

Sprawozdanie 2

„Projektowanie algorytmów i metod sztucznej inteligencji”

8 maja 2019

Temat projektu: Grafy

Autor : Paweł Gajda

Termin zajęć: Środa 7.30-9.00

Prowadzący: Dr inż. Łukasz Jeleń

1. Wprowadzenie

1.1 Opis projektu

Projekt opierał się na przetestowaniu działania jednego z algorytmów wyznaczającego najkrótszą drogę z danego wężła grafu (grafu ważonego oraz skierowanego) do każdego innego z wężłów.

Algorytm, który będzie testowany:

- Algorytm Bellmana-Forda

Grafy na których będzie testowany algorytm będą różnić się gęstością (ilością krawędzi), ilością wężłów oraz samą reprezentacją grafu. Testowane reprezentacje grafu:

- Macierz sąsiedztwa
- Lista sąsiedztwa

Testowane gęstości:

- 25%
- 50%
- 75%
- 100% (graf pełny)

Testowane ilości wężłów:

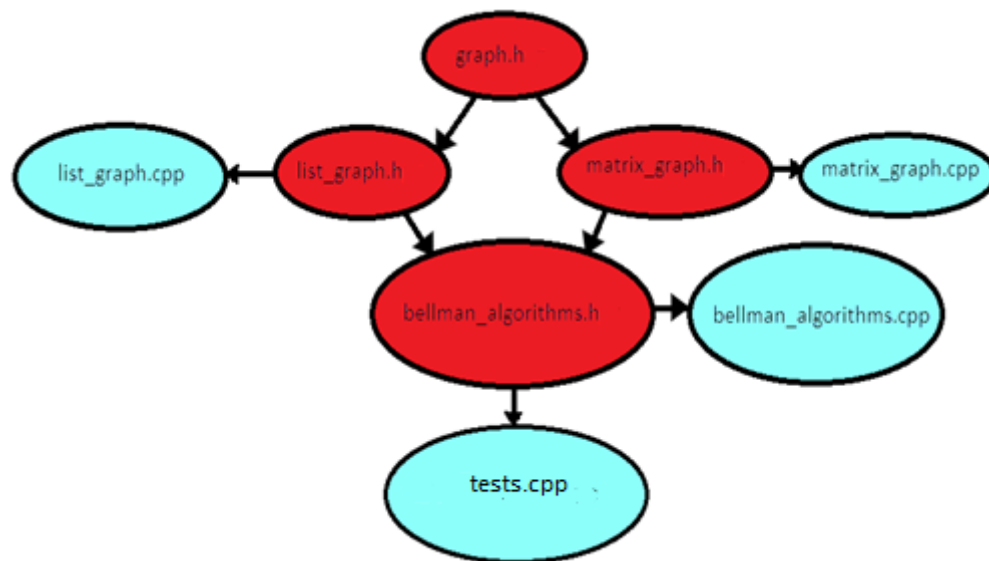
- 20
- 40
- 60
- 80
- 100

Gęstość grafu skierowanego wyraża się wzorem $\rightarrow D = \frac{|E|}{|V|(|V|-1)}$

, gdzie E to liczba krawędzi grafu, V to liczba wężłów grafu. Wzór ten będzie potrzebny aby wyliczyć ilość krawędzi grafu dla danej gęstości.

1.2 Struktura programu

Program stworzony do testowania algorytmu ma następującą strukturę plików nagłówkowych i źródłowych:



1.3 Opis algorytmu

Algorytm Bellmana-Forda w porównaniu z algorytmem Dijkstry jest wolniejszy, jednakże bardziej uniwersalny. Algorytm ten jest bowiem w stanie obsługiwać grafy z ujemnymi wartościami krawędzi oraz wykrywać ujemne cykle, które być może zostały stworzone przez te wartości.

