

POLITECHNIKA WROCŁAWSKA

PROJEKTOWANIE ALGORYTMÓW I METODY
SZTUCZNEJ INTELIGENCJI

PROJEKT
TERMIN: ŚR. 11:15

Projekt 2: Grafy

Autor:
Michał KUZEMCZAK

Prowadzący:
dr inż. Łukasz JELEŃ

8 maja 2019



Politechnika
Wrocławska

1 Wprowadzenie

Sprawozdanie zawiera omówienie algorytmu Dijkstry. Celem zbadania jego efektywności, zmierzono czas wykonywania algorytmu dla gęstości grafu (25%, 50%, 75%, 100%), w zależności od ilości wierzchołków w grafie (5, 10, 50, 100, 500).

Testy przeprowadzono dla reprezentacji grafu w postaci macierzy sąsiedztwa i listy sąsiedztwa.

Każdy przedstawiony czas to średnia ze 100 próbek.

2 Algorytm Dijkstry

Algorytm Dijkstry pozwala na znalezienie najkrótszych dróg do każdego wierzchołka w grafie z nieujemnymi wagami krawędzi, począwszy od wybranego wierzchołka startowego.

W testowanej implementacji użyto kolejki priorytetowej, zrealizowanej w postaci kopca. Pozwala to na uzyskanie najgorszego przypadku złożoności obliczeniowej $O(E + V \log V)$, gdzie E - liczba krawędzi, V - liczba wierzchołków. Ponadto $E = \frac{1}{2} \cdot G \cdot V \cdot (V - 1)$, gdzie G - gęstość grafu.

3 Wyniki badań

W punktach 3.1 i 3.2 przedstawiono pomiary czasu wykonywania algorytmu Dijkstry, gdzie parametrem jest gęstość grafu.

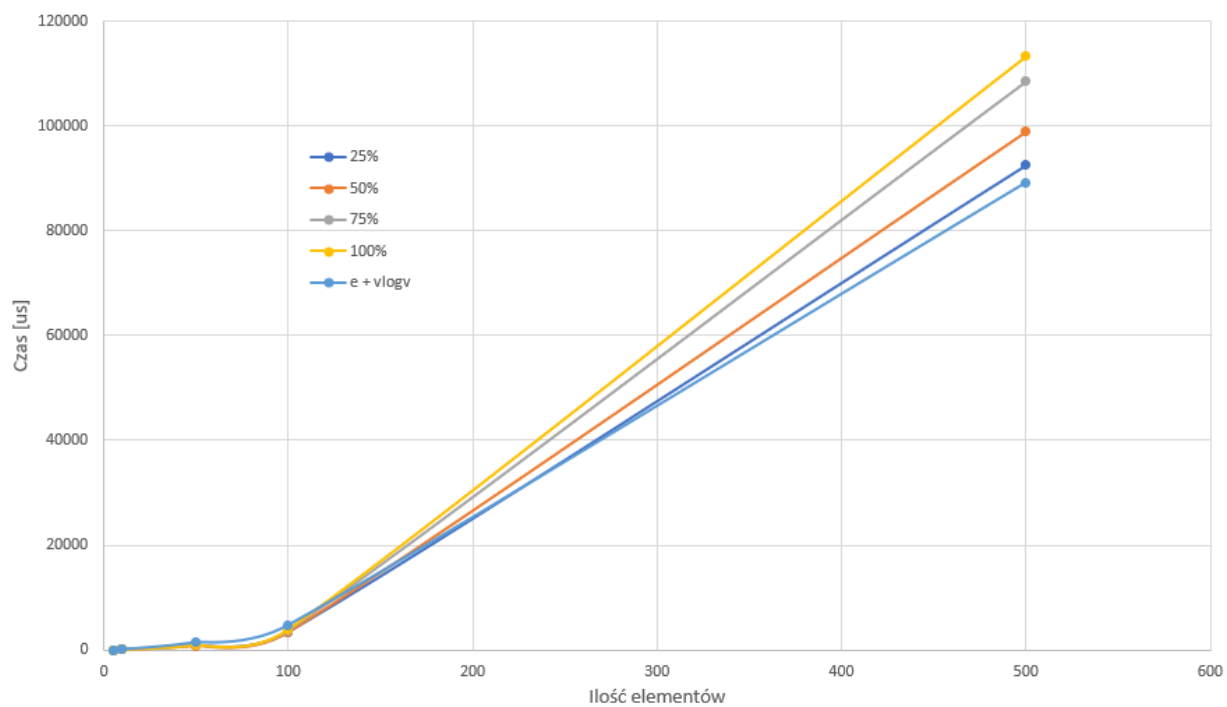
Na **Wykresie 1** i **Wykresie 2** możemy zauważyć, że czas wykonywania algorytmu jest zgodny z $O(E + V \log V)$, niezależnie od reprezentacji.

