

Projektowanie Algorytmów i Metody Sztucznej Inteligencji

Sprawozdanie z projektu 2

Temat: analiza grafów algorytmem Bellmana-Forda przy dwóch metodach implementacji grafu

10.05.2020

Bartosz Wójcik

249010

Kod załączony do sprawozdania: <https://github.com/bartech99/PAMSI2>

Prowadzący: mgr inż. Marta Emirsajłow

Grupa: piątek 7:30-9:00

1. Wstęp

Celem zadania było porównanie efektywności algorytmu Bellmana-Forda dla dwóch różnych implementacji grafów – metodą macierzy sąsiedztwa oraz list sąsiedztwa. Po wybraniu pozycji nr 2 w menu, program generuje losowo po 100 instancji grafów dla każdej gęstości (25%, 50%, 75%, 100%) i dla każdej ilości wierzchołków (30, 50, 100, 200, 300). Rezultaty pomiaru czasu zapisuje w plikach tekstowych, zawierających w nazwie daną ilość wierzchołków grafu. Czasy są uśrednione.

Macierz jest dwuwymiarową tabelą, której pierwszy indeks oznacza wierzchołek początkowy, a drugi końcowy w danym połączeniu. Jako wartości przyjmuje wagi tych połączeń.

Celem implementacji list sąsiedztwa, stworzono tablicę o ilości pól równej ilości wierzchołków w grafie. Każdy jej indeks oznacza wierzchołek początkowy, a każda komórka zawiera wskaźnik na pierwszy element listy, której kolejne składniki zawierają wierzchołek końcowy, wagę danego połączenia oraz wskaźnik na kolejny element.

2. Opis użytego algorytmu

W projekcie wybrano algorytm Bellmana-Forda. Jest on używany do znalezienia najkrótszej drogi między grafem startowym, a każdym innym. Jego działanie opiera się na zasadzie relaksacji. Polega ona na „przechodzeniu” przez każde połączenie i porównywaniu kosztów danych dróg. Złożoność obliczeniowa tego algorytmu wynosi $O(V \cdot E)$. Wyniki zapisuje w tablicy o ilości pól równej ilości wierzchołków i zapisuje ją do pliku tekstowego w formacie:

ilość-wierzchołków ilość-krawędzi wierzchołek-początkowy
wierzchołek-końcowy koszt-najkrótszej-drogi
...

3. Wyniki

3.1. Tabele

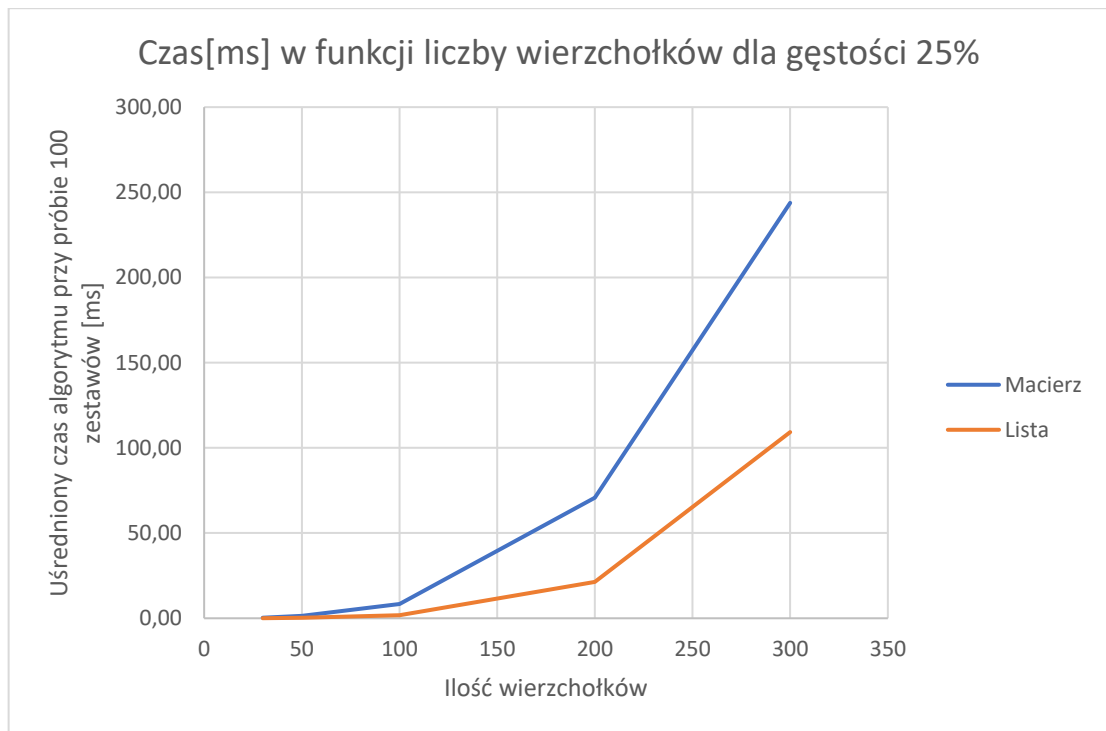
Liczba wierzchołków:	Gęstość:			
	25%	50%	75%	100%
30	0,34	0,38	0,40	0,25
50	1,35	1,28	1,46	1,03
100	8,46	7,97	9,48	8,05
200	70,70	71,44	69,06	73,40
300	243,81	249,74	249,27	240,89

