

Data: 02.05.2020

Imię i nazwisko: Janusz Domaradzki

Nr albumu: 249024

Dane prowadzącego: mgr inż. Marta Emirsajłow

Termin zajęć: Piątek 13:15-14:45

# Projektowanie Algorytmów i Metody Sztucznej Inteligencji

## Projekt 2 – Grafy

### 1. Wprowadzenie

Celem projektu było przetestowanie działania jednego z algorytmów wyznaczających najkrótszą ścieżkę z danego wężła grafu (ważonego oraz skierowanego) do każdego z innych wężłów.

Grafy używane do testów miały różne ilości wężłów, gęstości (ilości krawędzi) oraz reprezentacje grafu.

Testowane ilości wężłów:

- 5
- 15
- 30
- 50
- 100

Testowane gęstości:

- 25%
- 50%
- 75%
- 100 % (graf pełny)

Gęstość grafu skierowanego wyraża się wzorem  $D = \frac{|E|}{|V| \cdot (|V|-1)}$ , gdzie E jest ilością krawędzi grafu, a V – ilością wężłów grafu. Wzór ten jest potrzebny do policzenia ilości krawędzi grafu dla danej gęstości.

Testowane reprezentacje grafu:

- Macierz sąsiedztwa
- Lista sąsiedztwa

W projekcie testowany był algorytm **Bellmana-Forda**. Algorytm ten był sprawdzany na reprezentatywnej grupie 100 grafów dla każdej gęstości i ilości węzłów.

## 2. Opis algorytmu

Algorytm Bellmana-Forda jest wolniejszy od algorytmu Dijkstry, jednak bardziej uniwersalny. Jest on w stanie bowiem obsługiwać grafy z ujemnymi wartościami krawędzi oraz wykrywać ujemne cykle, które mogły zostać stworzone właśnie przez ujemne wartości krawędzi.

Złożoność obliczeniowa:

- Reprezentacja w postaci listy sąsiedztwa:  $O(VE)$
- Reprezentacja w postaci macierzy sąsiedztwa:  $O(V^3)$

Złożoność pamięciowa:

- Reprezentacja w postaci listy sąsiedztwa:  $O(V+E)$
- Reprezentacja w postaci macierzy sąsiedztwa:  $O(V+V^2) = O(V^2)$

