

Projektowanie Algorytmów i Metody Sztucznej Inteligencji

Projekt 3 - Grafy

Szymon Zajda 248917

Prowadzący: mgr.inż Marcin Ochman

WT 15¹⁵

05.05.2020

1 Wprowadzenie

Celem projektu było zbadanie efektywności wybranego algorytmu - w tym wypadku Bellmana-Forda - w zależności od sposobu reprezentacji grafu (w postaci macierzy i listy) oraz gęstości grafu. Badania przeprowadzono dla następującej ilości wierzchołków: 10, 50, 100, 500, 1000 oraz dla następujących gęstości: 0.25, 0.50, 0.75, 1.

2 Budowa programu

Program opiera się na 2 klasach: *MatrixGraph*, *ListGrap*. Obie kolejno przechowują metody znajdowania najkrótszej ścieżki w grafie. Dodatkowo została utworzona klasa dziedziczna *Graph*, która odpowiedzialna jest za przechowywanie liczby wierzchołków oraz wagi. Inicjalizowanie i tworzenie obiektu grafu odbywa się za pomocą inteligentnych wskaźników *unique_ptr*, które dbają o to by po zakończeniu działania programu skasowały wskazywany obiekt. Tworząc reprezentację grafu w programie głównym *main.cpp* jednocześnie zadaje się ilość wierzchołków oraz gęstość wypełnienia.

Funkcja znajdowania najkrótszej ścieżki w grafie to ***double BellmanFord***. Zwraca czas wykonywania się algorytmu. To pozwala na sumowanie wykonywania się algorytmu w pętli dla 100 instancji.

3 Opis algorytmu

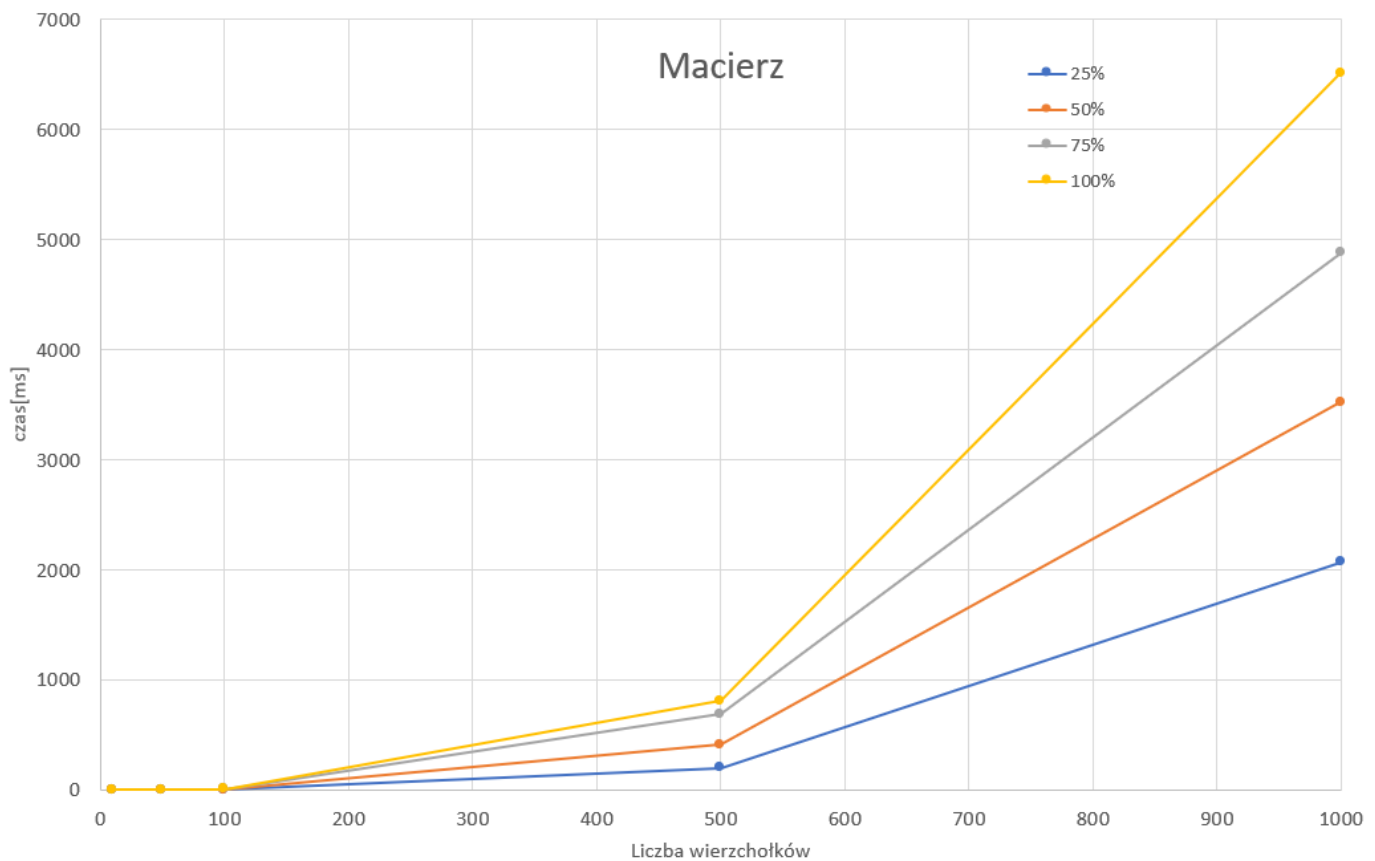
Algorytm służący do znalezienia najkrótszej ścieżki w grafie ważony z wierzchołka źródłowego do wszystkich pozostałych wierzchołków. Działanie algorytmu opiera się na metodzie relaksacji czyli sprawdzaniu, czy przy przejściu daną krawędzią grafu, nie otrzymamy krótszej ścieżki niż dotychczasowa). Algorytm 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: $O(V \cdot E)$, gdzie V to ilość wierzchołków, a E to ilość krawędzi.

