

Piątek 13:15-15:00

E08-06j

mgr Marta Emirsajłow

Projektowanie algorytmów i metody sztucznej inteligencji

Projekt 2 Grafy

1 Wstęp

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:

- 5
- 15
- 30
- 50
- 100

oraz następujących gęstości grafu:

- 25%
- 50%
- 75%
- 100% (graf pełny)

Dla każdego zestawu: reprezentacja grafu, liczba wierzchołków i gęstość wygenerowano po 100 losowych grafów i zapisano wyniki uśrednione.

2 Algorytm Bellmana-Forda

Algorytm służący do znalezienia najkrótszej ścieżki w grafie ważony z wierzchołka źródłowego 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:

$$O(|V| * |E|),$$

gdzie V to ilość wierzchołków, a E to ilość krawędzi.

Zapis w pseudokodzie:

Bellman-Ford(G, w, s):

dla każdego wierzchołka v w $V[G]$ wykonaj

$[v] = \text{nieskończone}$

 poprzednik $[v] = \text{niezdefiniowane}$

$d[s] = 0$

dla i od 1 do $|V[G]| - 1$ wykonaj

 dla każdej krawędzi (u, v) w $E[G]$ wykonaj

 jeżeli $d[v] > d[u] + w(u, v)$ to

$[v] = d[u] + w(u, v)$

 poprzednik $[v] = u$

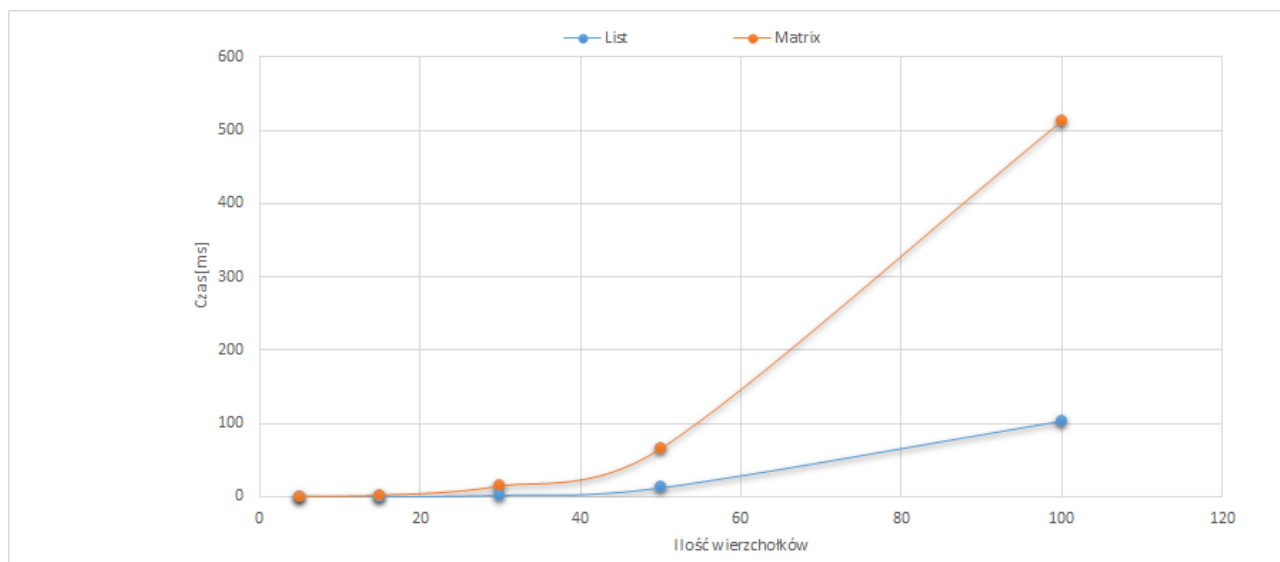
3 Opis programu

Program realizujący test efektywności został napisany obiektowo tworząc dwie klasy, ListGraph i MatrixGraph, kolejno odpowiadające za reprezentację grafu w postaci listy oraz macierzy.