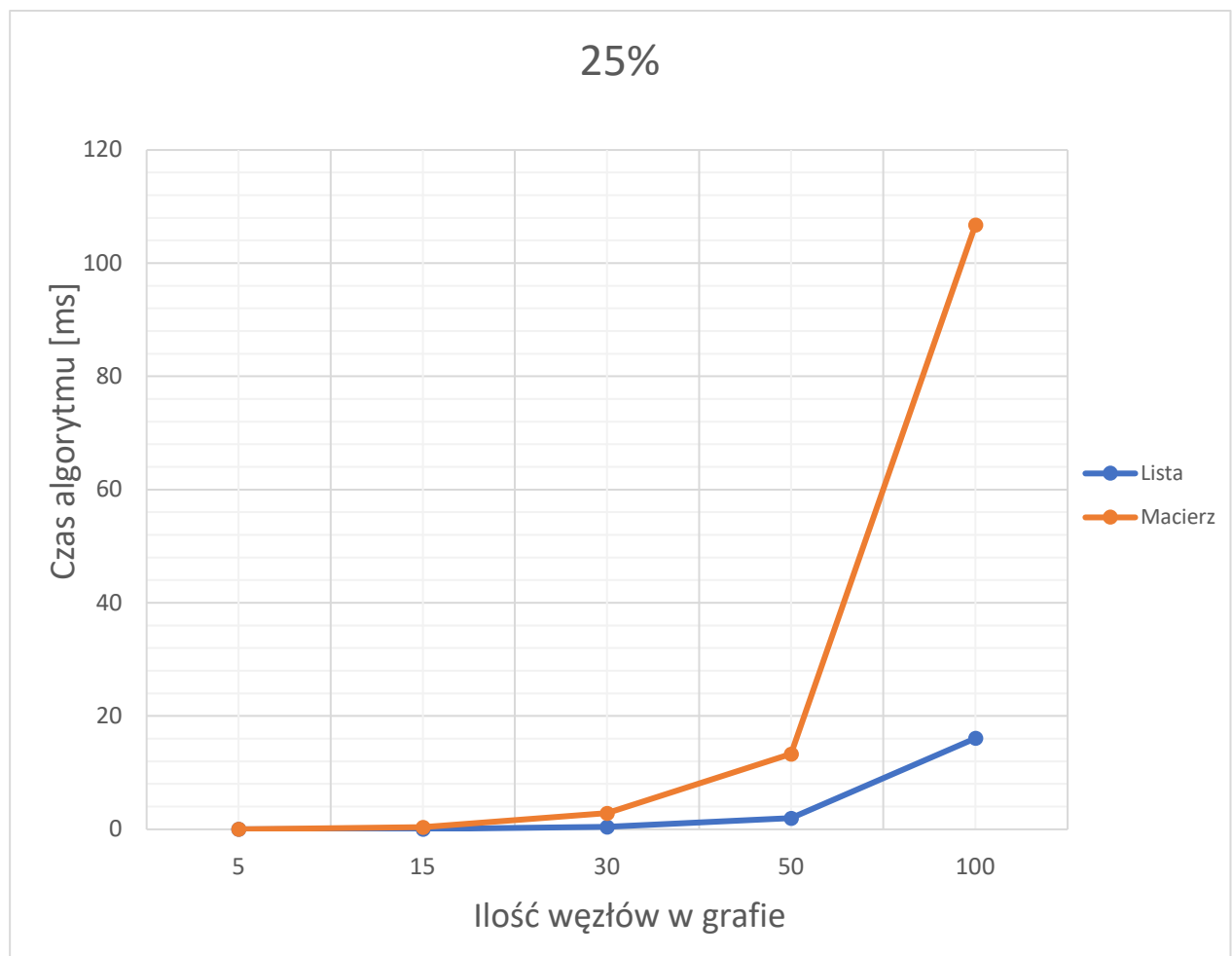
### 3. Wyniki testów

Czasy algorytmu grafów mierzone były w milisekundach. W sprawozdaniu podano wyniki uśrednione. (t.j. czas wykonywania algorytmu na jednym grafie)

