

PAMSI - Sprawozdanie 7

Filip Guzy 218672

2 maja 2016

Graf nieskierowany

Graf jest strukturą składającą się z wierzchołków i krawędzi. Zarówno wierzchołki jak i krawędzie mogą przechowywać różnego rodzaju dane, te drugie również mogą posiadać wagę oraz kierunek. W celu przetestowania przeszukiwania grafów wgłąb (Depth-First Search) i wszerz (Breadth-First Search) zaimplementowano strukturę opartą na liście sąsiedztwa. Wybrany sposób charakteryzuje się lepszą złożonością pamięciową ($O(V + E)$), gdzie V - liczba wierzchołków, E - liczba krawędzi niż macierz sąsiedztwa ($O(V^2)$). Dużą rolę odgrywa również fakt, że zaimplementowany graf posiada stosunkowo małą ilość krawędzi w porównaniu do ilości wierzchołków, przez co wykorzystanie listy sąsiedztwa czyni operacje na nim dużo efektywniejszymi.

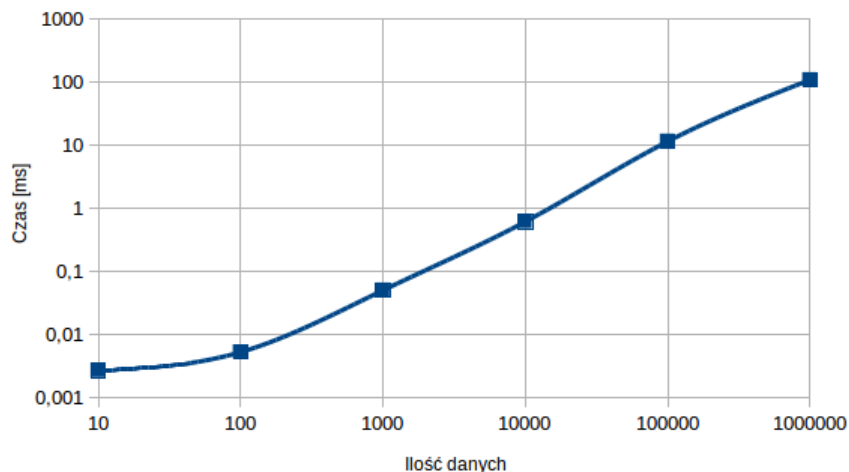
Pomiary czasów przeszukiwań DFS i BFS

W celu sprawdzenia efektywności przeszukiwań wgłąb i wszerz tworzono grafy kolejno: 10, 100, 1000, 10000, 100000 oraz 1000000 elementowe, a następnie poszukiwano w grafie nieistniejącego w nim elementu, aby zminimalizować błędy statystyczne wynikające ze zbyt szybkiego stopu algorytmu w przypadku napotkania szukanego elementu na samym początku struktury. Pierwszym etapem pomiarów było przetestowanie algorytmu przeszukiwania w głąb - DFS. Zgodnie z teorią jego złożoność czasowa powinna wynosić $O(|V| + |E|)$, zatem algorytm powinien wykonywać się w czasie stałym, co implikuje złożoność n przeszukań równą $O(n)$. Poniżej przedstawiono otrzymane wyniki w tabeli oraz na wykresie.

Tabela 1: Przeszukiwanie DFS

Ilość	10	100	1000	10000	100000	1000000
Czas [ms]	0,00200009	0,005	0,051	0,585	11,653	107,108
	0,00200009	0,00600004	0,054	0,723	10,275	109,629
	0,00300002	0,00600004	0,0500001	0,562	11,976	108,875
	0,00199997	0,005	0,049	0,633	11,918	107,714
	0,00300002	0,005	0,051	0,585	11,803	110,372
	0,00300002	0,005	0,05	0,684	12,207	109,708
	0,00300002	0,005	0,05	0,602	11,072	108,304
	0,0029999	0,005	0,049	0,642	11,004	107,896
	0,00300002	0,005	0,051	0,567	11,037	109,545
	0,0029999	0,00600004	0,05	0,576	12,035	106,833
Średnia	0,002700005	0,005300012	0,05050001	0,6159	11,498	108,5984

