Leonard Feszczuk 249027
gr. Pt 13:15-15:00

Projektowanie Algorytmów i Metody Sztucznej Inteligencji
Projekt 2- Grafy

1 Wprowadzenie

1.1 Celem projektu było zbadanie efektywności wybranego algorytmu (Dijkstry lub Bellmana-Forda) w zależności od sposobu reprezentacji grafu (w postaci macierzy i listy) oraz gęstości grafu. Badania wykonano dla 5 różnych liczb wierzchołków

- 10
- 50
- 75
- 100
- 125

Wykonano eksperymenty dla następujących przypadków gęstości:

- 25%, 50%, 75%, 100%

Dla każdego zestawu: reprezentacja grafu, liczba wierzchołków i gęstość wygenerowano po 100 losowych grafów i zapisano wyniki uśrednione(kod wypisuje czas robienia się 100 grafów lecz w excelu został on później podzielony przez 100).

2 Budowa programu

Program realizujący test efektywności napisano obiektowo tworząc dwie klasy, List i Matrix, dziedziczące po klasie graph kolejno odpowiadające za reprezentację grafu w postaci listy oraz macierzy.
Algorytm Bellmana-Forda zaimplementowano w postaci funkcji przyjmującej na wejściu wskaźnik na graf, wierzchołek startowy oraz wartość informującą czy algorytm wykonuje się w celach testowych. Jeśli tak, pomijane jest wyświetlanie rozwiązania. Funkcja zwraca strukturę Path, składającą się z dwóch tablic. Jedna
przechowuje odległości do pozostałych wierzchołków od źródła, druga służy do odczytywania ścieżki.

Poza tym, program umożliwia wczytywanie grafów z plików tekstowych. Zapis danych w pliku powinien wyglądać następująco:

- *ilość krawędzi* *ilość wierzchołków* *wierzchołek startowy*
- *wierzchołek początkowy wierzchołek końcowy* *waga*

Po znalezieniu najkrótszej ścieżki rozwiązanie jest wyświetlane oraz zapisywane do

pliku tekstowego.