Tabela 1: Wyniki pomiarów czasu w [ms] przy implementacji metodą macierzy sąsiedztwa.

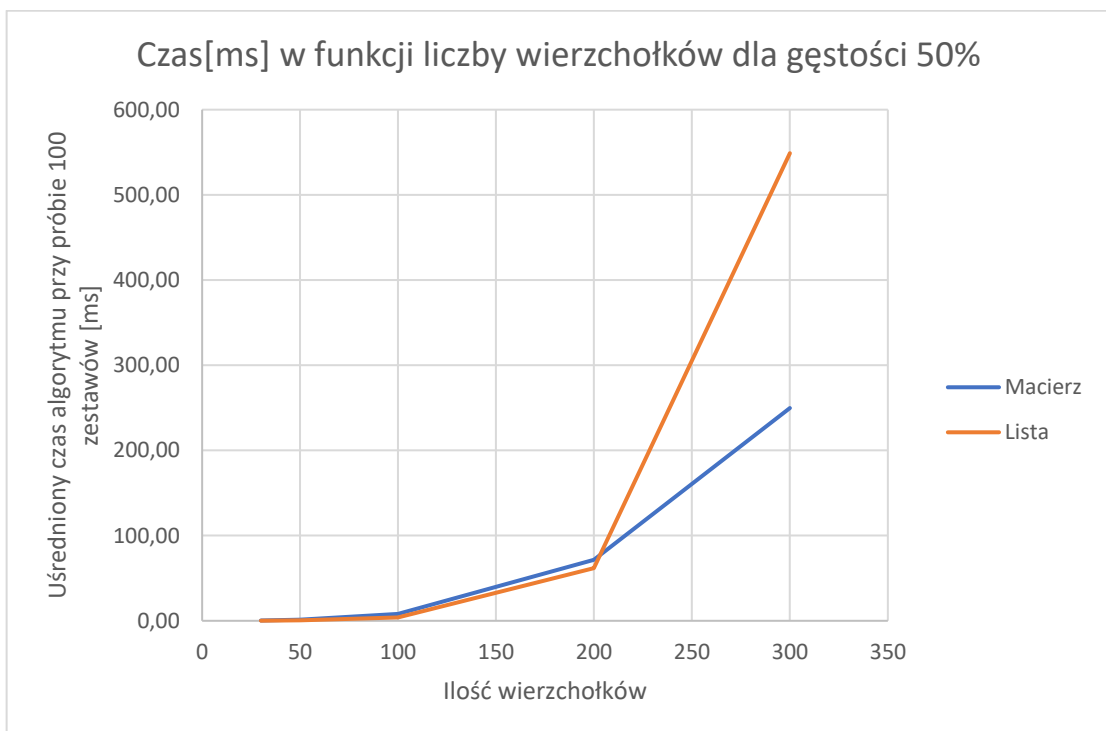
Liczba wierzchołków:	Gęstość:			
	25%	50%	75%	100%
30	0,05	0,15	0,09	0,14
50	0,31	0,56	0,95	0,84
100	1,85	3,94	7,30	10,38
200	21,39	61,96	110,76	219,50
300	109,26	548,87	2040,86	3014,13

Tabela 2: Wyniki pomiarów czasu w [ms] przy implementacji metodą list sąsiedztwa.

