



Politechnika Wrocławska

Projektowanie i Analiza Algorytmów

Projekt 2 - Grafy

Mateusz Marko ISA nr 273168

Grupa nr 8 – Poniedziałki 18:55-20:35

6.05.2024

Spis treści

1.	Wprowadzenie	3
•	Cel zadania	3
•	Reprezentacje	3
•	Dane do badań	3
2.	Przebieg Badań i Wyniki	3
a)	Gęstość Grafu 25%	4
b)	Gęstość Grafu 50%	5
c)	Gęstość Grafu 75%	6
d)	Gęstość Grafu 100%	7
3.	Wnioski	8
4.	Bibliografia	8

1. Wprowadzenie

- **Cel zadania**

Zadanie polegało na zbadaniu różnych reprezentacji grafu dla różnych gęstości i ilości wierzchołków w kontekście efektywności czasowej wybranego algorytmu (w moim przypadku był to algorytm Dijkstry). Algorytm ten jest jednym z najlepszych w kontekście rozwiązywania problemu najkrótszej ścieżki w grafach ważonych z nieujemnymi wagami. Jednakże efektywność jego działania jest ściśle związana ze strukturą danych, na której jest zaimplementowany. Rozważane będą reprezentacje w postaci Listy sąsiedztwa oraz Macierzy sąsiedztwa.

- **Reprezentacje**

- a) **Lista sąsiedztwa**

składa się z tablicy list, gdzie i -ta lista zawiera wszystkich sąsiadów i -tego wierzchołka. Lista sąsiedztwa jest bardziej efektywna pod względem pamięci i jest idealna dla grafów rzadkich.

- b) **Macierz sąsiedztwa**

składa się z $n \times n$ macierzy (gdzie n to liczba wierzchołków), w której wartość w i -tym wierszu i j -tej kolumnie reprezentuje wagę krawędzi między wierzchołkami i i j . Macierz sąsiedztwa jest łatwa do implementacji i idealna do grafów gęstych, ale wymaga dużo pamięci i nie jest efektywna pod względem czasu dla grafów rzadkich.

- **Dane do badań**

Badania efektywności algorytmu Dijkstry zostały przeprowadzone z wykorzystaniem dwóch głównych reprezentacji grafu: listy sąsiedztwa oraz macierzy sąsiedztwa. Celem eksperymentów była analiza wpływu różnych gęstości grafu na czas wykonania algorytmu. Gęstość grafu definiuje stosunek liczby krawędzi do maksymalnej możliwej liczby krawędzi w grafie. Badania przeprowadzono dla pięciu różnych rozmiarów grafu: 10, 50, 100, 500, 1000 wierzchołków. Program został napisany obiektowo z podziałem na pliki nagłówkowe, a procesor w komputerze na którym były wykonywane badania to ośmiorzeniowy AMD Ryzen 7 2700.

2. Przebieg Badań i Wyniki

Dla każdego zestawu parametrów (liczba wierzchołków i gęstość grafu), wygenerowano 100 losowych instancji grafów, aby uzyskać reprezentatywną próbę wyników. Dla każdej instancji grafu, przeprowadzono kolejno: Generowanie grafu za pomocą specjalnie przygotowanej funkcji „randG”, graf był tworzony z określoną liczbą wierzchołków i gęstością. Funkcja ta losowo rozdzielała wagi krawędzi między wierzchołkami, następnie algorytm był uruchamiany na grafie reprezentowanym przez listę/macierz sąsiedztwa, mierząc czas wykonania od momentu startu do zakończenia. Czasy wykonania dla obu reprezentacji grafu były rejestrowane i uśredniane po przeprowadzeniu 100 iteracji dla każdego zestawu parametrów.

Zakresy Gęstości

25% - grafy rzadkie, gdzie liczba krawędzi wynosi 25% maksymalnej możliwej liczby krawędzi.

50% - grafy o średniej gęstości, z połową możliwych krawędzi.