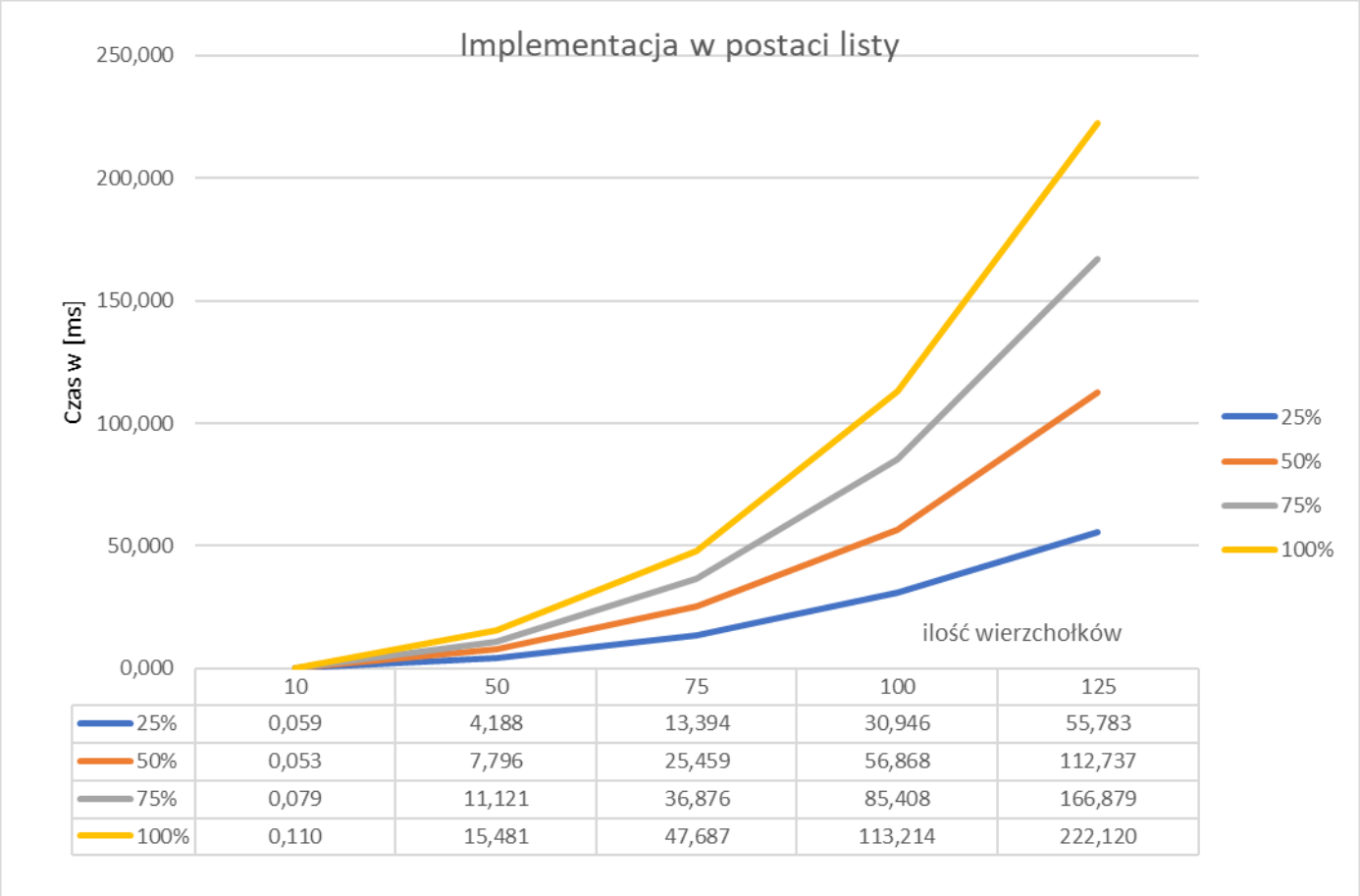
3 Badany algorytm: Bellman-Ford

Algorytm służący do znalezienia najkrótszej ścieżki w grafie ważonym z wierzchołka startowego do wszystkich pozostałych wierzchołków. Działanie algorytmu opiera się na metodzie relaksacji (sprawdzaniu, czy przy przejściu daną krawędzią grafu, nie otrzymamy krótszej ścieżki niż dotychczasowa). Algorytm Bellmana-Forda, w odróżnieniu od algorytmu Dijkstry, można stosować dla grafów z wagami ujemnymi, nie może jednak wystąpić cykl ujemny (cykl o łącznej ujemnej wadze osiągalny ze źródła).
Złożoność obliczeniowa:

O(VE)