#### a) Wykresy dla parametru w postaci reprezentacji grafu

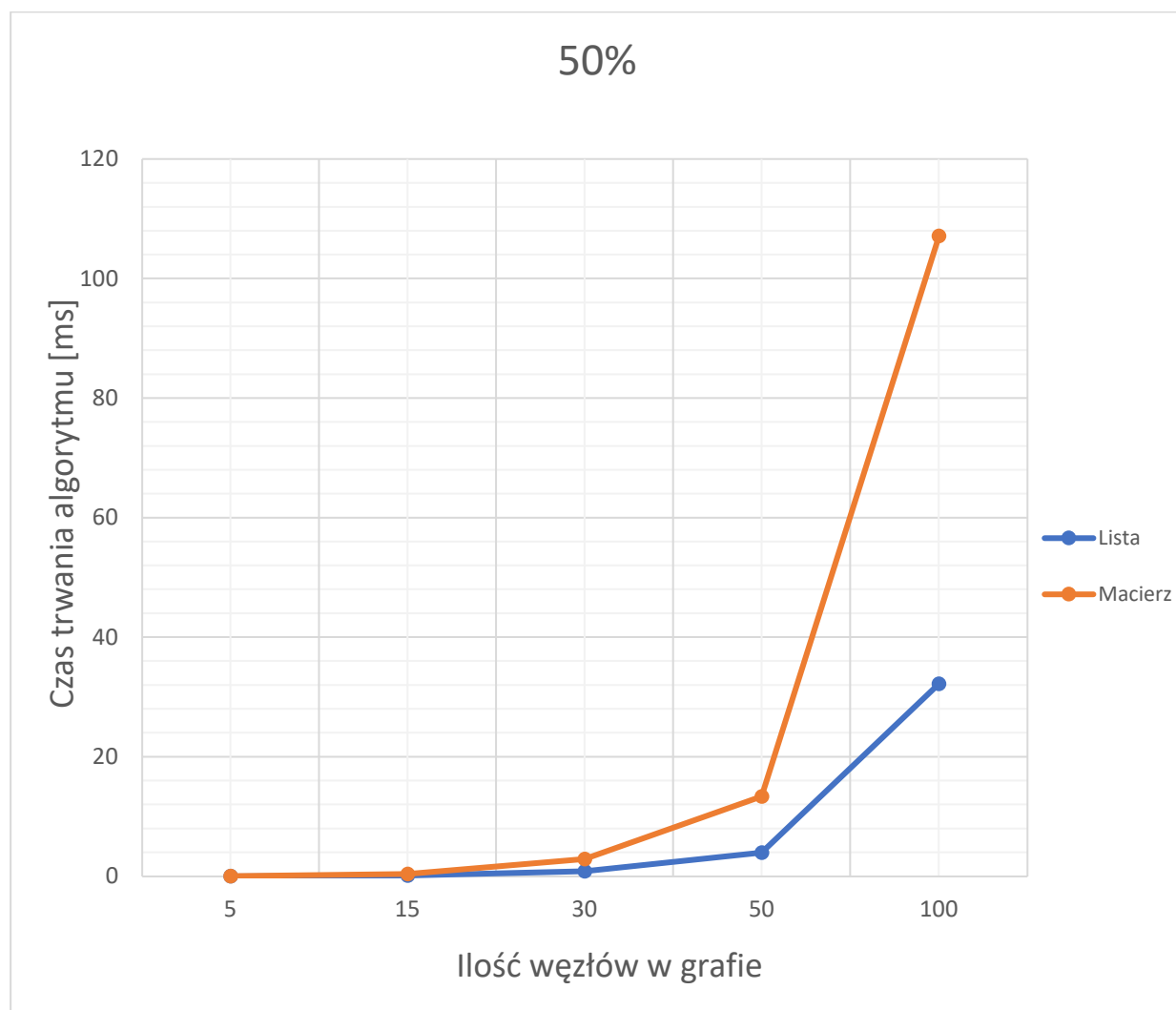
- 25%

Gęstość grafu	25%				
Ilość węzłów	5	15	30	50	100
Lista	0,00555	0,05288	0,42378	1,9775	16,0957
Macierz	0,01234	0,35496	2,8474	13,2803	106,778