Złożoność obliczeniowa algorytmu:

- Dla reprezentacji w postaci listy sąsiedztwa $\rightarrow \mathcal{O}(VE)$
- Dla reprezentacji w postaci macierzy sąsiedztwa $\rightarrow \mathcal{O}(V^3)$

Złożoność pamięciowa algorytmu (przy założeniu, że bierzemy pod uwagę również sam graf):

- Dla reprezentacji w postaci listy sąsiedztwa $\rightarrow \mathcal{O}(V + E)$
- Dla reprezentacji w postaci macierzy sąsiedztwa $\rightarrow \mathcal{O}(V + V^2) = \mathcal{O}(V^2)$

1.4 Przewidywane wyniki

Można zauważyć że złożoność macierzy sąsiedztwa zależy tylko i wyłącznie od liczby węzłów. Oznacza to, że wzrost gęstości grafu nie powinien wpływać na czas wykonania algorytmu; znaczenie powinna mieć tutaj tylko ilość węzłów. Jednakże dla listy sąsiedztwa ilość krawędzi ma już znaczenie więc gęstość grafu będzie wpływać na czas działania.

2. Przebieg testów

Wszystkie pomiary czasów podane w tabelach są w mili sekundach.

2.1 Tabele z pomiarami w zależności od gęstości oraz ilości wierzchołków

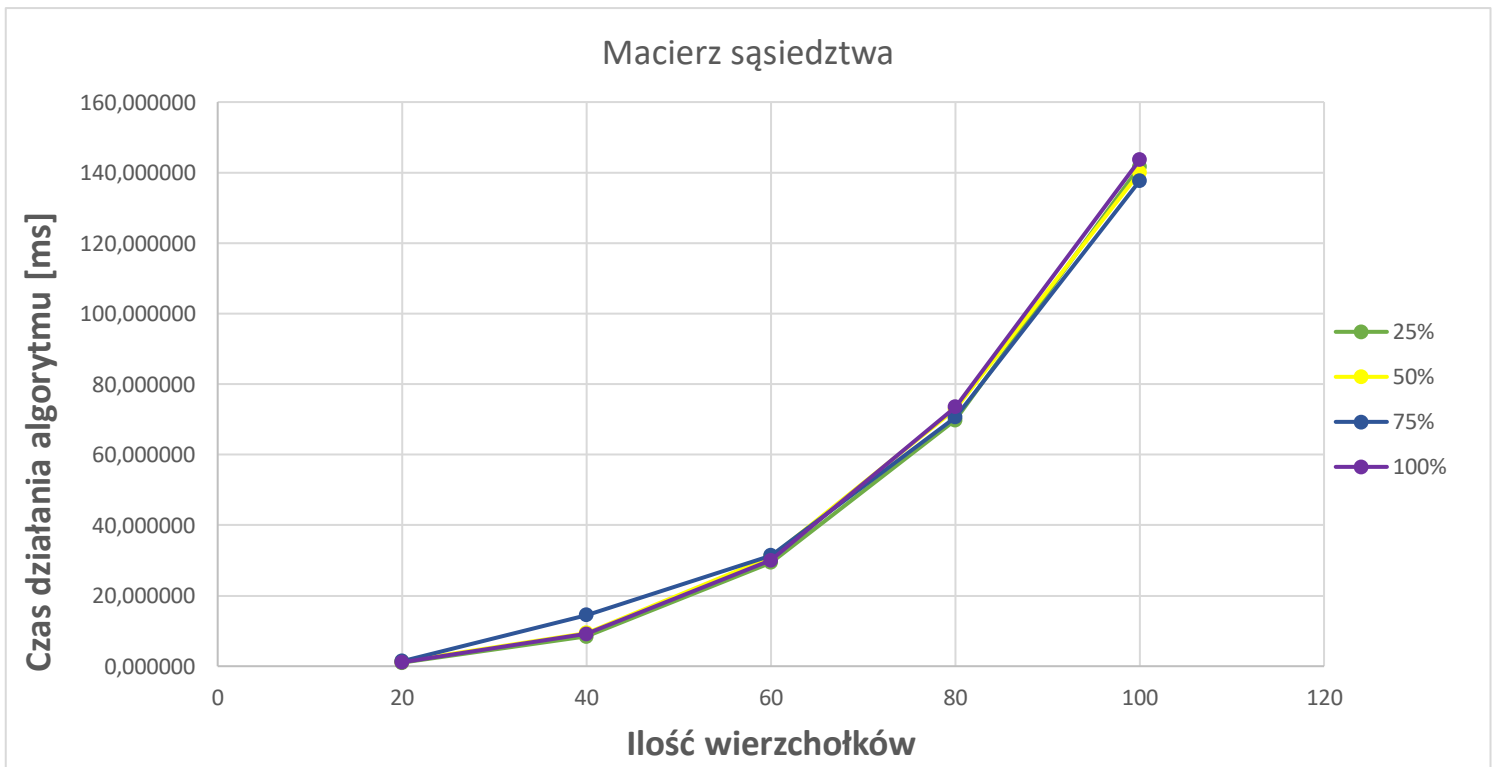
Rozmiar \ Gęstość		20	40	60	80	100
25	%	1,010630	8,415340	29,419300	69,826300	141,694000
50	%	1,340780	9,386230	31,034800	72,848300	139,992000
75	%	1,400870	14,519400	31,410700	70,646800	137,641000
100	%	1,110600	9,190650	30,049800	73,508200	143,575000