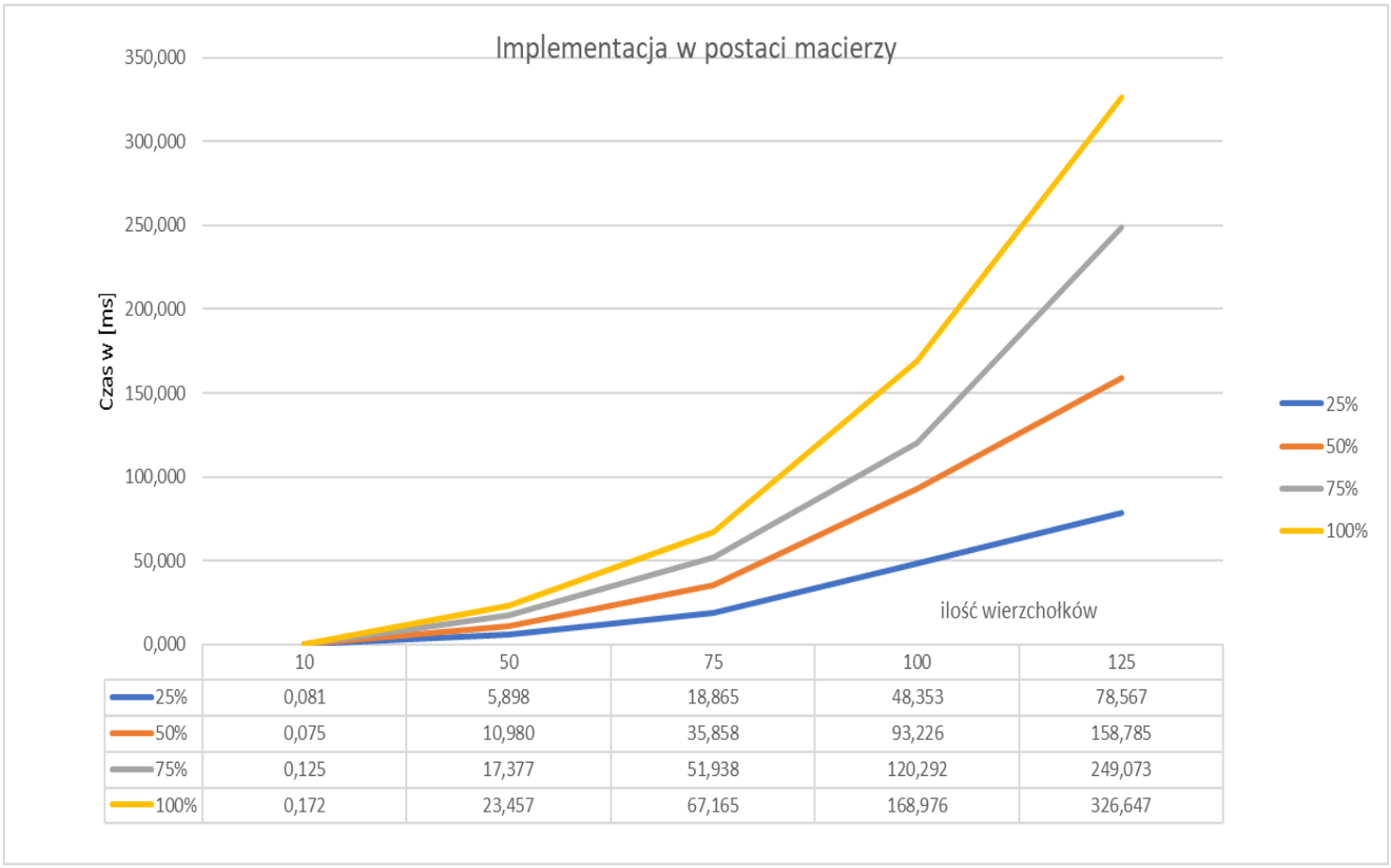
Gdzie V to ilość wierzchołków a e to ilość krawędzi.

4. Wyniki



	10	50	75	100	125
25%	0,059	4,188	13,394	30,946	55,783
50%	0,053	7,796	25,459	56,868	112,737
75%	0,079	11,121	36,876	85,408	166,879
100%	0,110	15,481	47,687	113,214	222,120