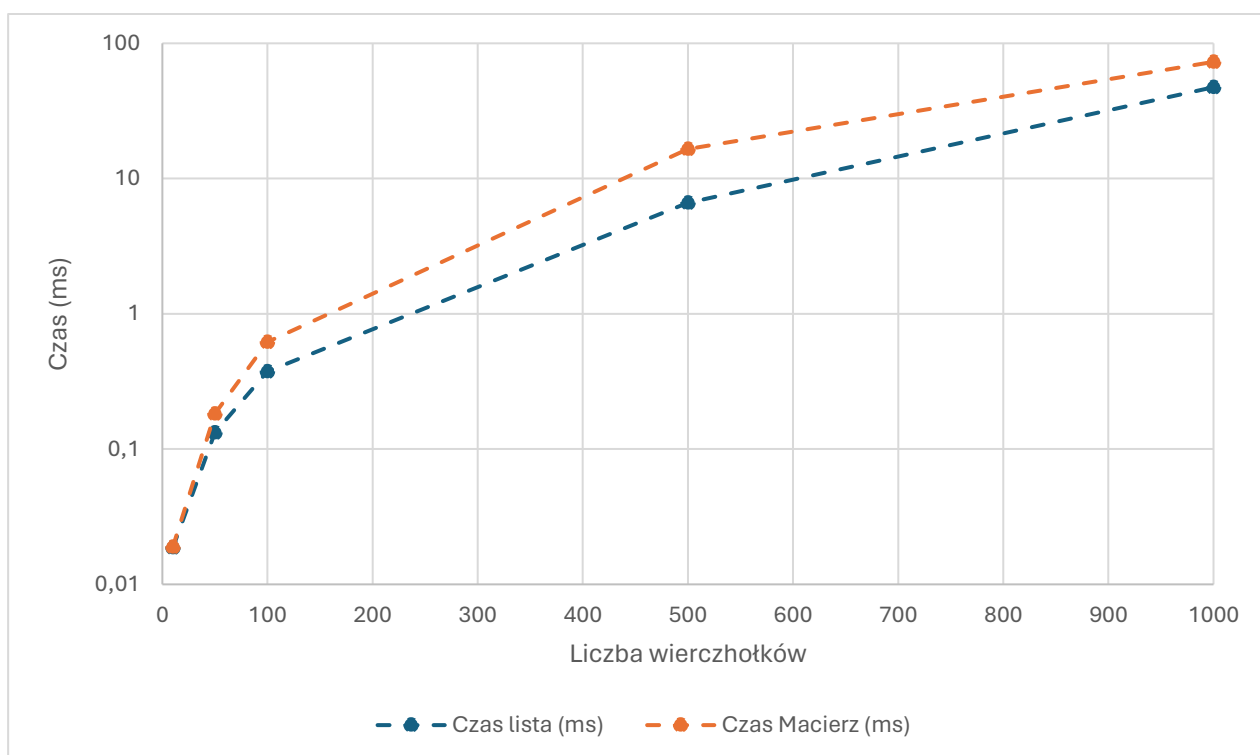
75% - grafy dość gęste, gdzie 75% możliwych krawędzi jest obecnych.

100% - grafy pełne, gdzie każdy wierzchołek jest połączony z każdym innym wierzchołkiem.

a) Gęstość Grafu 25%

Tabela 1 Czas wykonania algorytmu Dijkstry zależny od ilości wierzchołków dla gęstości 25%

Ilość wierzchołków	Czas lista (ms)	Czas Macierz (ms)
10	0.018668	0.018962
50	0.132569	0.182308
100	0.374288	0.617246
500	6.62517	16.5119
1000	47.3872	72.931