4 Sporządzone wykresy i wyniki testów

4.1 Reprezentacja macierzy

Tabela zawiera uśrednione czasy[w ms]

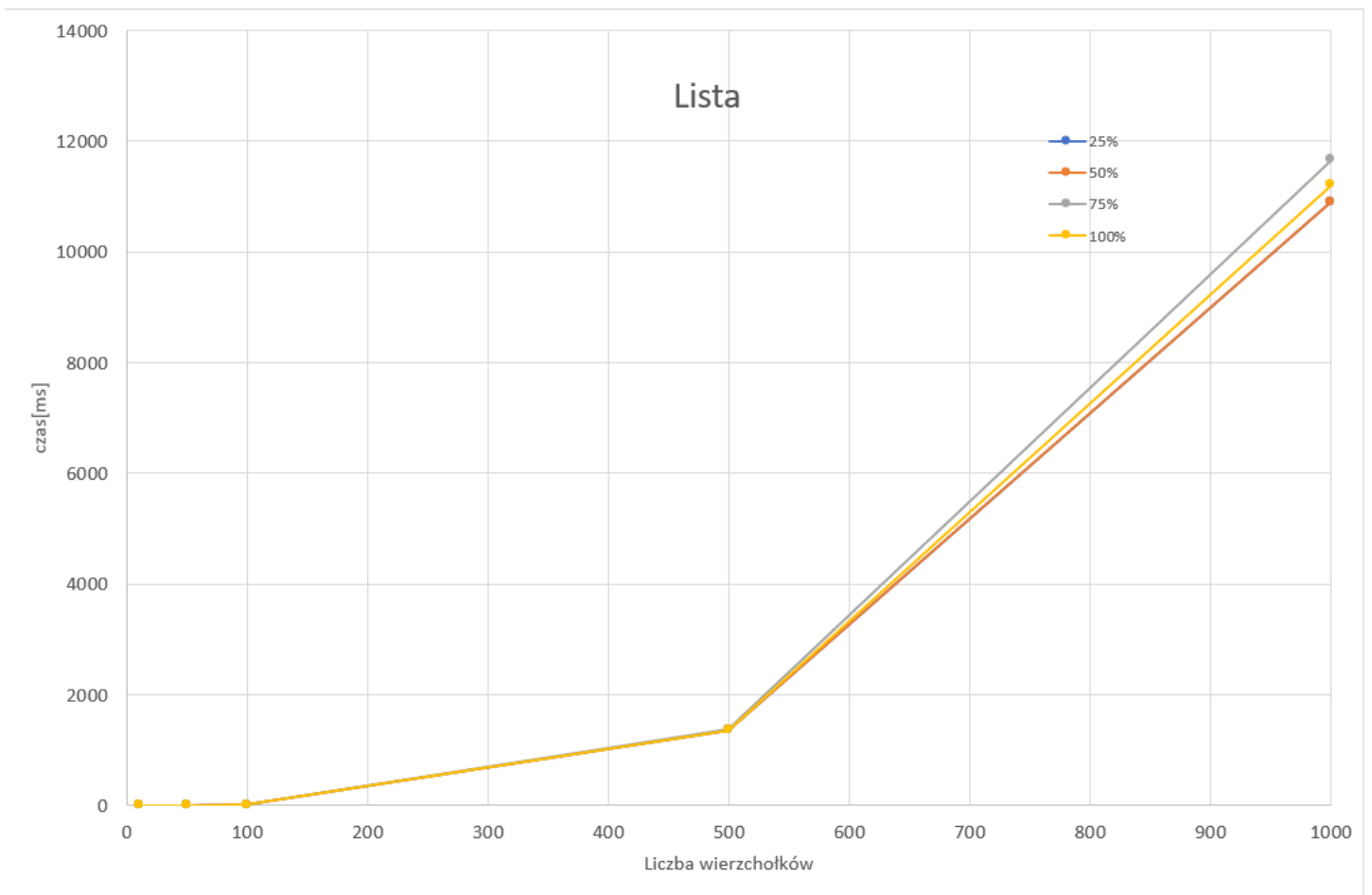
MACIERZ		Liczba wierzchołków				
		10	50	100	500	1000
Procent wypełnieni a grafu	25%	0,01997	0,300102	2,29466	201,745	2065,62
	50%	0,031961	0,469956	3,21785	409,791	3520,35
	75%	0,043512	0,645283	4,75751	685,634	4878,39
	100%	0,057141	0,902676	7,2709	808,881	6509,84



