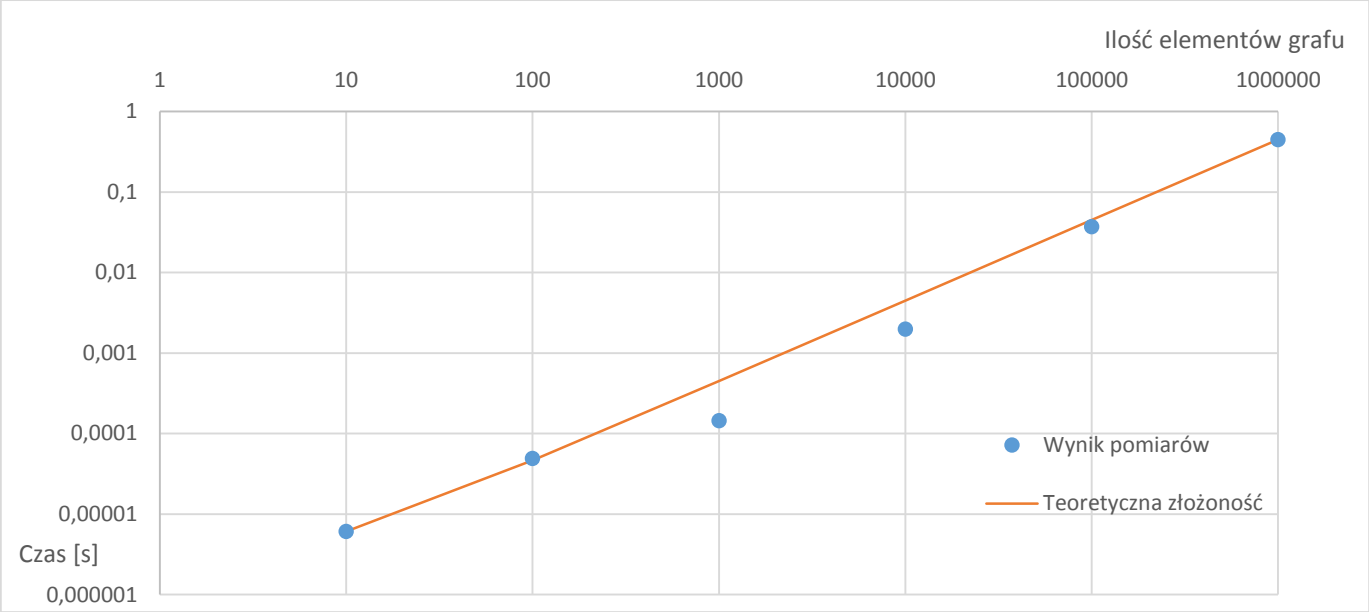
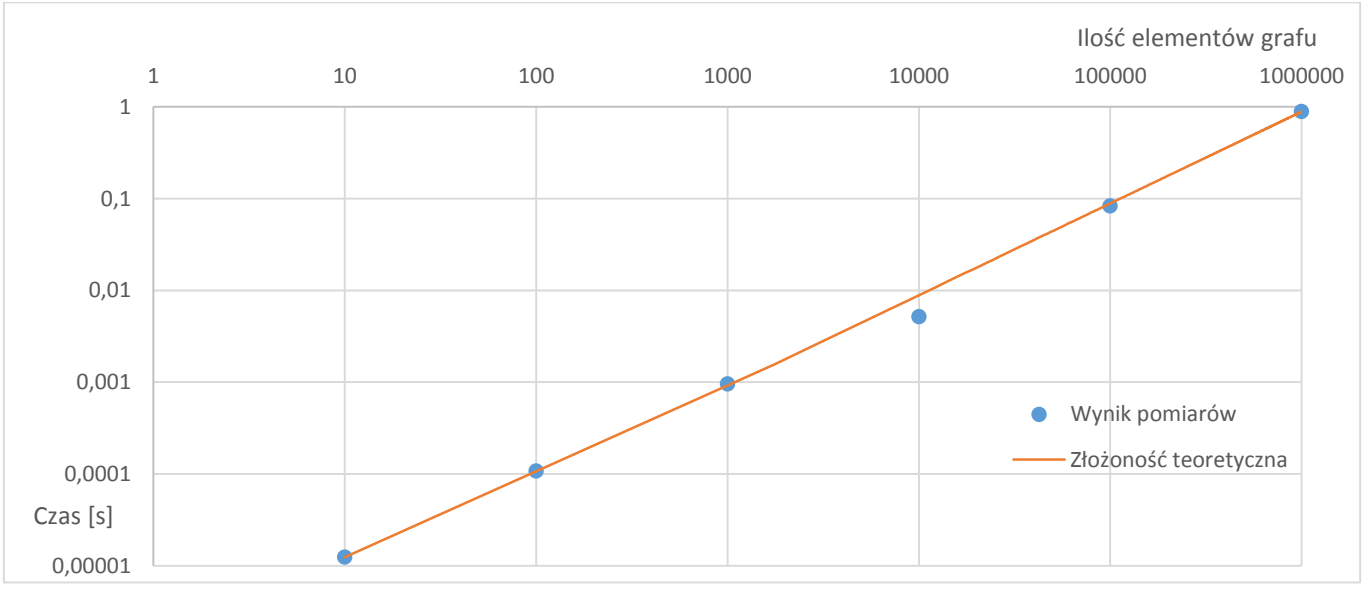


Tabela 2 Czasy przeszukania grafu za pomocą algorytmu BFS w zależności od ilości elementów grafu podane w sekundach

Ilość elementów grafu:						
Lp.	10	10^2	10^3	10^4	10^5	10^6
1	0,000015	0,000113	0,001072	0,007222	0,068694	0,829453
2	0,000013	0,000111	0,001052	0,004807	0,080153	0,842676
3	0,000013	0,000103	0,001061	0,004070	0,081366	0,839911
4	0,000012	0,000107	0,001066	0,005336	0,097350	0,830917
5	0,000011	0,000112	0,001063	0,006101	0,093640	0,973832
6	0,000010	0,000109	0,001036	0,006062	0,082401	0,908871
7	0,000017	0,000105	0,001072	0,003702	0,084331	0,910446
8	0,000012	0,000106	0,001443	0,004267	0,078685	0,906457
9	0,000011	0,000104	0,000359	0,005260	0,082882	0,910994
10	0,000010	0,000108	0,000311	0,004847	0,080346	0,905957
Średnia:	0,000012	0,000108	0,000954	0,005167	0,082985	0,885951

Wykres 2 Czas przeszukania grafu za pomocą algorytmu BFS w zależności od ilości elementów



Wnioski:

Teoretyczna złożoność obliczeniowa algorytmu BFS wynosi O(V+E) i jest taka sama jak algorytmu DFS. Osiągnięto zbliżoną do teoretycznej złożoność obliczeniową w obu przypadkach. Wyniki pomiarów wskazują na około dwukrotnie większą szybkość algorytmu DFS. Różnica może wynikać z potrzeby zapamiętania wierzchołków prowadzących z pierwszego wierzchołka w algorytmie BFS do obecnego, co jest szczególnie istotne przy dłuższych ścieżkach.