Algorytm Bellmana-Forda został zaimplementowany w postaci funkcji przyjmującej na wejściu wskaźnik na graf, wierzchołek startowy oraz wartość logiczną informującą czy wynik ma zostać wypisany. Funkcja zwraca strukturę Path, składającą się z dwóch tablic. Jedna przechowuje odległości do pozostałych wierzchołków od źródła, druga służy do odczytywania ścieżki.

Ponadto, program umożliwia wczytywanie grafów z plików tekstowych. Format danych oczekiwanych przez program jest zgodny z tymi podanymi w opisie zadania. Po znalezieniu najkrótszej ścieżki rozwiązanie jest wyświetlane oraz zapisywane do pliku tekstowego.

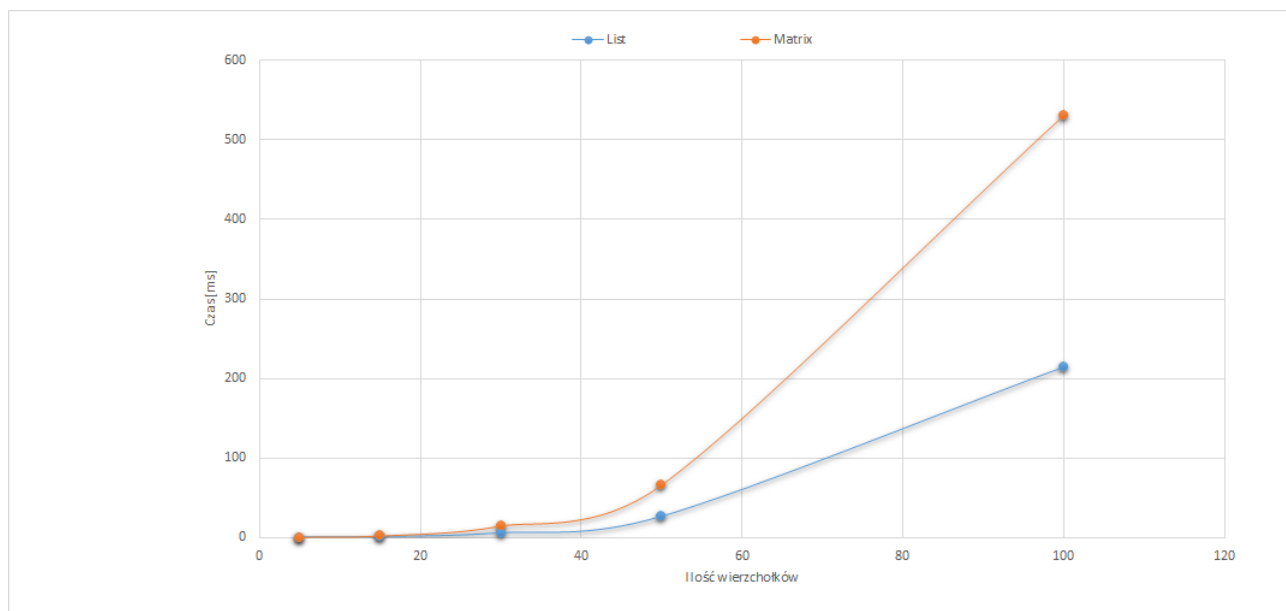
4 Wyniki pomiarów



Rysunek 1: Wykres dla gęstości grafu 25%

	Ilość krawędzi				
	5	15	30	50	100
Czas[ms] (lista)	0,012742	0,347415	2,67556	12,72529	103,0098
Czas[ms](macierz)	0,058981	1,80294	14,42459	65,68619	533,0648
Stosunek(lista/macierz)	0,212828	0,1894836	0,182276	0,190519	0,190031