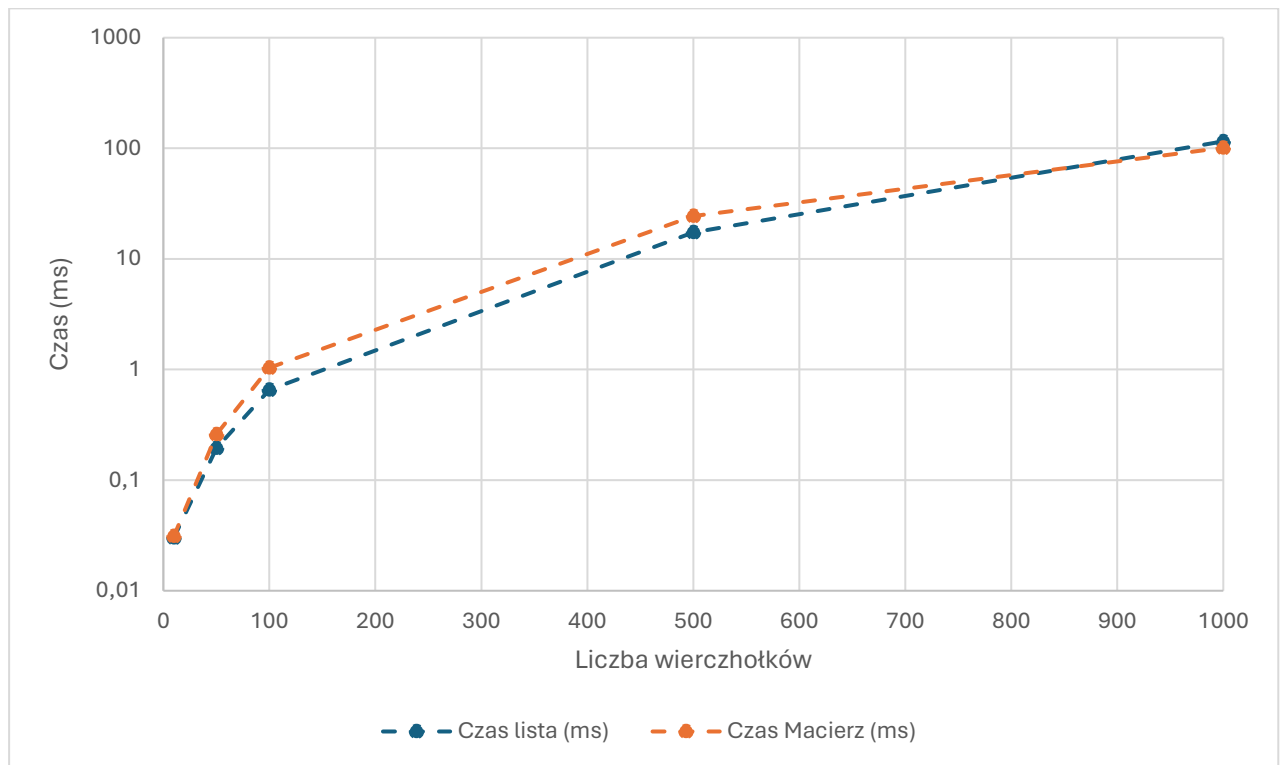


Rysunek 1 Wykres zależności czasu od ilości wierzchołków dla gęstości 25% dla reprezentacji macierzą sąsiedztwa oraz listą sąsiedztwa

b) Gęstość Grafu 50%

Tabela 2 Czas wykonania algorytmu Dijkstry zależny od ilości wierzchołków dla gęstości 50%

Ilość wierzchołków	Czas lista (ms)	Czas Macierz (ms)
10	0.030189	0.031187
50	0.195142	0.258469
100	0.65244	1.03914
500	17.377	24.3839
1000	115.537	101.218