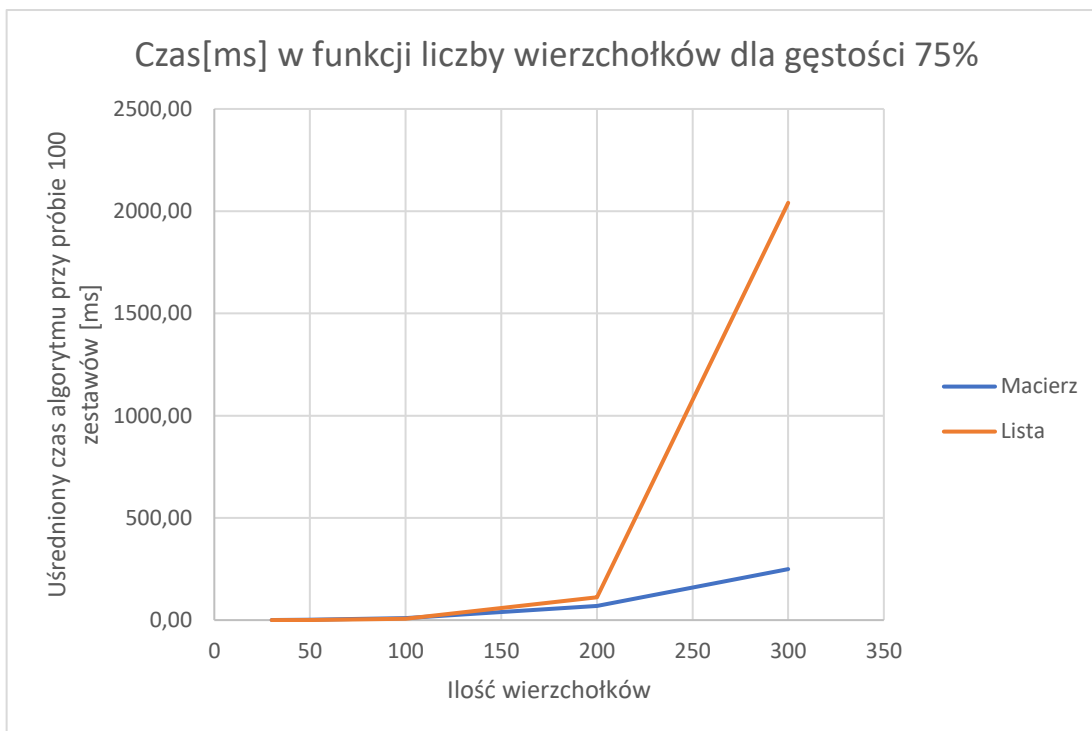
3.2. Wykresy typu 1



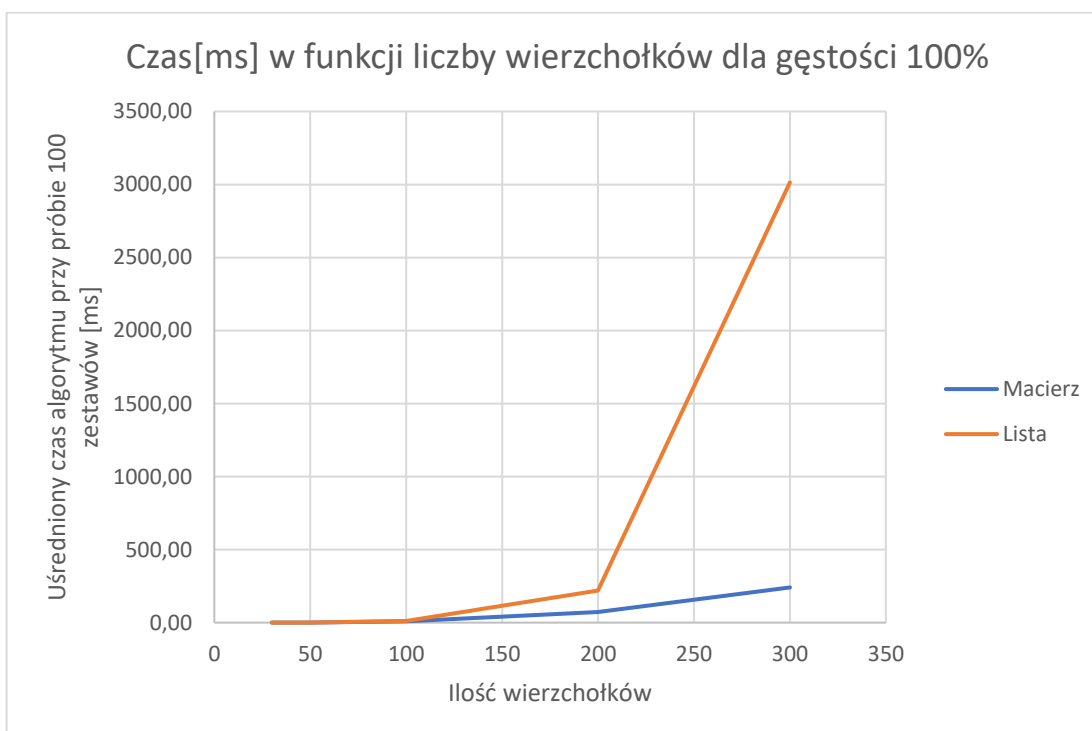
Wykres 1: Porównanie czasów działania algorytmu dla obu metod, przy gęstości grafu 25%.