Tab.1 Pomiary czasu dla reprezentacji macierzy sąsiedztwa

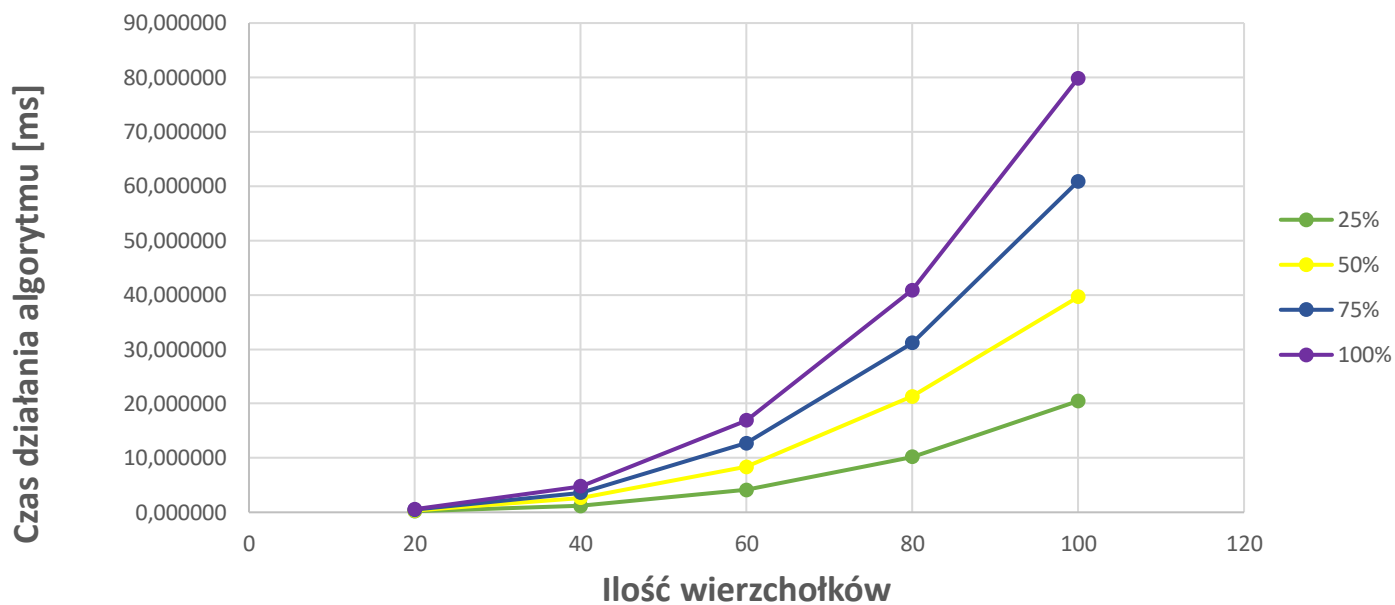
Rozmiar \ Gęstość		20	40	60	80	100
25	%	0,200136	1,170870	4,143090	10,206900	20,473700
50	%	0,330149	2,631750	8,345570	21,344300	39,606500
75	%	0,500338	3,622340	12,718600	31,180800	60,890500
100	%	0,580339	4,762950	16,890800	40,857000	79,882800