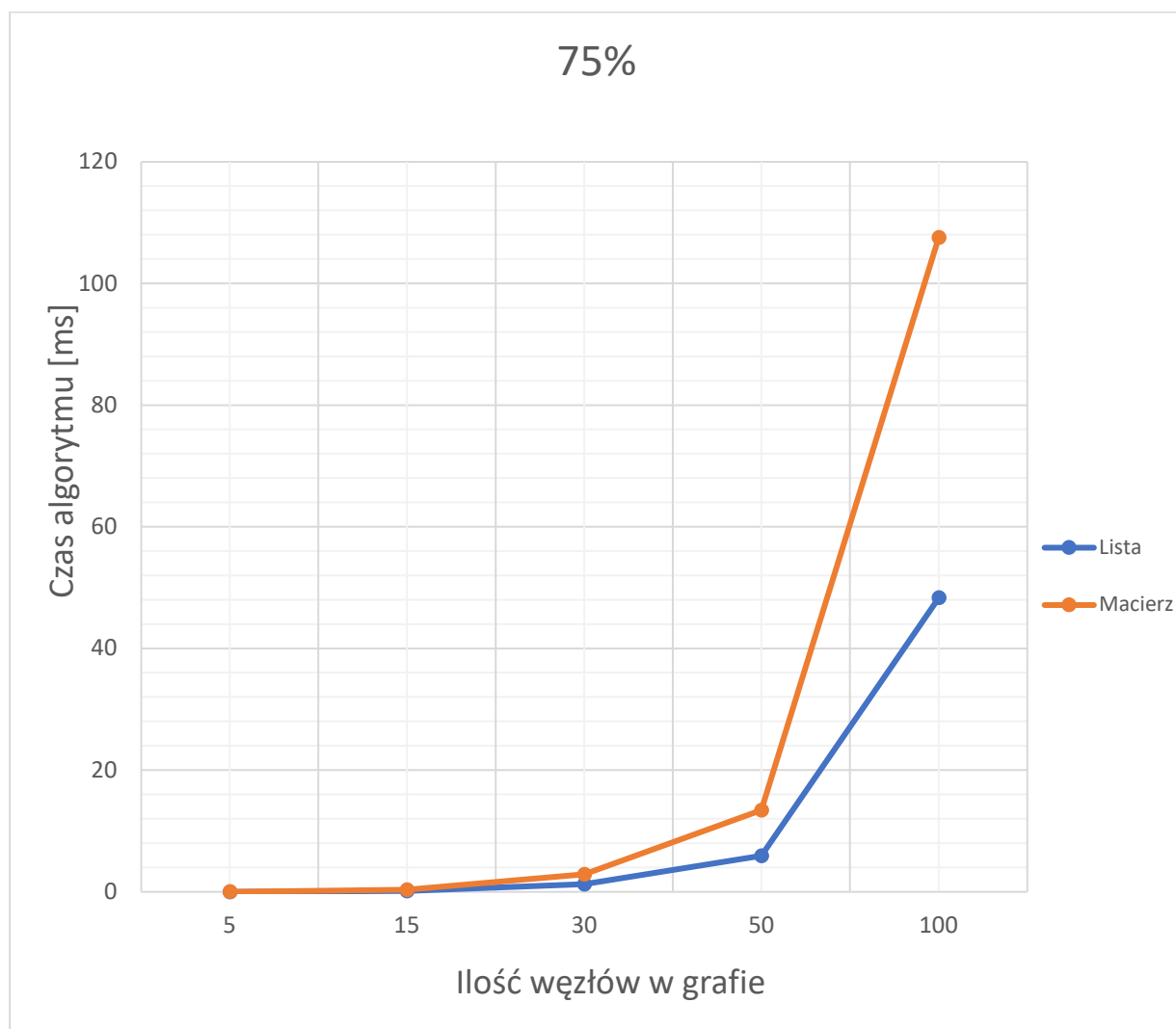
- 50%

Gęstość grafu	50%				
Ilość węzłów	5	15	30	50	100
Lista	0,00395	0,09766	0,8322	3,9519	32,2044
Macierz	0,01235	0,35227	2,8625	13,3319	107,104