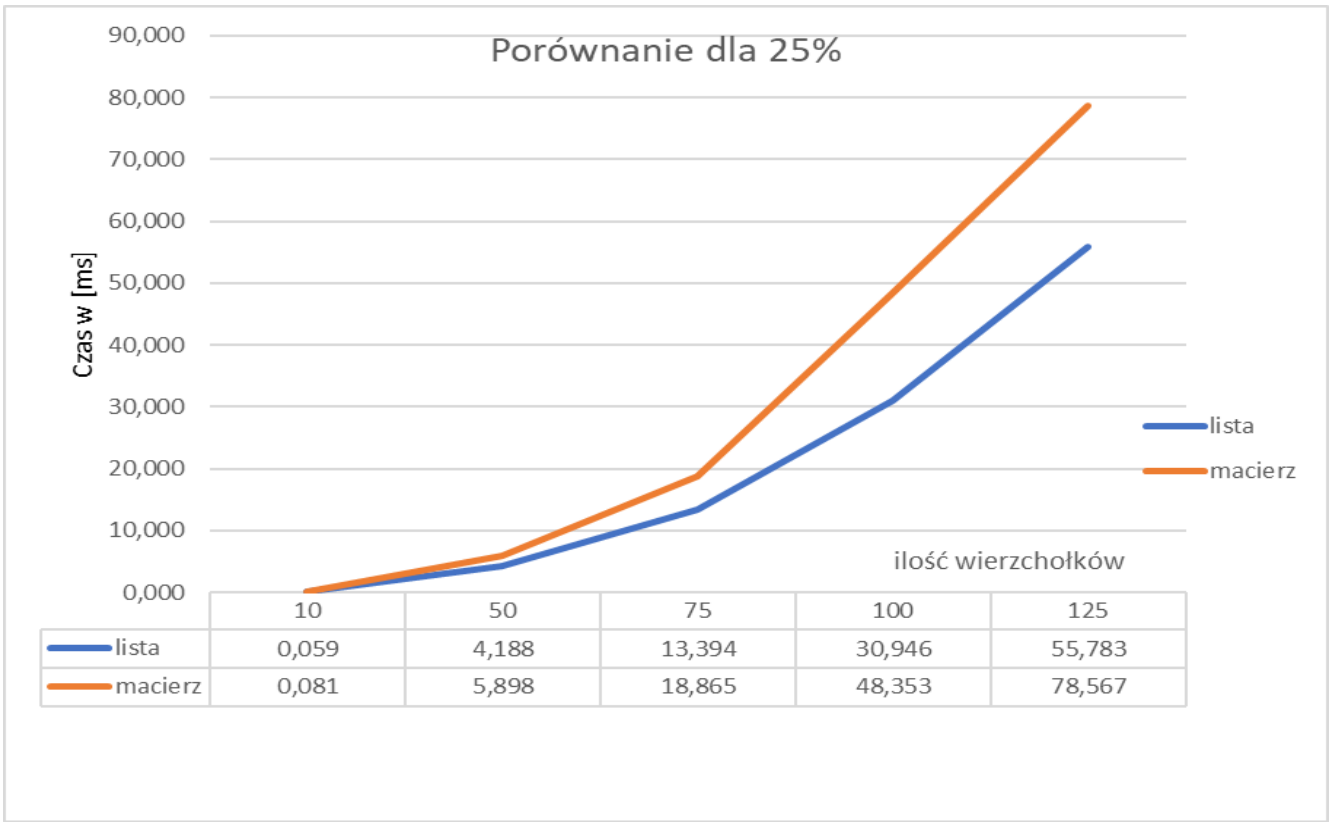
Wykresy dla różnych gęstości w implementacji listy