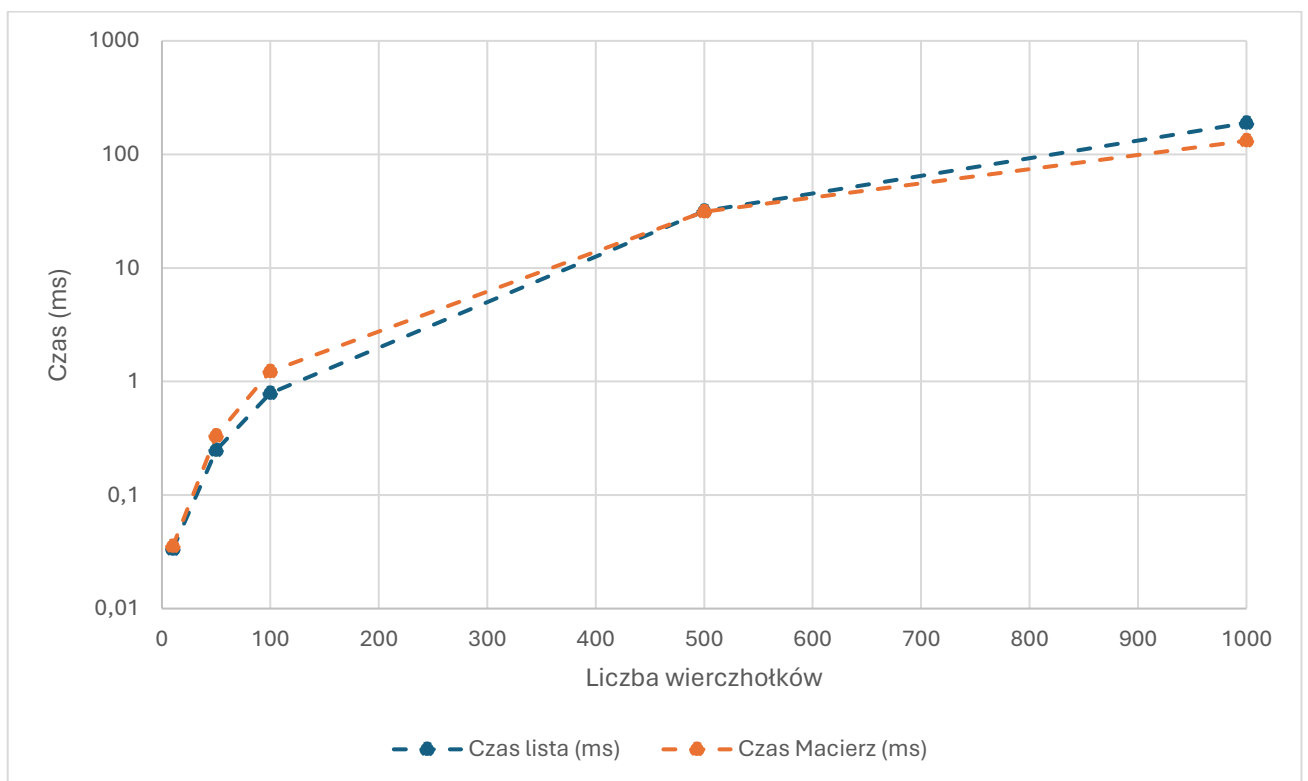


Rysunek 2 Wykres zależności czasu od ilości wierzchołków dla gęstości 50% dla reprezentacji macierzą sąsiedztwa oraz listą sąsiedztwa

c) Gęstość Grafu 75%

Tabela 3 Czas wykonania algorytmu Dijkstry zależny od ilości wierzchołków dla gęstości 75%

Ilość wierzchołków	Czas lista (ms)	Czas Macierz (ms)
10	0.033297	0.035325
50	0.247499	0.333457
100	0.787162	1.22832
500	31.7074	31.273
1000	189.248	131.858