Tab.2 Pomiary czasu dla reprezentacji listy sąsiedztwa

2.2 Wykresy typu pierwszego – w zależności od reprezentacji grafu



Lista sąsiedztwa

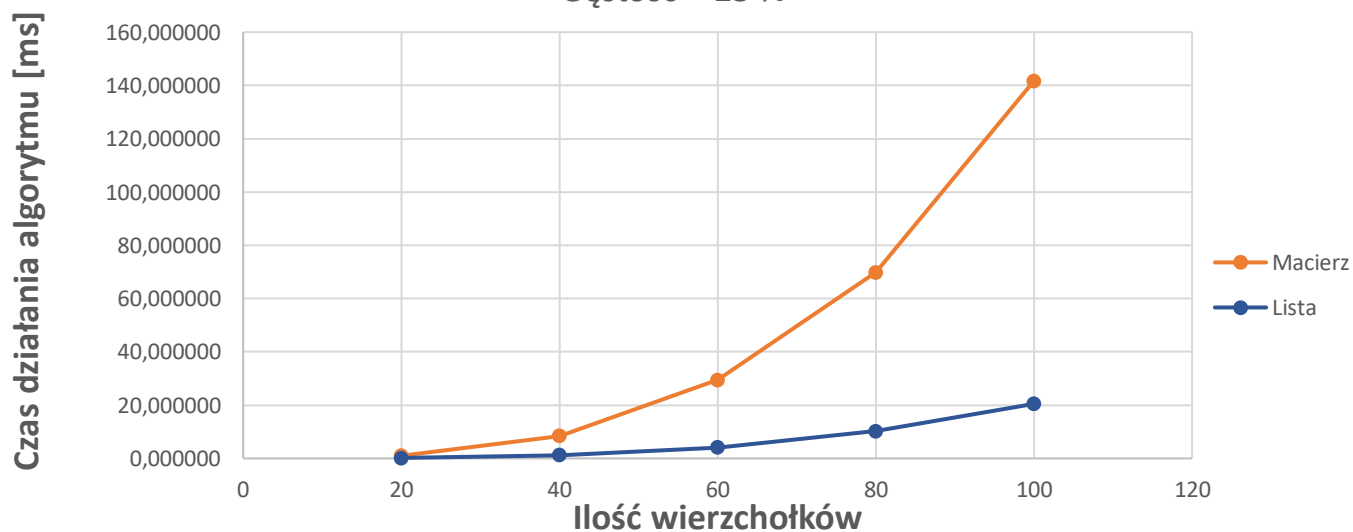


2.3 Wykresy typu drugiego – w zależności od gęstości grafu