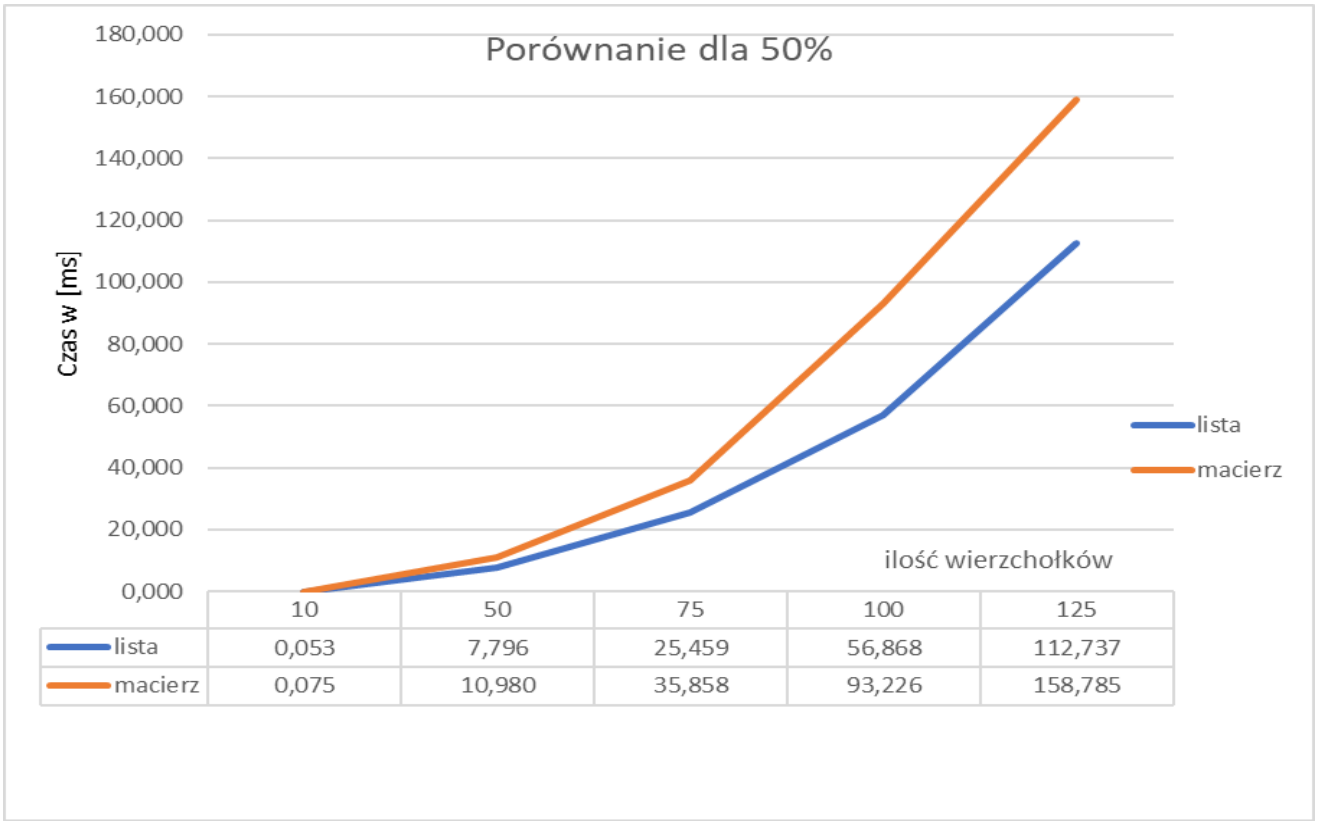
	10	50	75	100	125
25%	0,081	5,898	18,865	48,353	78,567
50%	0,075	10,980	35,858	93,226	158,785
75%	0,125	17,377	51,938	120,292	249,073
100%	0,172	23,457	67,165	168,976	326,647