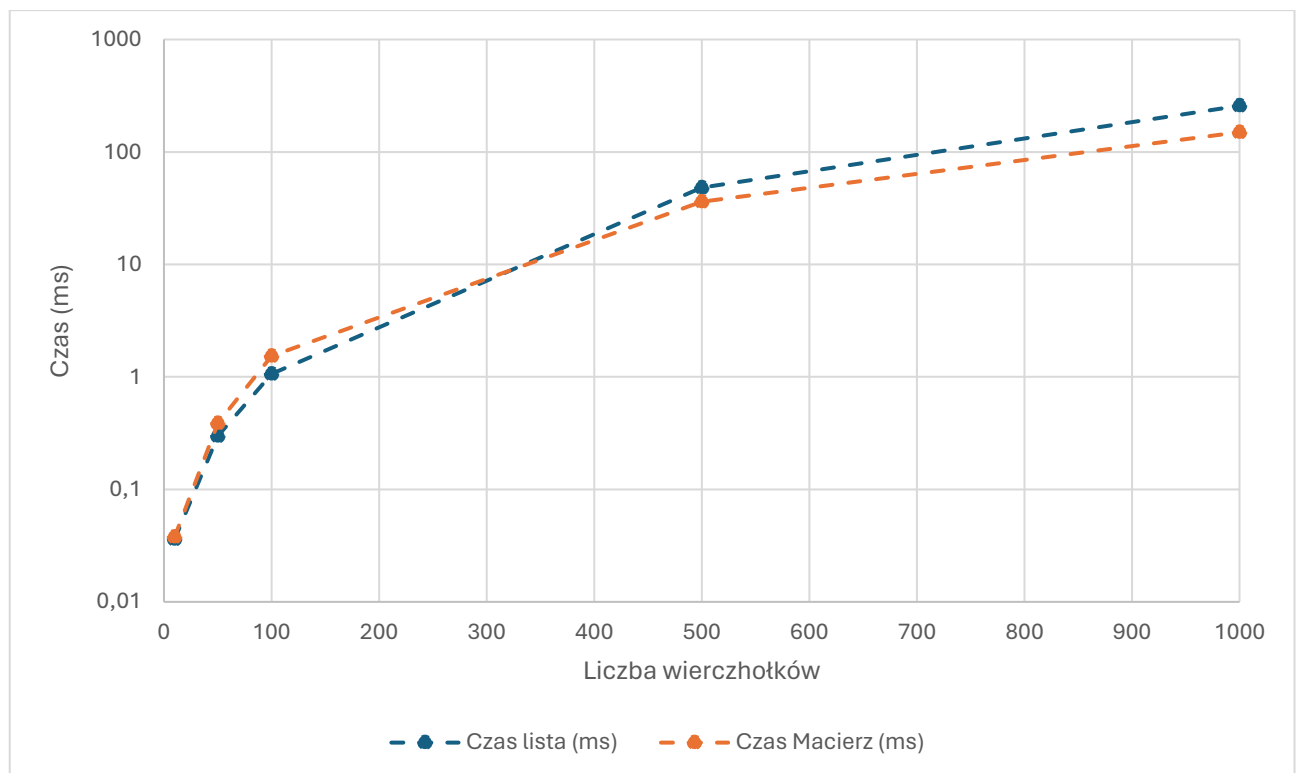


Rysunek 3 Wykres zależności czasu od ilości wierzchołków dla gęstości 75% dla reprezentacji macierzą sąsiedztwa oraz listą sąsiedztwa

d) Gęstość Grafu 100%

Tabela 4 Czas wykonania algorytmu Dijkstry zależny od ilości wierzchołków dla gęstości 100% (pełny)

Ilość wierzchołków	Czas lista (ms)	Czas Macierz (ms)
10	0.036173	0.037919
50	0.297643	0.388838
100	1.06656	1.53438
500	48.0768	36.1348
1000	257.394	149.813



Rysunek 4 Wykres zależności czasu od ilości wierzchołków dla gęstości 100% dla reprezentacji macierzą sąsiedztwa oraz listą sąsiedztwa

3. Wnioski

- Czas wykonania algorytmu rośnie wraz ze wzrostem liczby wierzchołków, co jest zgodne z oczekiwaniami, ponieważ większa liczba wierzchołków generuje więcej potencjalnych ścieżek do przetworzenia. Wzrost jest ciut bardziej wyraźny w przypadku listy sąsiedztwa, co sugeruje, że dla dużych grafów reprezentacja macierz sąsiedztwa jest bardziej efektywna czasowo.
- Czas wykonania algorytmu również rośnie wraz ze wzrostem gęstości grafu, ale ten wzrost jest bardziej zauważalny w reprezentacji macierzy. Przy największej gęstości (100%) różnica czasu wykonania między listą a macierzą zmniejsza się, co może wskazywać na to, że w przypadku grafów bardzo gęstych narzut zarządzania większą liczbą krawędzi staje się podobny dla obu reprezentacji.
- Lista sąsiedztwa wykazuje lepszą wydajność czasową dla mniejszych konfiguracji, co jest spowodowane mniejszym narzutem przy przeglądaniu sąsiadów danego wierzchołka, szczególnie w rzadszych grafach. Macierz sąsiedztwa, choć szybsza w dostępie do konkretnych połączeń, wymaga przeglądania większej liczby elementów, co wpływa na czas wykonania, szczególnie gdy graf staje się gęstszy i większy lecz także sobie radzi a czasem nawet wychodzi lepiej czasowo niż lista.
- Wyniki pokazują, że wybór struktury danych do reprezentacji grafu ma znaczący wpływ na wydajność algorytmów grafowych zwłaszcza w kontekście złożoności obliczeniowej i skalowalności. Dla małych lub rzadkich grafów reprezentacja listy może być bardziej optymalna, natomiast dla grafów gęstych różnice w wydajności między reprezentacjami zmniejszają się. Wybór więc powinien być podejmowany na podstawie specyfiki problemu, rozmiaru danych i wymaganej wydajności.

4. Bibliografia

- <http://www.algorytm.org/algorytmy-grafowe/algorytm-dijkstry.html>
- <http://www.algorytm.org/klasyczne/grafy-i-ich-reprezentacje/grafy-1-c.html>
- https://eduinf.waw.pl/inf/alg/001_search/0138.php
- <https://pl.khanacademy.org/computing/computer-science/algorithms/graph-representation/a/representing-graphs>
- <https://home.agh.edu.pl/~horzyk/lectures/wdi/WDI-Grafy.pdf>