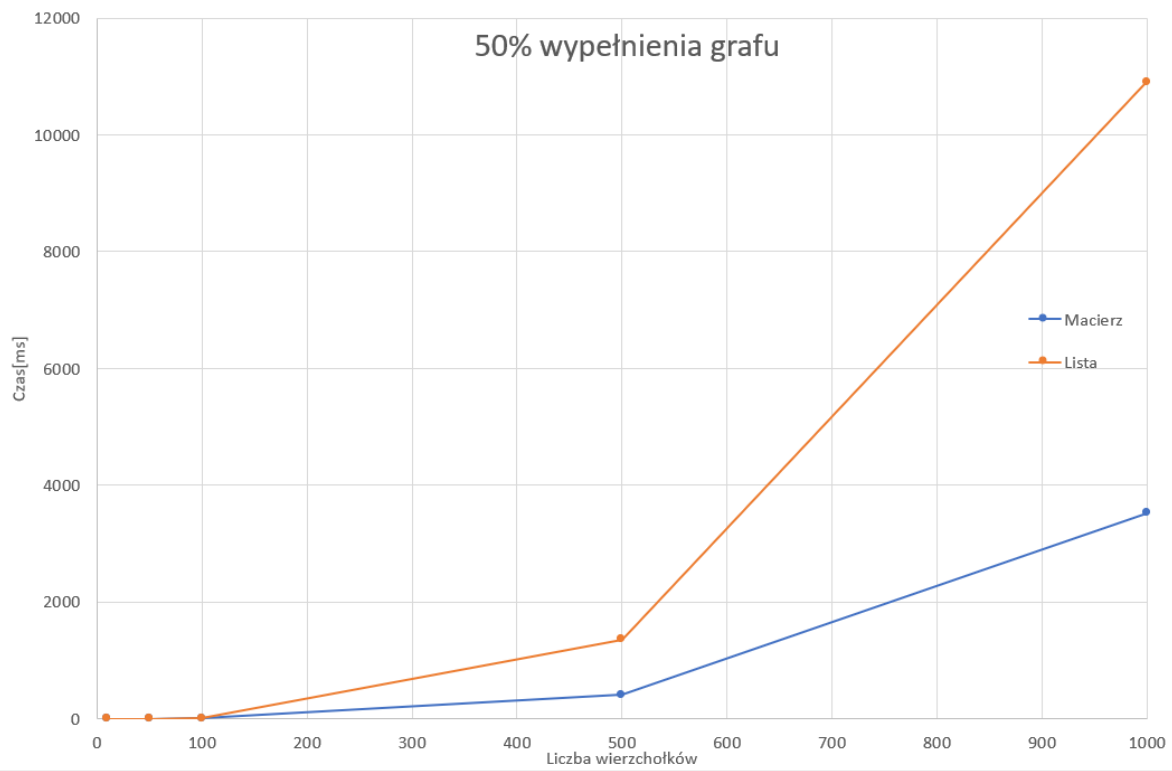
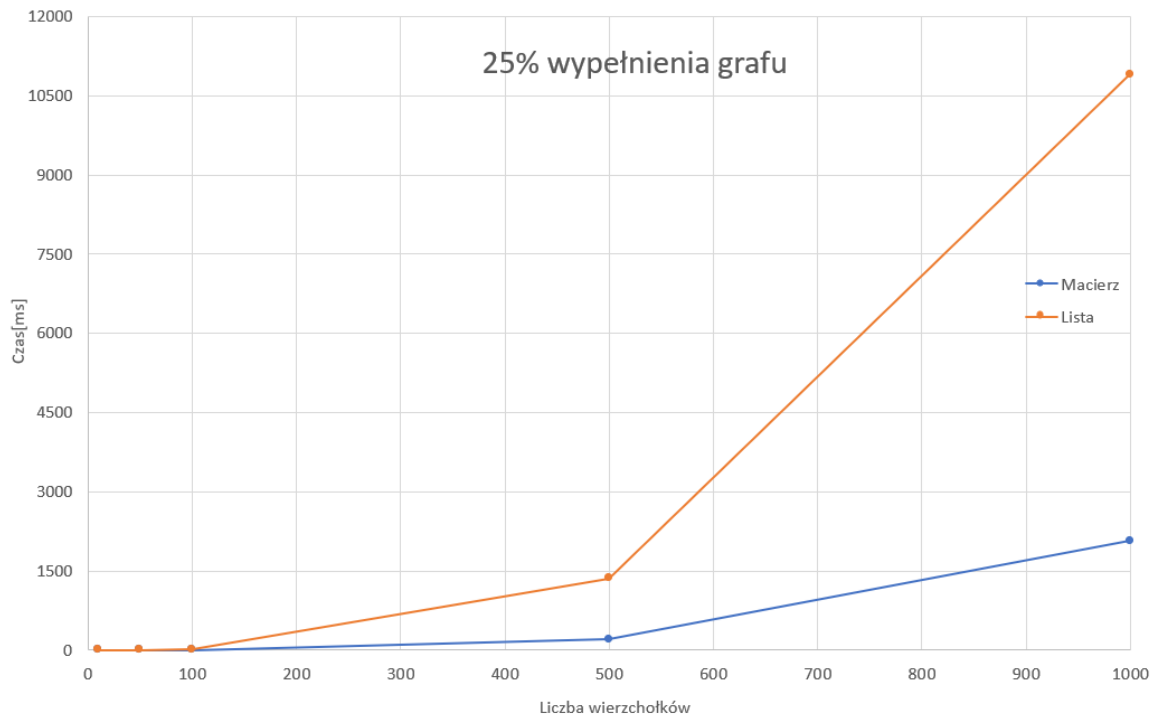
4.2 Reprezentacja listy