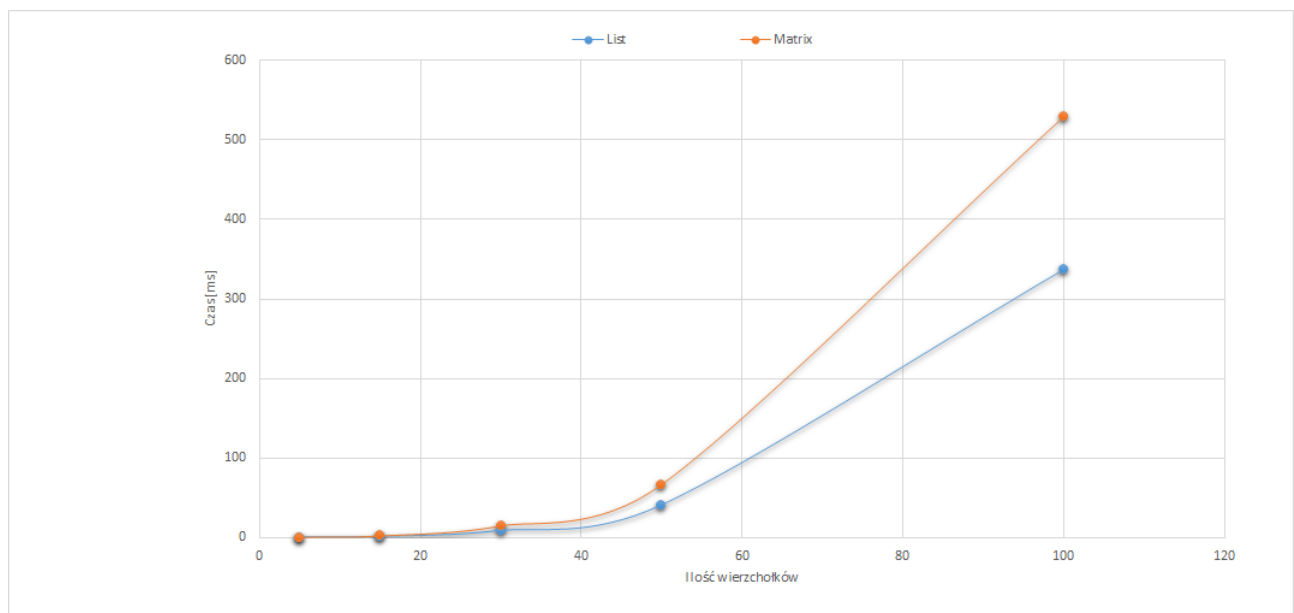
Rysunek 2: Tabela czasów wykonywania dla grafu 25%



Rysunek 3: Wykres dla gęstości grafu 50%

	Ilość krawędzi				
	5	15	30	50	100
Czas[ms] (lista)	0,017111	0,645648	5,41401	26,20949	215,0468
Czas[ms](macierz)	0,059408	1,72431	14,13859	65,74269	530,9988
Stosunek(lista/macierz)	0,284811	0,371228471	0,379714	0,395458	0,401775

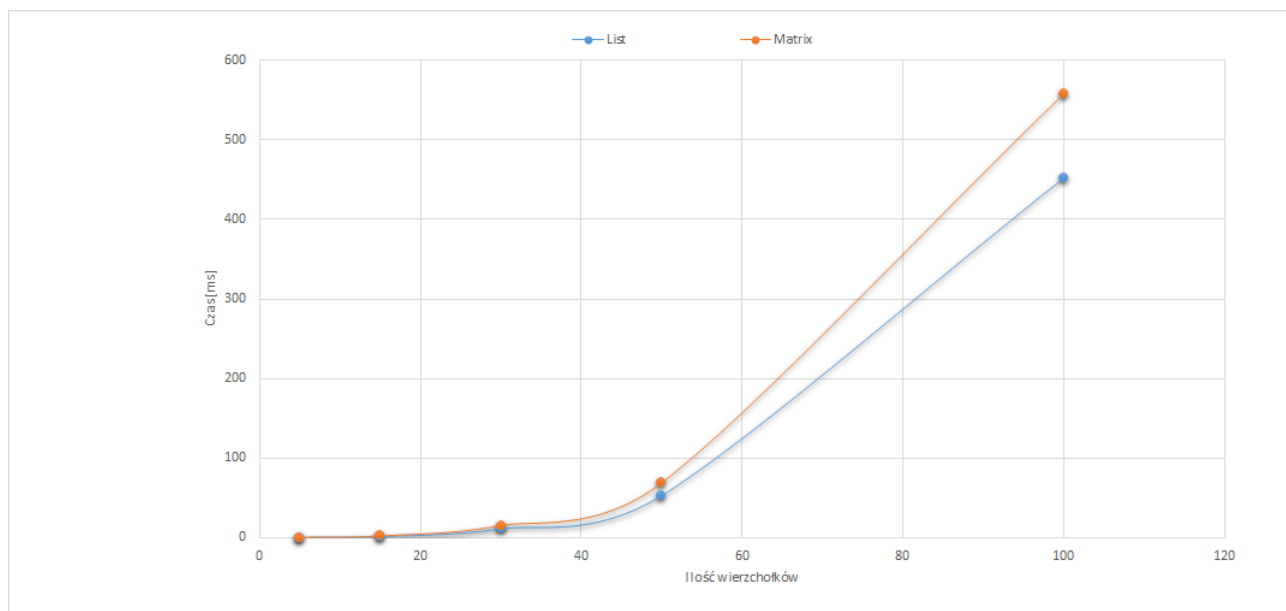
Rysunek 4: Tabela czasów wykonywania dla grafu 50%