Wykres 2: Porównanie czasów działania algorytmu dla obu metod, przy gęstości grafu 50%.

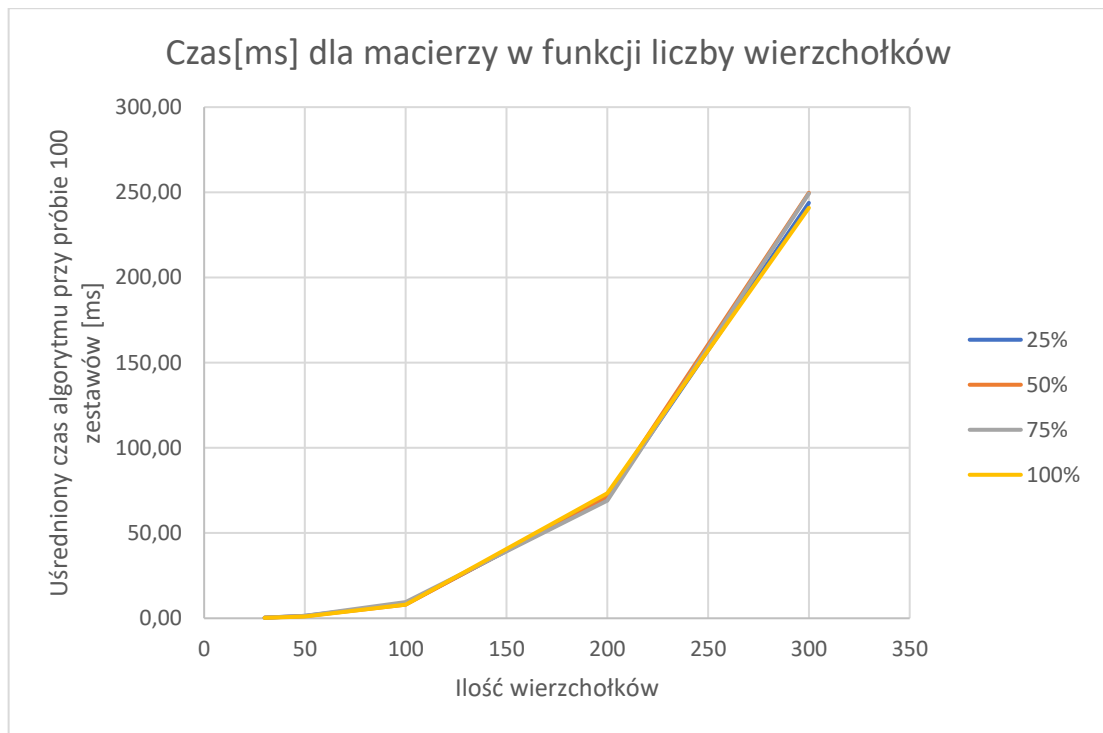


Wykres 3: Porównanie czasów działania algorytmu dla obu metod, przy gęstości grafu 75%.