Zgodnie z oczekiwaniami, czas rośnie wraz ze wzrostem gęstości grafu. Warto zauważyć, że zmiany są większe w przypadku listy. Ma na to wpływ konstrukcja listy sąsiedztwa: zmienia się w niej liczba iteracji przez sąsiadów wierzchołka. W macierzy liczba iteracji jest stała i równa V , zmienia się jedynie odsetek z nich, który wykonuje jakieś operacje.

3.1 Macierz sąsiedztwa

	5	10	50	100	500
25%	32	127	895	3395	92571
50%	15	43	796	3363	98911
75%	15	42	888	3695	108528
100%	16	57	919	3865	113288

Tabela 1: Czas w μs . Wyniki pomiarów czasu wykonywania algorytmu dla reprezentacji grafu w postaci macierzy sąsiedztwa.

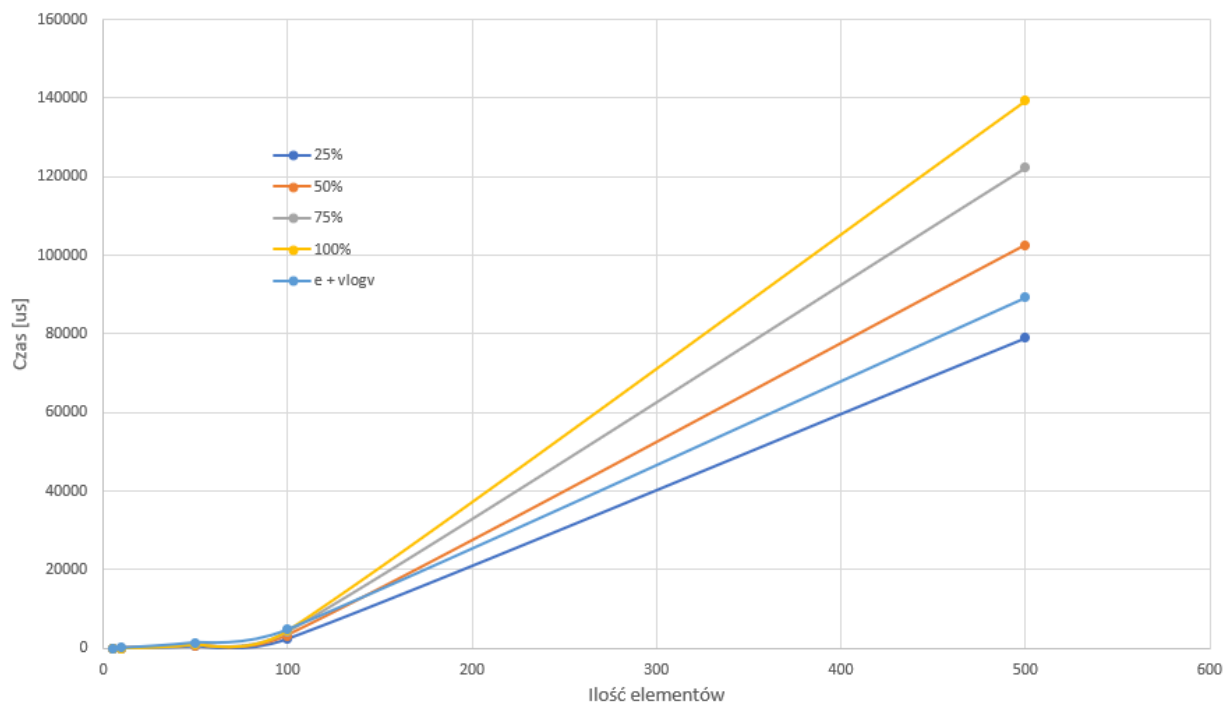


Wykres 1: Zależność czasu wykonywania algorytmu od ilości wierzchołków w grafie reprezentowanym przez macierz sąsiedztwa.

3.2 Lista sąsiedztwa

list	5	10	50	100	500
25%	15	33	543	2383	78965
50%	17	39	732	3243	102639
75%	18	46	920	4157	122283
100%	17	52	1029	4535	139314

Tabela 2: Czas w μs . Wyniki pomiarów czasu wykonywania algorytmu dla reprezentacji grafu w postaci listy sąsiedztwa.



Wykres 2: Zależność czasu wykonywania algorytmu od ilości wierzchołków w grafie reprezentowanym przez listę sąsiedztwa.

3.3 Porównanie macierzy i listy

Wykres 3 pokazuje zestawienie czasów wykonywania, gdzie parametrem jest reprezentacja grafu. Można zauważyć, że dla mniejszych wartości gęstości grafu, reprezentacja za pomocą listy jest bardziej efektywna, niż w przypadku macierzy. Sytuacja zaczyna być odwrotna wraz ze wzrostem gęstości. Żeby znaleźć powód takiego zachowania, musimy spojrzeć na różnice między reprezentacjami.