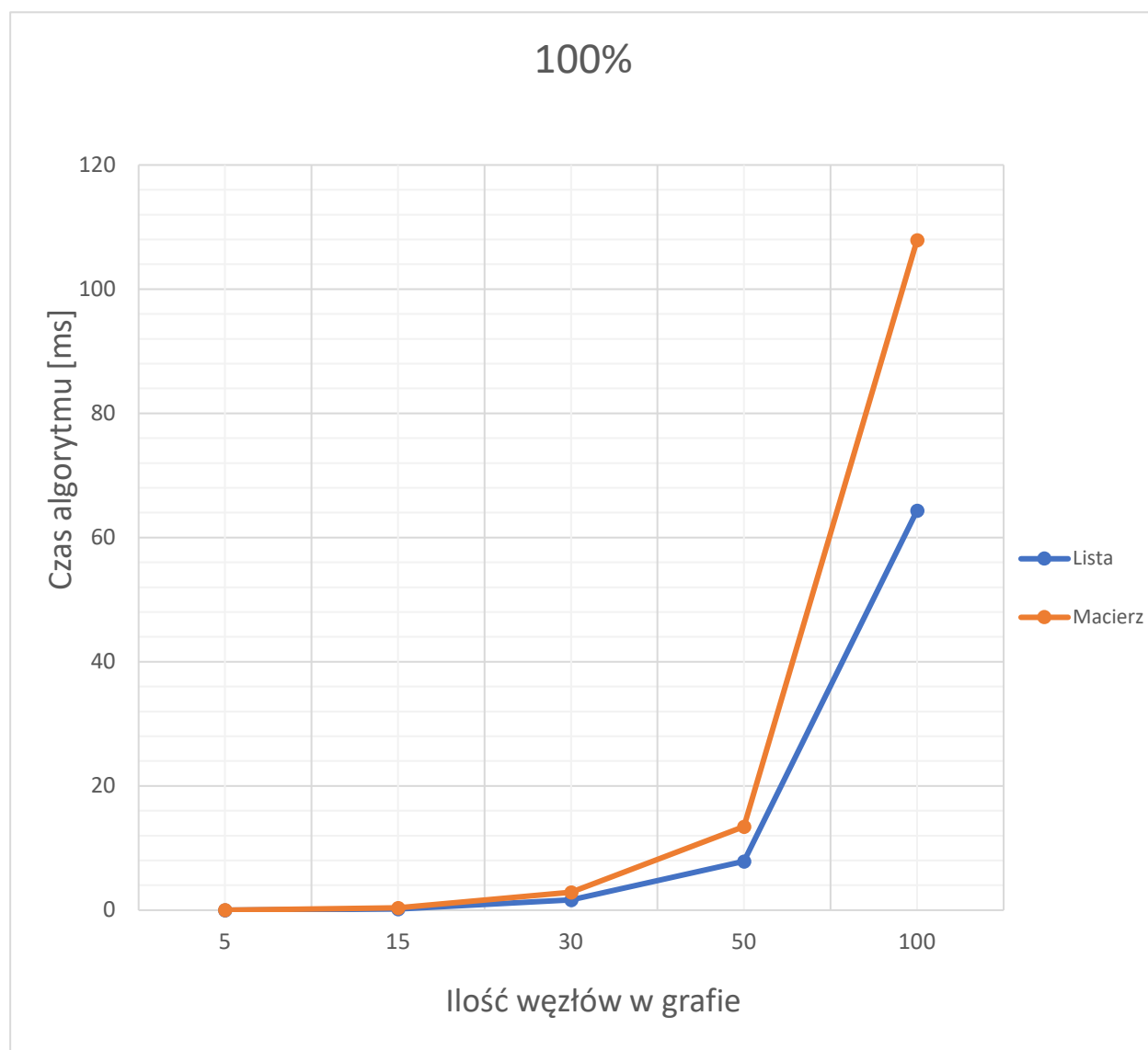
- 75%

Gęstość grafu	75%				
Ilość węzłów	5	15	30	50	100
Lista	0,00581	0,14522	1,24404	5,9053	48,3197
Macierz	0,012351	0,35422	2,8736	13,3991	107,572