2.3.1 Gęstość 25%

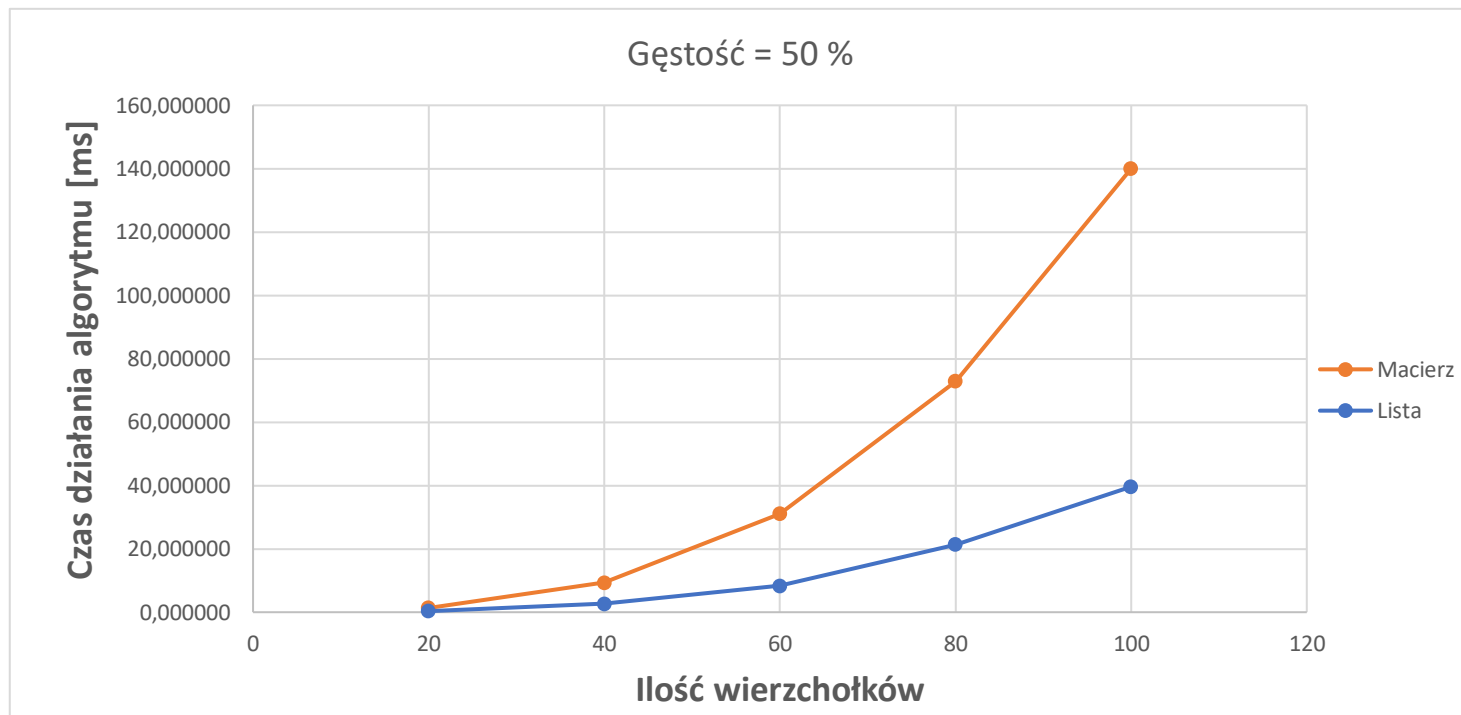
25%	20	40	60	80	100
macierz	1,010630	8,415340	29,419300	69,826300	141,694000
lista	0,200136	1,170870	4,143090	10,206900	20,473700

