

Sprawozdanie PAMSI

17.05.2016

Zimoń Robert

Graf, algorytmy BFS i DFS

218682

Wprowadzenie:

Celem ćwiczenia było zaimplementowania struktury, jaką jest graf oraz algorytmów BFS i DFS służących do jego trawersowania, a w przypadku tego ćwiczenia ich rozszerzeń służących do znajdowania ścieżki między dwoma wierzchołkami.

Algorytm DFS to algorytm przeszukiwania grafu w głąb. Za pomocą tego algorytmu jesteśmy w stanie znaleźć drogę między wierzchołkami o ile taka istnieje, jednak prawdopodobnie nie będzie ona najkrótszą możliwą biorąc pod uwagę ilość przebytych krawędzi.

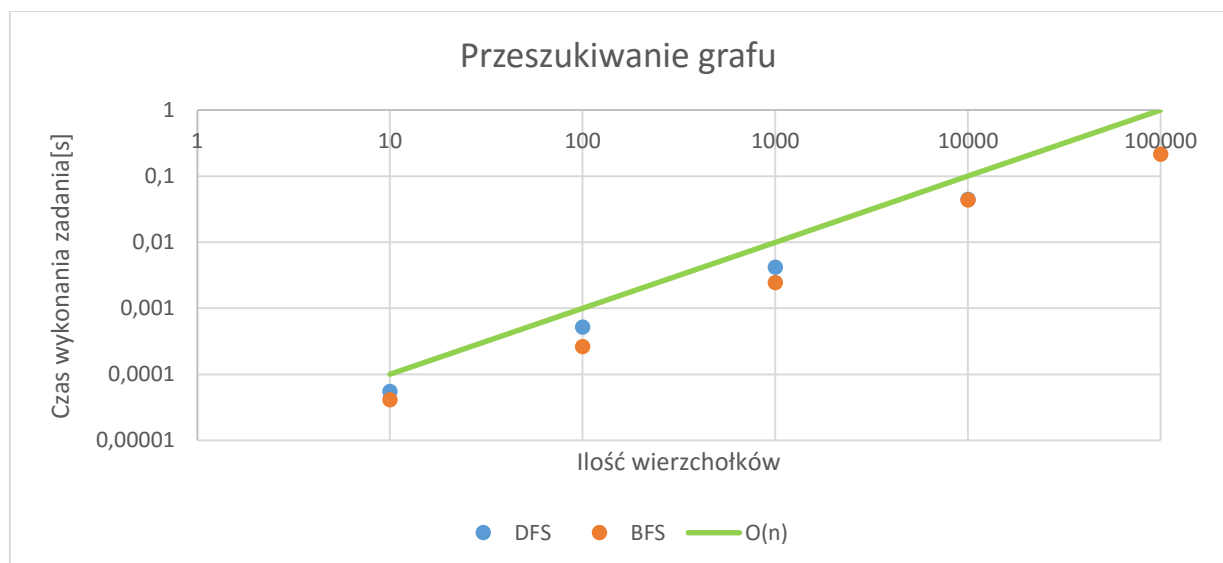
Algorytm BFS jest algorytmem przeszukiwania wszerz, co oznacza tyle, że sprawdza on wszystkie sąsiednie wierzchołki od wierzchołka startowego, a jeżeli nie odnajdzie szukanej informacji ponawia operacje dla wierzchołków już sprawdzonych. W ten sposób jesteśmy w stanie odnaleźć drogę o najmniejszej ilości krawędzi między dwoma wierzchołkami jednak kosztem zwiększonego zapotrzebowania na pamięć w stosunku do algorytmu DFS.

Oba algorytmy teoretycznie posiadają złożoność czasową przeszukiwania zależną liniowo od ilości krawędzi i wierzchołków, i w notacji dużego O wynosi $O(n+m)$, gdzie n to ilość wierzchołków, a m to ilość krawędzi.

Zaimplementowany graf przechowuje dane o posiadanych wierzchołkach i krawędziach w liście. Lista z krawędziami zawiera informację o poszczególnych wagach, które w tym ćwiczeniu nie miały znaczenia oraz informację odnośnie tego, jakie wierzchołki łączy.

Badania:

Badanie dotyczyło czasu przeszukiwania ścieżki między zadanymi wierzchołkami przez algorytmy BFS i DFS:



(ilość danych i czas przedstawione w skali logarytmicznej).

Zgodnie z teorią czas przeszukiwania dla obu zaimplementowanych algorytmów jest liniowy i wynosi $O(n)$. Badania zostały przeprowadzone dla maksymalnej ilości wierzchołków równej 100 000 w przypadku algorytmu BFS oraz 10 000 wierzchołków dla algorytmu DFS oraz proporcjonalnie rosnącej ilości krawędzi. Przy próbie działania na większym grafie w przypadku zaimplementowanego algorytmu BFS program zaczął korzystać z obszaru wymiany na dysku twardym, co znacznie spowolniło jego pracę, a algorytm DFS nie mógł działać na większej ilości danych przez błąd, po którym następował zrzut pamięci.

Podsumowanie:

- Złożoność obliczeniowa obu algorytmów jest liniowa i wynosi $O(n)$
- Algorytm BFS wyznacza ścieżkę o najmniejszej ilości krawędzi, ale kosztem zwiększonego zapotrzebowania na pamięć względem algorytmu DFS