Rysunek 5: Wykres dla gęstości grafu 75%

	Ilość krawędzi				
	5	15	30	50	100
Czas[ms] (lista)	0,025549	1,00017	8,37085	40,51429	337,1528
Czas[ms](macierz)	0,059478	1,7089	14,35289	66,34649	528,8418
Stosunek(lista/macierz)	0,426336	0,582061227	0,580007	0,607437	0,634321

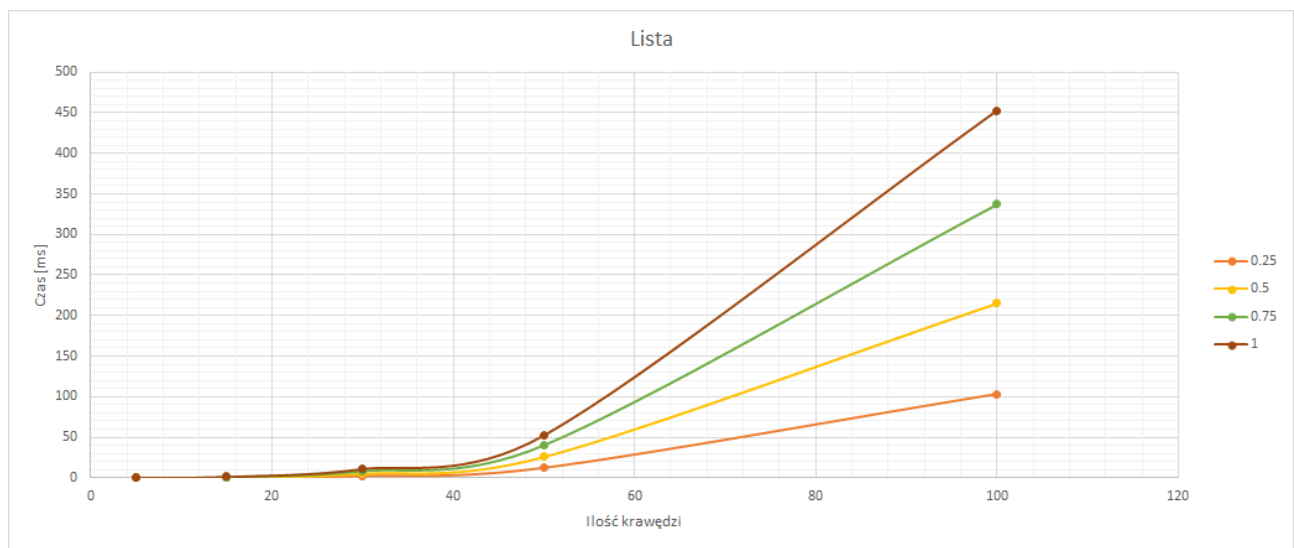
Rysunek 6: Tabela czasów wykonywania dla grafu 75%