Wykresy dla różnych gęstości w implementacji macierzowej

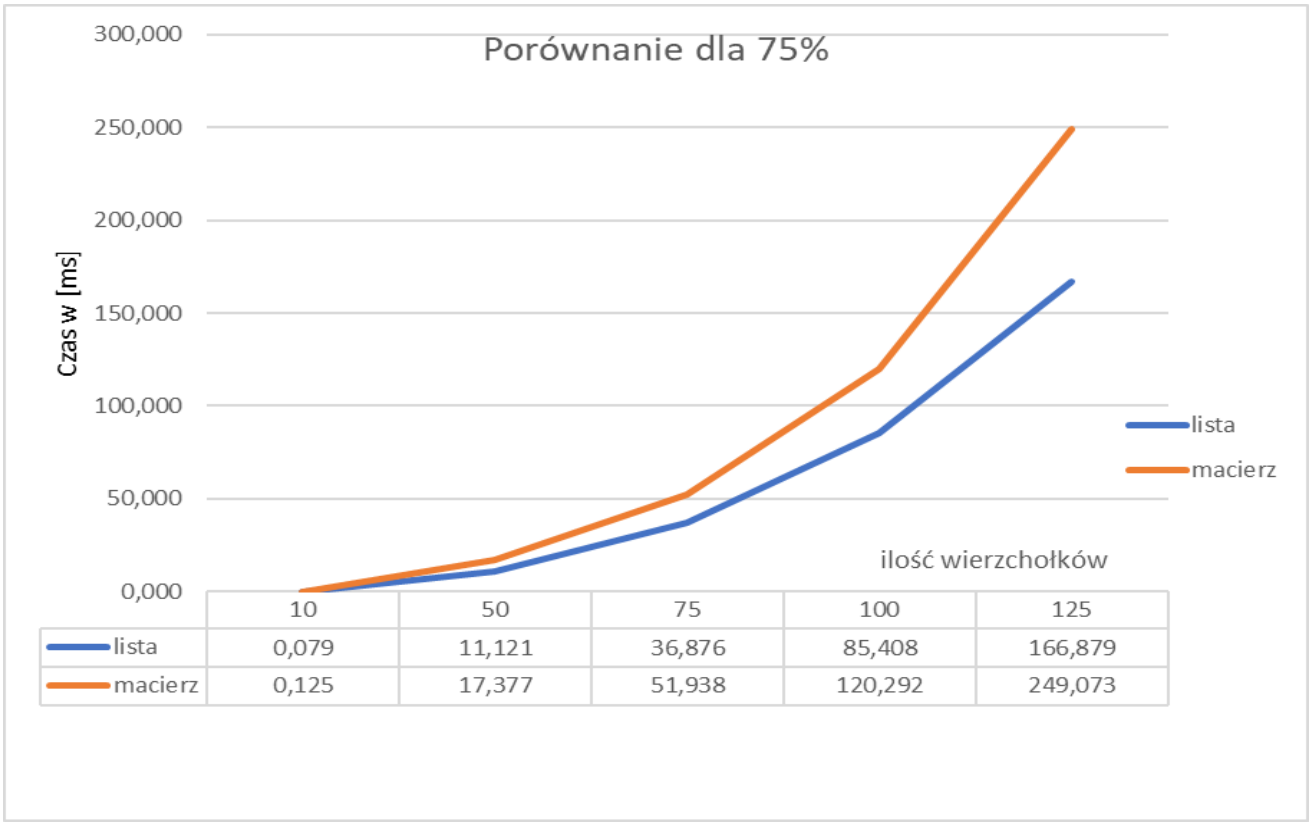
Wykresy dla poszczególnych gęstości



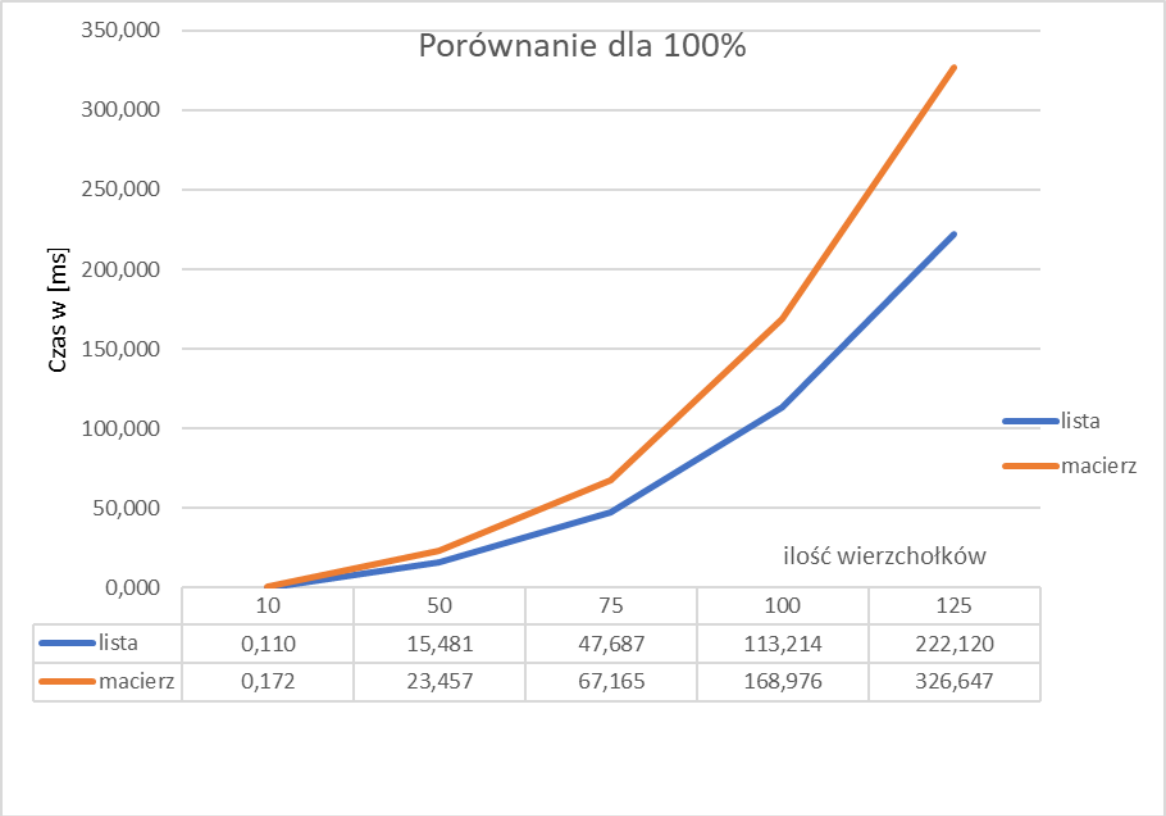
	10	50	75	100	125
lista	0,059	4,188	13,394	30,946	55,783
macierz	0,081	5,898	18,865	48,353	78,567



	10	50	75	100	125
lista	0,053	7,796	25,459	56,868	112,737
macierz	0,075	10,980	35,858	93,226	158,785



	10	50	75	100	125
lista	0,079	11,121	36,876	85,408	166,879
macierz	0,125	17,377	51,938	120,292	249,073



	10	50	75	100	125
lista	0,110	15,481	47,687	113,214	222,120
macierz	0,172	23,457	67,165	168,976	326,647

5. Wnioski

Uzyskane wyniki pokazują, że algorytm Bellmana-Forda dla implementacji grafu za pomocą listy jest bardziej efektywny niż dla implementacji za pomocą macierzy,
co można zauważyć na wykresach. Oczywiście wraz ze wzrostem gęstości rośnie tez czas rozwiązywania grafu. Przedstawione wyniki, to tylko czas działania algorytmu Bellmana-Forda, jednak biorąc pod uwagę cały proces tworzenia grafu, dla macierzy jest on zauważalnie dłuższy. Dzieje się tak , gdyż przy wywołaniu konstruktora klasy matrix wywoływana jest metoda zliczająca powtarzające się krawędzie co znacząco wydłuża czas działania programu.

Źródła:

[1] Wikipedia

Graf https://pl.wikipedia.org/wiki/Graf

[2] Wikipedia Algorytm Bellmana-Forda

https://pl.wikipedia.org/wiki/AlgorytmBellmana-Forda

[3] GeeksForGeeks Graph and its representation

https://www.geeksforgeeks.org/graph-and-its-representations/

[4] GeeksForGeeks Bellman–Ford Algorithm

https://www.geeksforgeeks.org/bellman-ford-algorithm-dp-23/