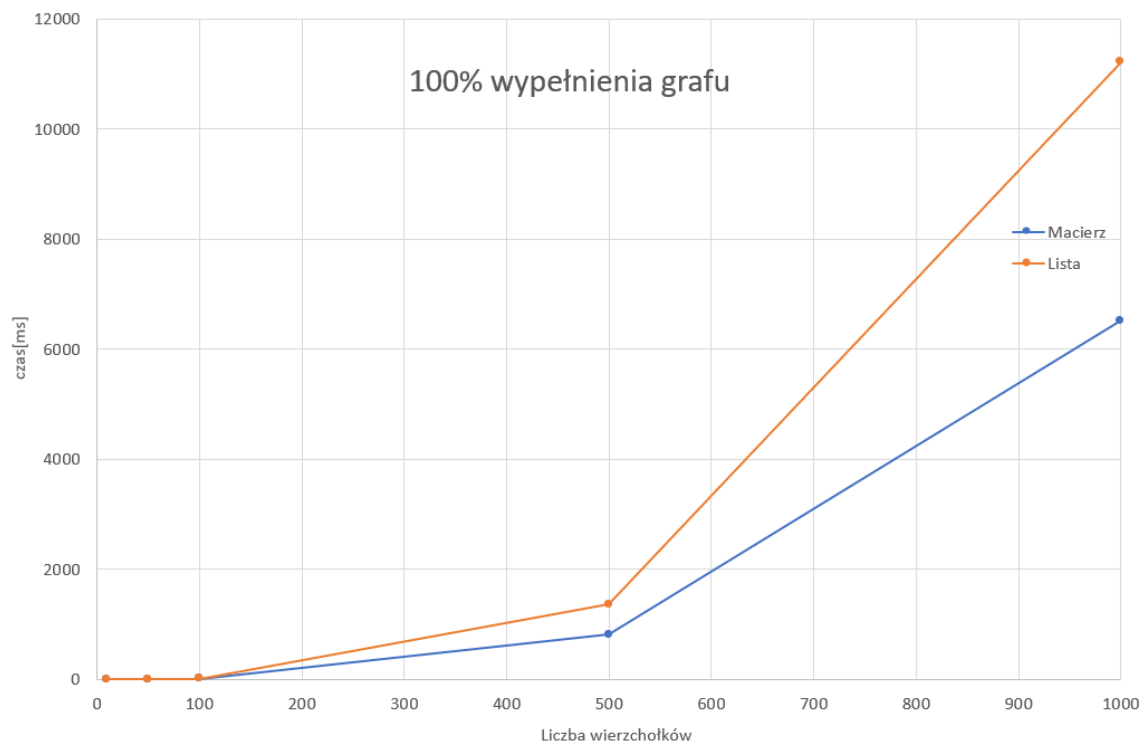
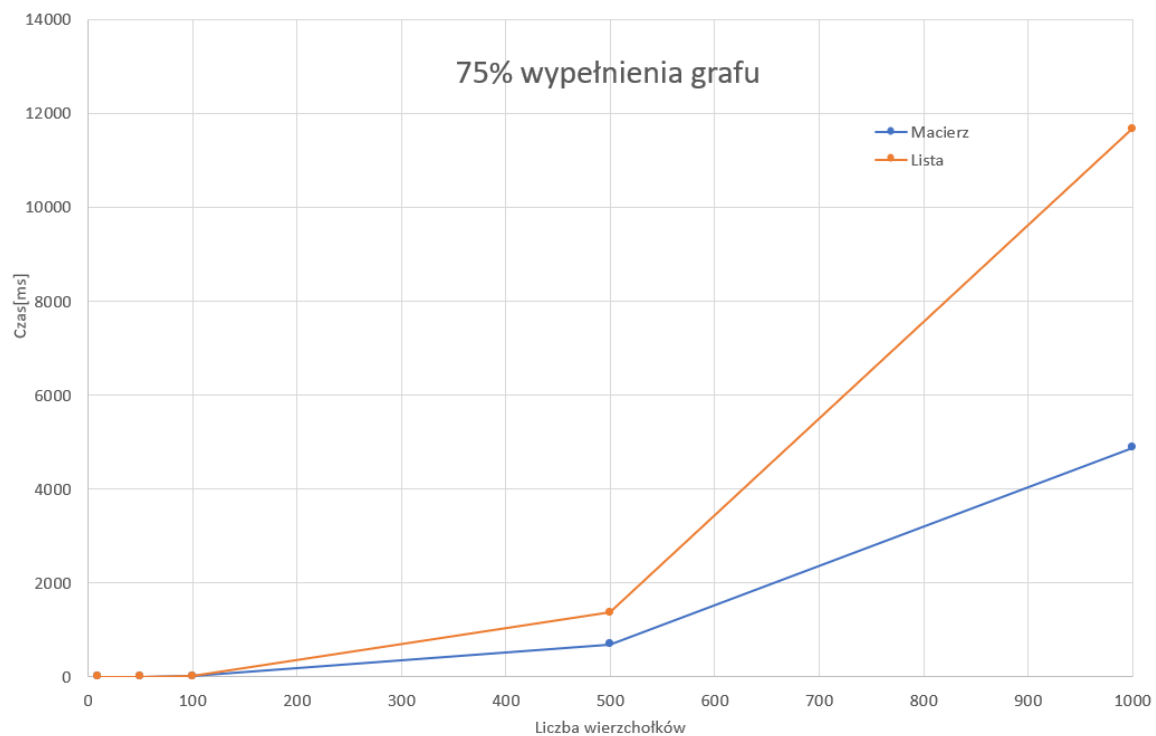
Tabela zawiera uśrednione czasy[w ms]