Gęstość = 25 %



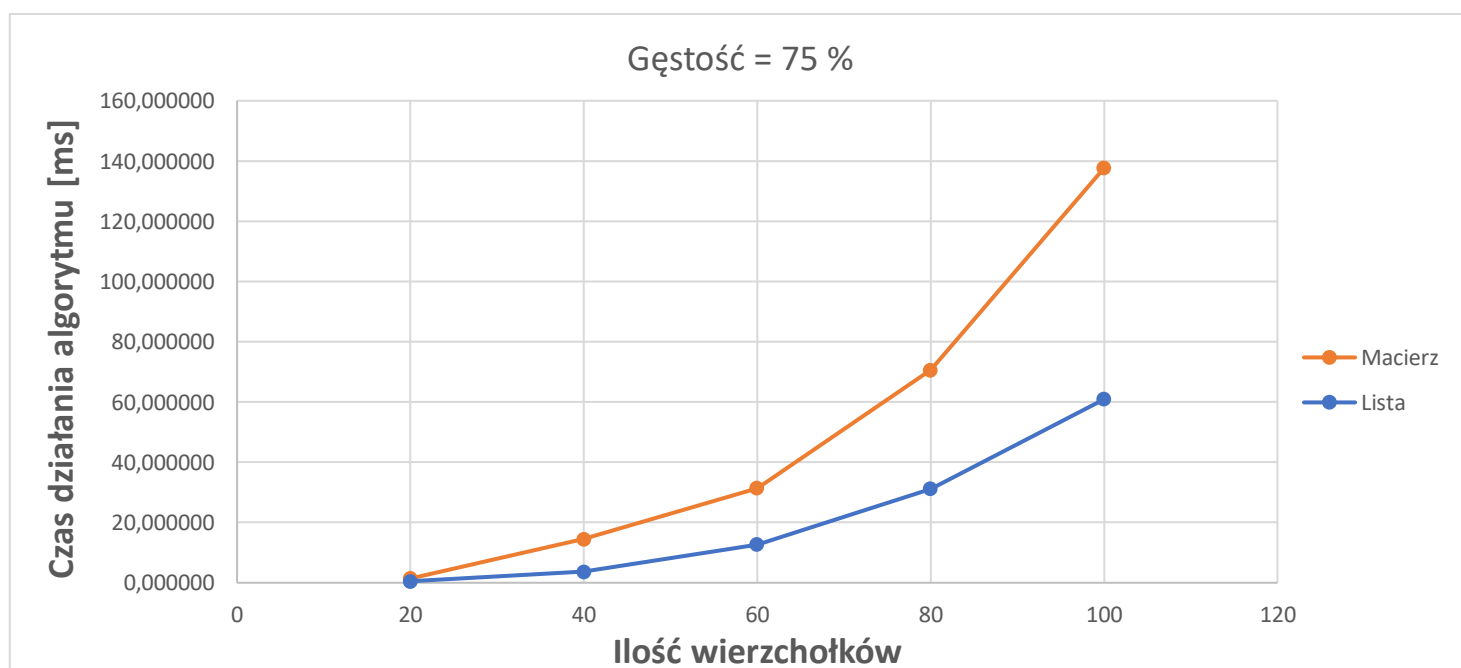
2.3.2 Gęstość 50%

50%	20	40	60	80	100
macierz	1,340780	9,386230	31,034800	72,848300	139,992000
lista	0,330149	2,631750	8,345570	21,344300	39,606500



