

Projektowanie algorytmów i metody sztucznej inteligencji

Grafy

Mateusz Tkacz

248961

1. Wprowadzenie

Zadanie polegało na zaimplementowaniu grafów – zbioru wierzchołków oraz łączących ich krawędzi - za pomocą listy sąsiedztwa oraz macierzy sąsiedztwa. Następnie zaimplementowany został algorytm Dijkstry do wyznaczania najkrótszej ścieżki w grafie z danego wierzchołka. Jednym z prostszych zastosowań grafów jest wyznaczanie drogi, np. między przystankami w mieście lub między miastami (np. między lotniskami). Grafy mogą być skierowane bądź nieskierowane. W skierowanym połączenie między wierzchołkami jest jednostronne, a w nieskierowanym dwustronne. Grafy posiadają swoją gęstość, która jest równa ilorazowi krawędzi grafu, przez wszystkie możliwe krawędzie. W testach algorytm był wykorzystywany dla 4 różnych gęstości grafu: 25%, 50%, 75% i 100% oraz różnych liczb wierzchołków: 10, 50, 100, 250, 500. Badane było 100 grafów, a wyniki zostały uśredniane dla jednego grafu.

2. Opis algorytmu

W projekcie wykorzystany został algorytm Dijkstry. Polega on na tym, że wybierany zostaje wierzchołek startowy. Następnie każdemu wierzchołkowi ustawia się etykietę odległości od tego wierzchołka, równą nieskończoność, poza startowym, dla którego jest ona równa 0. Następnie przechodzi się po krawędziach incydentnych bieżącego wierzchołka i aktualizuje etykietę odległości w innym wierzchołku. Jest ona równa sumie etykiety bieżącego wierzchołka i wagi krawędzi między wierzchołkami. Aktualizacja następuje tylko gdy wartość bieżąca będzie mniejsza niż aktualna etykieta wierzchołka. Do przechowywania wierzchołków sąsiednich do bieżącego najlepiej jest wykorzystać kolejkę priorytetową, dzięki której złożoność obliczeniowa jest równa $O(E \log V)$. W przypadku użycia innej metody z liniową złożonością złożoność wzrasta do $O(V^2)$.

3. Wyniki

1. Czasy trwania algorytmu Dijkstry z wykorzystaniem listy sąsiedztwa (czas w s)

Czasy trwania algorytmu Dijkstry dla listy sąsiedztwa				
Liczba wierzchołków	Gęstość grafu			
	25%	50%	75%	100%
10	0,000009	0,000010	0,000012	0,000013
50	0,000085	0,000129	0,000171	0,000204
100	0,000270	0,000432	0,000611	0,000798
250	0,001416	0,002639	0,003950	0,005464
500	0,005679	0,012287	0,019522	0,025219

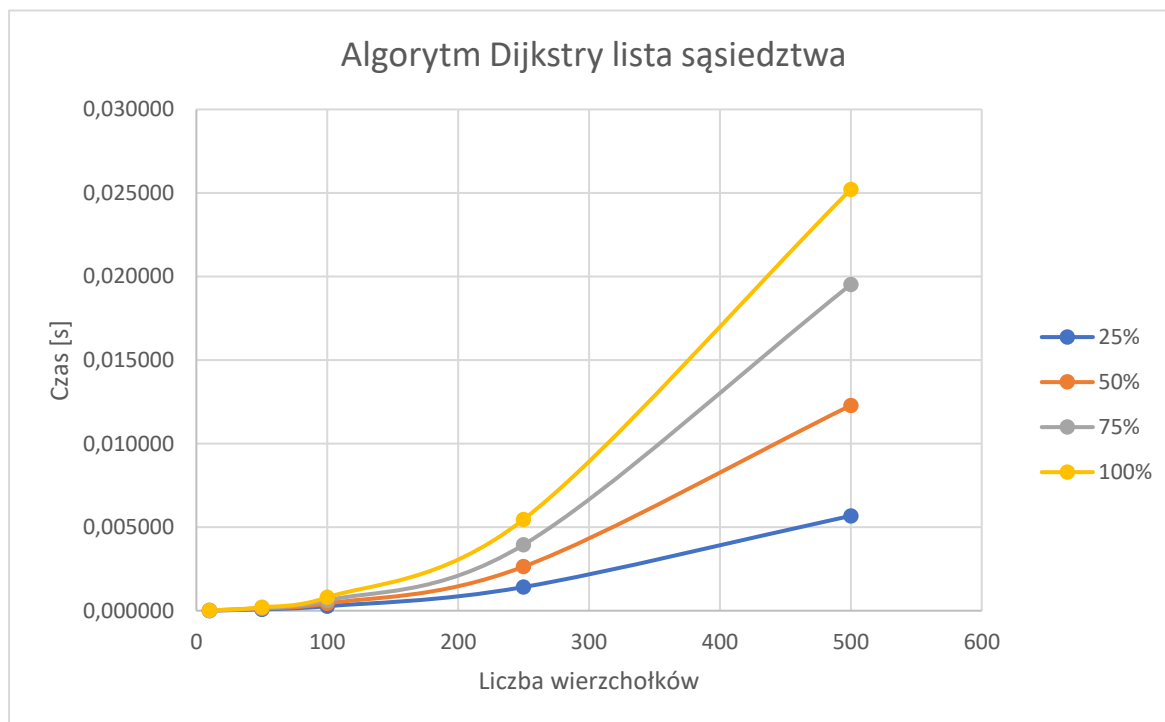
2. Czasy trwanie algorytmu Dijkstry z wykorzystaniem macierzy sąsiedztwa (czas w s)