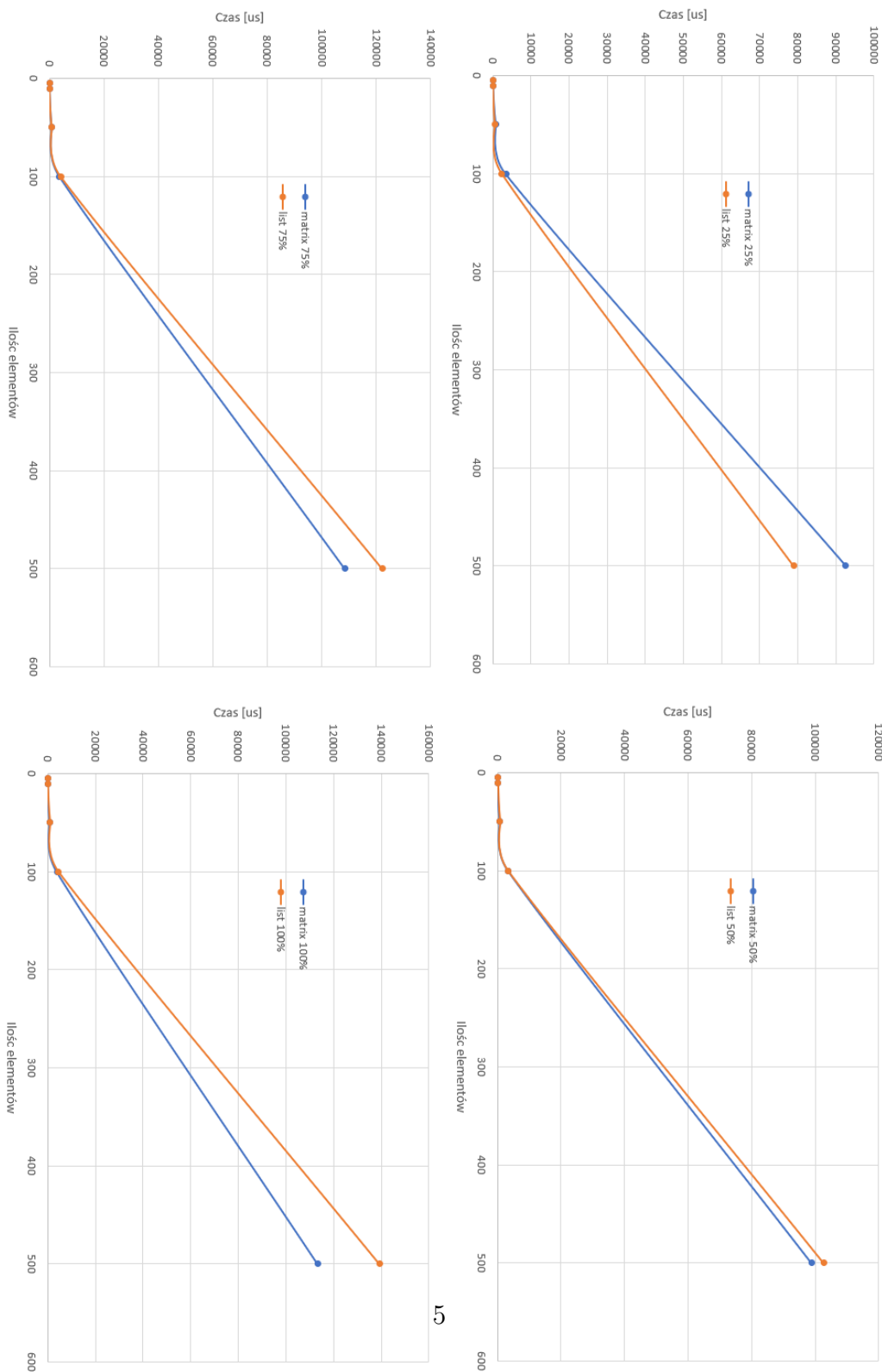
W przypadku macierzy, zaletą jest bezpośredni dostęp do każdej komórki, natomiast wadą jest to, że niezależnie od ilości sąsiadów wierzchołka, musimy przeiterować przez V elementów.

W liście sąsiedztwa zaletą jest to, że iterujemy przez tylko tyle elementów, ile sąsiadów ma wierzchołek, natomiast wadą jest utrudniony dostęp do elementów listy. Częściowym rozwiązaniem niebezpośredniego dostępu podczas przeglądania listy w pętli jest zastosowanie iteratorów, jednak liczba operacji przypadających na jedną iterację jest nadal większa niż w przypadku iteracji przez macierz.