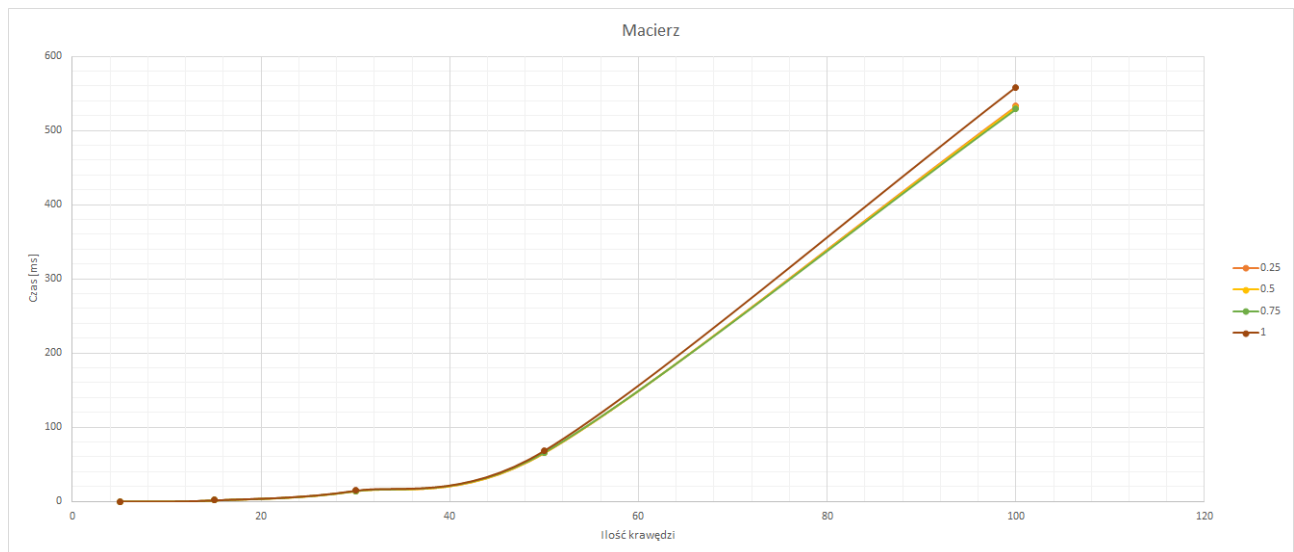
Rysunek 7: Wykres dla grafu pełnego

	Ilość krawędzi				
	5	15	30	50	100
Czas[ms] (lista)	0,047347	1,2802	11,14959	52,84779	452,4588
Czas[ms](macierz)	0,063216	1,81135	14,88169	68,54899	558,0498
Stosunek(lista/macierz)	0,745756	0,703555672	0,746005	0,767739	0,807576

Rysunek 8: Tabela czasów wykonywania dla grafu pełnego



Rysunek 9: Wykres dla listy sąsiedztwa



Rysunek 10: Wykres dla macierzy sąsiedztwa

5 Podsumowanie i wnioski

- Zgodnie z przypuszczeniami, gęstość grafu wpływa na czas egzekucji algorytmu w reprezentacji listy, natomiast nie ma wpływu na reprezentacje w postaci macierzy.
- Zgodnie z założeniem, wraz ze wzrostem gęstości (ilości krawędzi) grafu, czas egzekucji dla reprezentacji w postaci listy zwiększa się.
- Uzyskane wyniki pokazują, że algorytm dla reprezentacji grafu za pomocą listy sąsiedztwa jest bardziej efektywny niż dla reprezentacji za pomocą macierzy sąsiedztwa, co wyraźnie można zauważyć na powyższych wykresach. Może być to spowodowane przez to, że przeszukiwanie całej tablicy jest wolniejsze niż dostęp do danych w za- implementowanej liście jednokierunkowej.

6 Literatura

Podczas wykonywania projektu wspierałem się następującymi stronami:

1. Wikipedia Algorytm Bellmana-Forda
2. GeeksForGeeks Bellman–Ford Algorithm
3. GeeksForGeeks Graph and its representation