LISTA		Liczba wierzchołków				
		10	50	100	500	1000
Procent wypełnieni a grafu	25%	0,170631	2,21027	13,3041	1363,81	10902,2
	50%	0,113128	1,5657	11,592	1360,72	10904,1
	75%	0,107148	1,56136	13,0694	1365,31	11661,4
	100%	0,107142	1,47514	11,6773	1363,21	11211,9



Wykresy porównawcze dla obu reprezentacji w tej samej gęstości





5 Wnioski

- Algorytm BellmanaFord'a okazuje się być bardziej efektywny dla reprezentacji macierzy sąsiedztwa niż dla reprezentacji listy.
- Dla macierzy sąsiedztwa widać wyraźne zróżnicowanie wyników pomiędzy poszczególnymi gęstościami

grafów aniżeli dla listy

- Zestawiając rozbieżność wyników w reprezentacji macierzy a jednolite wyniki w reprezentacji listy można wytłumaczyć tym, że w macierzy znajdowanie najkrótszej ścieżki odbywa się przez iterowanie, natomiast lista ma bezpośredni dostęp do swoich komórek. Toteż gęstość grafu, czyli ilość elementów nie ma wpływu dla reprezentacji listy.

6 Literatura

- https://eduinf.waw.pl/inf/alg/001_search/0138a.php
- https://pl.wikipedia.org/wiki/Algorytm_Bellmana-Forda
- [https://pl.wikipedia.org/wiki/Graf_\(matematyka\)](https://pl.wikipedia.org/wiki/Graf_(matematyka))
- <https://www.geeksforgeeks.org/bellman-ford-algorithm-dp-23/>