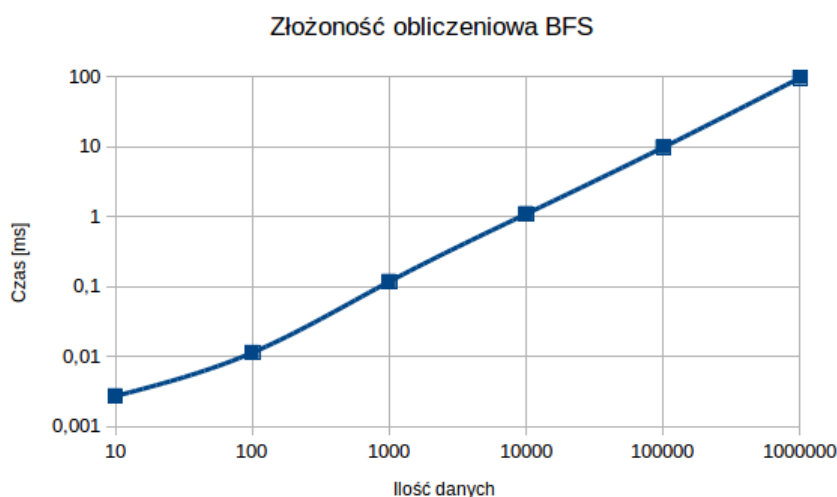
Złożoność obliczeniowa DFS



Z tabel oraz wykresów wynika, że złożoność obliczeniowa n wyszukiwań wynosi $O(n)$, zatem jest zgodna z założeniami. Kolejnym etapem pomiarów było sprawdzenie złożoności czasowej algorytmu przeszukiwania wszerz - BFS. Zgodnie z teorią jego złożoność czasowa także powinna wynosić $O(|V| + |E|)$, czyli n przeszukań powinno wykonywać się ze złożonością $O(n)$. Poniżej przedstawiono otrzymane wyniki w tabeli oraz na wykresie.

Tabela 2: Przeszukiwanie BFS

Ilość	10	100	1000	10000	100000	1000000
Czas [ms]	0,00399995	0,011	0,107	1,142	9,483	95,207
	0,00300002	0,012	0,109	1,065	10,08	99,223
	0,00199997	0,0109999	0,108	1,058	11,397	99,004
	0,00300002	0,0109999	0,106	1,098	9,946	96,385
	0,00300002	0,0120001	0,109	1,131	9,435	94,585
	0,0029999	0,0120001	0,207	1,096	9,532	96,25
	0,00300002	0,011	0,106	1,094	9,536	94,939
	0,00199997	0,0109999	0,108	1,092	9,46	96,161
	0,00199997	0,0109999	0,106	1,071	9,705	99,127
	0,00199997	0,0109999	0,106	1,073	9,08	95,152
Średnia	0,002699981	0,01129997	0,1172	1,092	9,7654	96,6033



Można zauważyć, że otrzymana złożoność obliczeniowa n przeszukiwań wynosi $O(n)$, zatem jest zgodna z założoną.

Wnioski

Wykorzystanie listy sąsiedztwa w strukturze grafu umożliwia efektywne operowanie na nim oraz zapewni wydajną złożoność pamięciową dla mało gęstych grafów. Otrzymane złożoności czasowe są zgodne z teoretycznymi, czyli $O(|V| + |E|)$ dla obu przypadków. W działaniu obu algorytmów przeszukania widać nieznaczne różnice czasowe, które wynikają ze struktury grafów, ponieważ dla różnych rozkładów krawędzi można otrzymać lepszą efektywność DFS lub BFS.