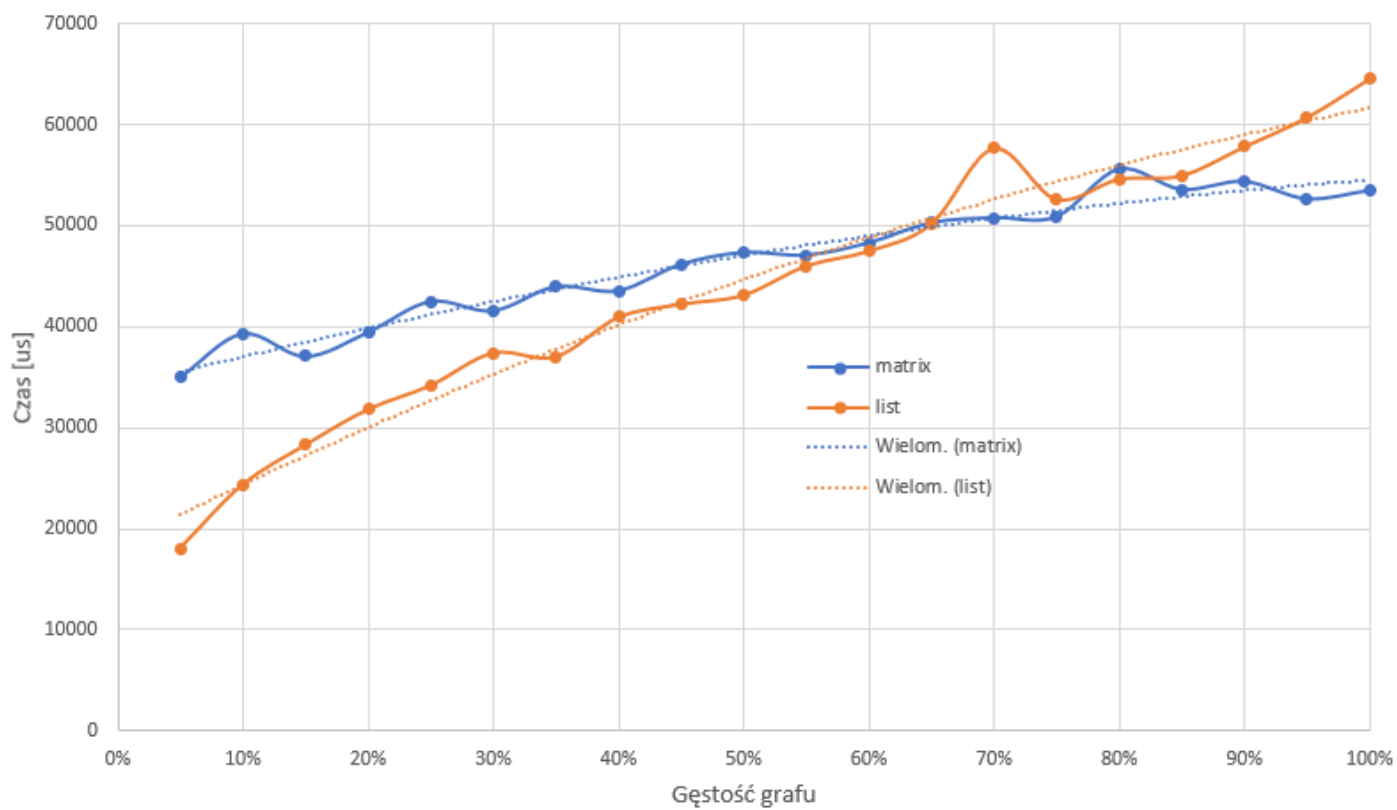
W grafach o małej gęstości wierzchołki mają mało sąsiadów. Bardziej opłacalne jest wtedy użycie listy sąsiedztwa. Liczba iteracji przez sąsiadów wierzchołka jest wtedy znikoma.

Przy dużej gęstości uzasadnione jest zastosowanie macierzy sąsiedztwa. W ilości iteracji przez sąsiadów, w porównaniu do listy, nie ma wtedy różnicy, za to zyskujemy na prędkości pętli.

Na **Wykresie 4** widzimy, jak dla małych gęstości lista góruje w wydajności.



Wykres 3: Zbiorcze porównanie czasów dla listy i macierzy.



Wykres 4: Porównanie czasu wykonywania dla listy i macierzy w zależności od gęstości grafu, dla $V = 300$.

4 Bibliografia

- https://pl.wikipedia.org/wiki/Algorytm_Dijkstry, ostatnie użycie: 18 kwietnia 2019;
- https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm, ostatnie użycie: 18 kwietnia 2019;
- https://eduinf.waw.pl/inf/alg/001_search/0138.php, ostatnie użycie: 17 kwietnia 2019.