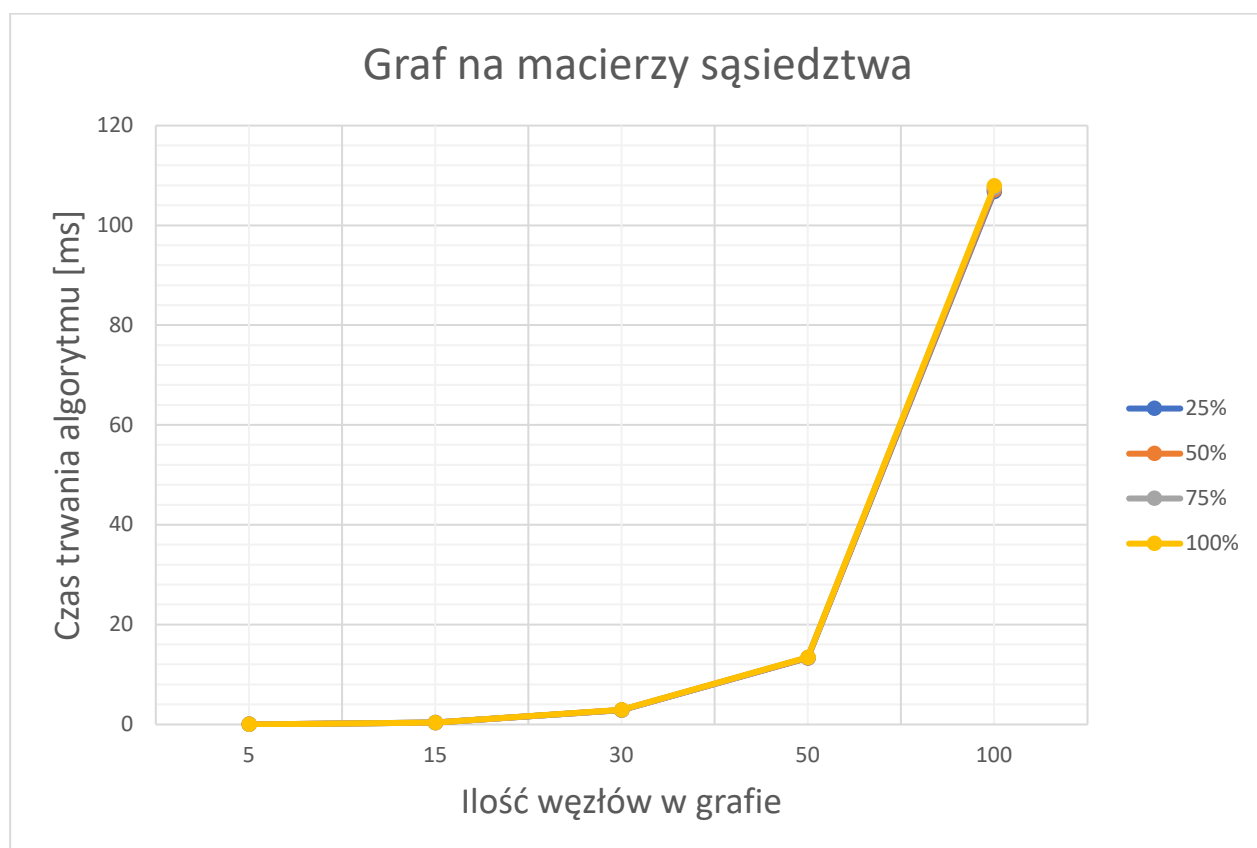
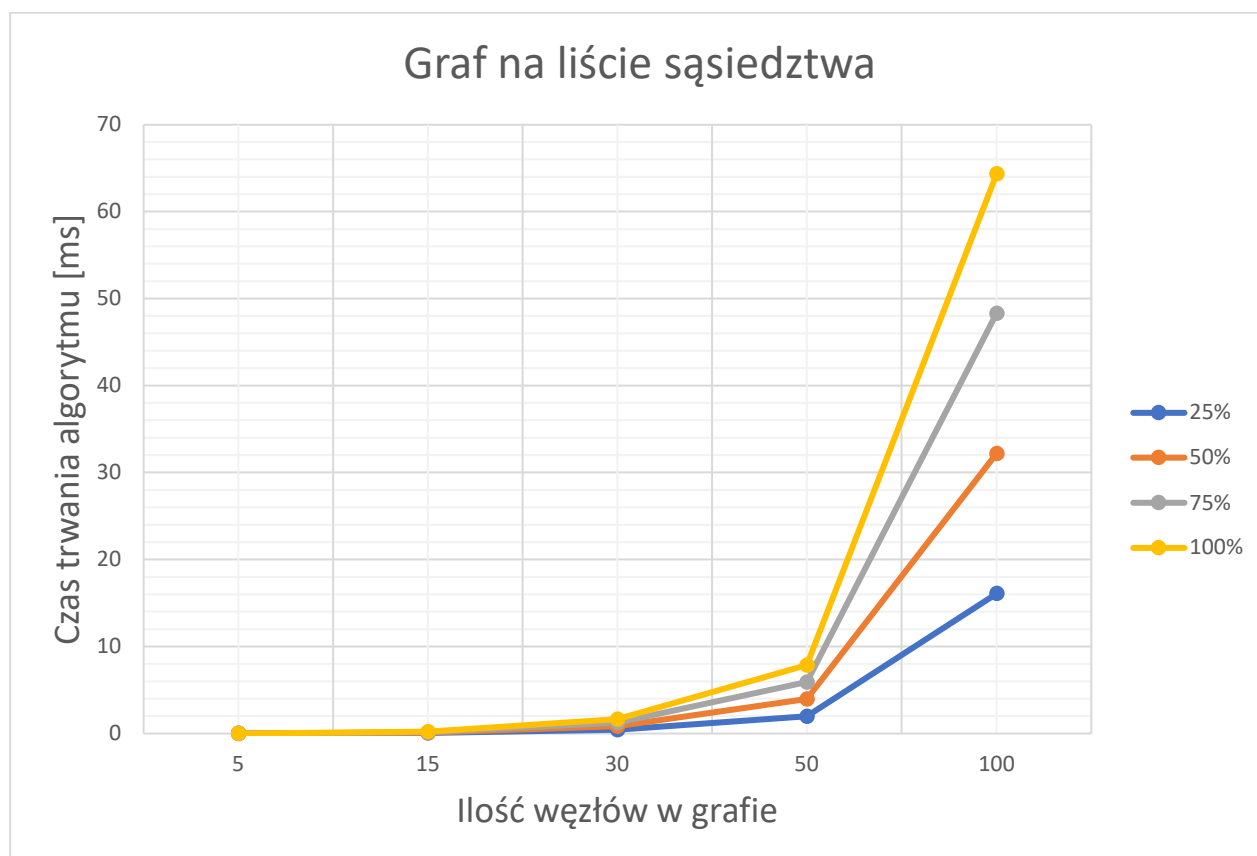


- 100%

Gęstość grafu	100%				
Ilość węzłów	5	15	30	50	100
Lista	0,00556	0,19461	1,66021	7,8697	64,3504
Macierz	0,015057	0,35902	2,8906	13,442	107,911



## b) Wykresy dla parametru w postaci gęstości grafu



## 4. Wnioski

Algorytm wykonuje się dłużej dla grafu reprezentowanego na macierzy sąsiedztwa niż na liście sąsiedztwa. Różnica ta staje się bardziej wyraźna wraz ze wzrostem ilości węzłów w grafie.

Gęstość grafu nie ma większego wpływu w realizacji algorytmu dla grafu reprezentowanego na macierzy sąsiedztwa – czasy wykonywania algorytmu są do siebie bardzo zbliżone.



(Zbliżenie na wykres „Graf na macierzy sąsiedztwa” ukazujący niewielką różnicę czasów w realizacji algorytmu)

Gęstość grafu ma wpływ w realizacji algorytmu dla grafu reprezentowanego na liście sąsiedztwa – im większa gęstość, tym algorytm wykonuje się dłużej. Różnice czasów są dużo bardziej widoczne niż w przypadku grafu reprezentowanego na macierzy sąsiedztwa.