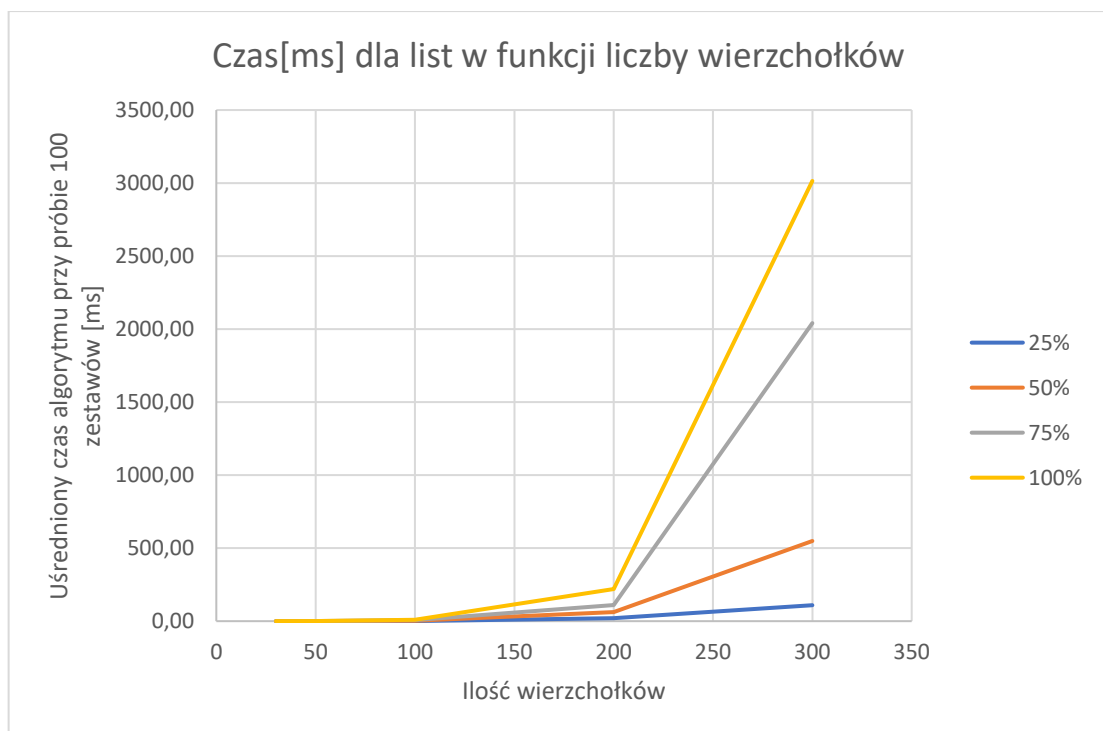


Wykres 4: Porównanie czasów działania algorytmu dla obu metod, przy gęstości grafu 100%.

3.3. Wykresy typu 2



Wykres 5: Porównanie czasów działania algorytmu dla metody macierzy sąsiedztwa, w funkcji gęstości grafu.



Wykres 6: Porównanie czasów działania algorytmu dla metody list sąsiedztwa, w funkcji gęstości grafu.

4. Wnioski

Rosnąca gęstość grafu nie skutkuje wydłużeniem czasu wykonywania algorytmu przy implementacji grafu metodą macierzy sąsiedztwa. Jest to spowodowane faktem, że wymiary macierzy nie ulegają zmianie przy zwiększeniu ilości krawędzi. Przeciwnie jest w przypadku list – im większa gęstość badanego grafu, tym czas obliczeń się wydłuża. Skutkiem tego jest wprost proporcjonalnie rosnąca ilość pozycji w liście do ilości połączeń.

Jak widać na wykresach typu 1, za nagły wzrost czasu wykonywania algorytmu przy implementacji grafów metodą list sąsiedztwa, odpowiada również rosnąca liczba wierzchołków. W przypadku metody macierzy sąsiedztwa, zwiększająca się ilość wierzchołków nie powoduje tak znacznego wydłużenia operacji.

5. Bibliografia

- https://eduinf.waw.pl/inf/alg/001_search/0124.php
- <https://www.geeksforgeeks.org/bellman-ford-algorithm-dp-23/>
- <https://www.tutorialspoint.com/bellman-ford-algorithm-in-cplusplus>
- <https://www.geeksforgeeks.org/graph-and-its-representations/>