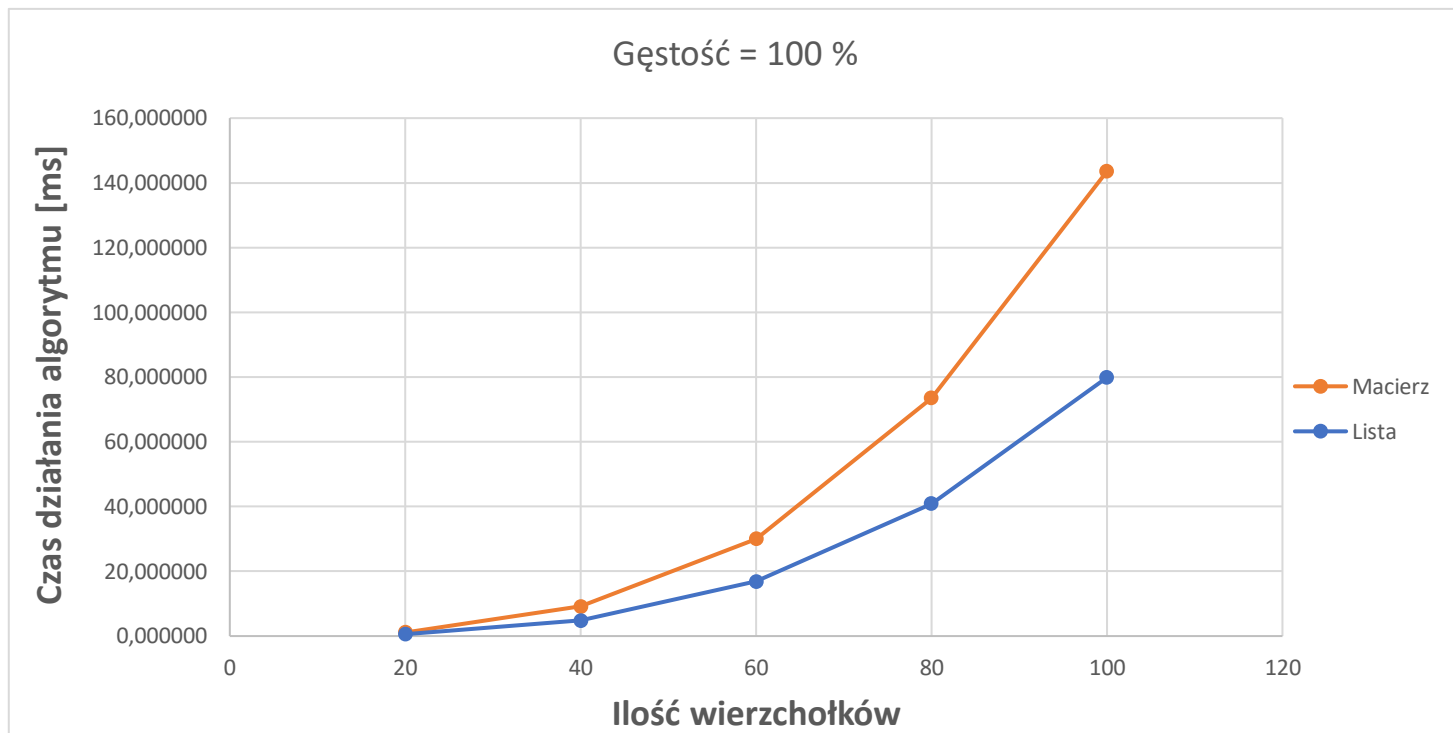
2.3.3 Gęstość 75%

75%	20	40	60	80	100
macierz	1,400870	14,519400	31,410700	70,646800	137,641000
lista	0,500338	3,622340	12,718600	31,180800	60,890500



2.3.4 Gęstość 100%

	20	40	60	80	100
macierz	1,110600	9,190650	30,049800	73,508200	143,575000
lista	0,580339	4,762950	16,890800	40,857000	79,882800



3. Podsumowanie i wnioski

1. Zgodnie z przypuszczeniami, gęstość grafu wpływa na czas wykonania algorytmu w reprezentacji listy, natomiast nie ma wpływu na reprezentacje w postaci macierzy. Można to zauważyć na wykresach 1 i 2.
2. Wraz ze wzrostem gęstości grafu w reprezentacji listy sąsiedztwa wzrasta czas wykonywania algorytmu.
3. Zgodnie z przypuszczeniami, lista jest zawsze szybsza od macierzy, co widać na wykresach 3-6.

4. Bibliografia

1. https://pl.wikipedia.org/wiki/Algorytm_Bellmana-Forda
2. https://eduinf.waw.pl/inf/alg/001_search/0138a.php
3. http://algorytmy.ency.pl/tutorial/algorytm_bellmana_forda
4. <http://lukasz.jelen.staff.iiar.pwr.edu.pl/styled-2/page-2/index.php>
5. <http://www.cs.put.poznan.pl/arybarczyk/GrafReprezentacje.htm>