Czasy trwania algorytmu Dijkstry dla macierzy sąsiedztwa				
Liczba wierzchołków	Gęstość grafu			
	25%	50%	75%	100%
10	0,000035	0,000039	0,000042	0,000044
50	0,000258	0,000358	0,000449	0,000540
100	0,000778	0,001196	0,001604	0,002017
250	0,003721	0,006505	0,009345	0,011759
500	0,013534	0,024852	0,038544	0,049809

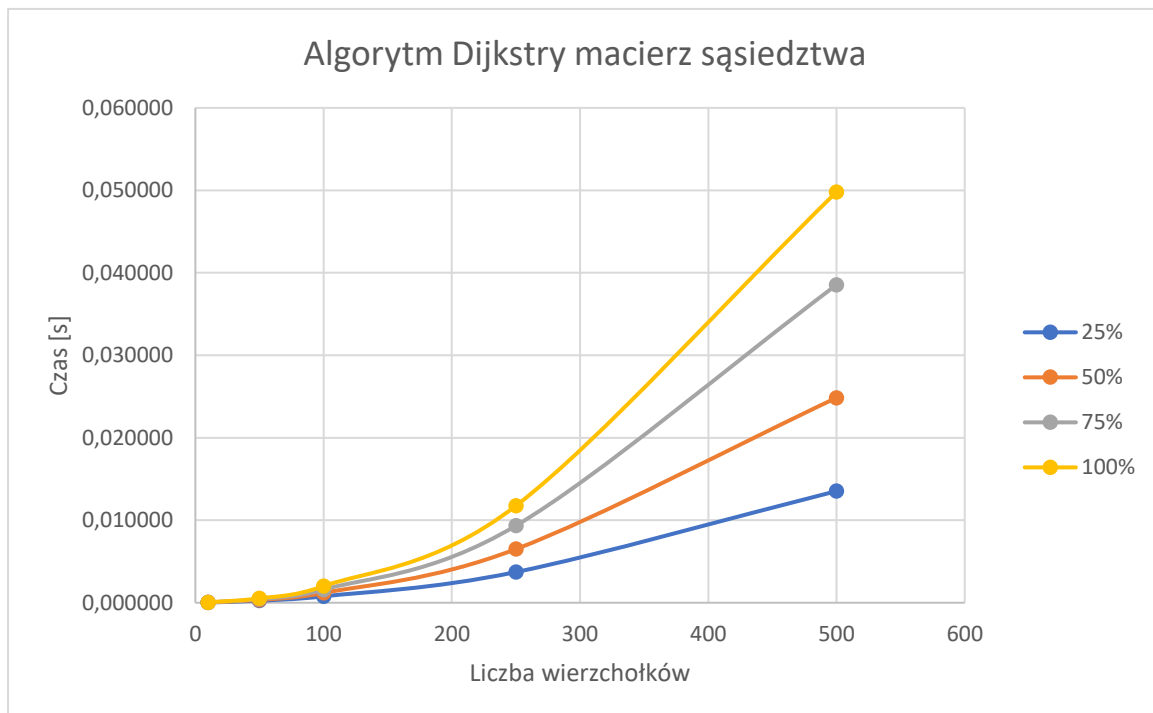
4. Wykresy

1. Typ1 (1 implementacja, wszystkie gęstości)

a) Z wykorzystaniem listy sąsiedztwa

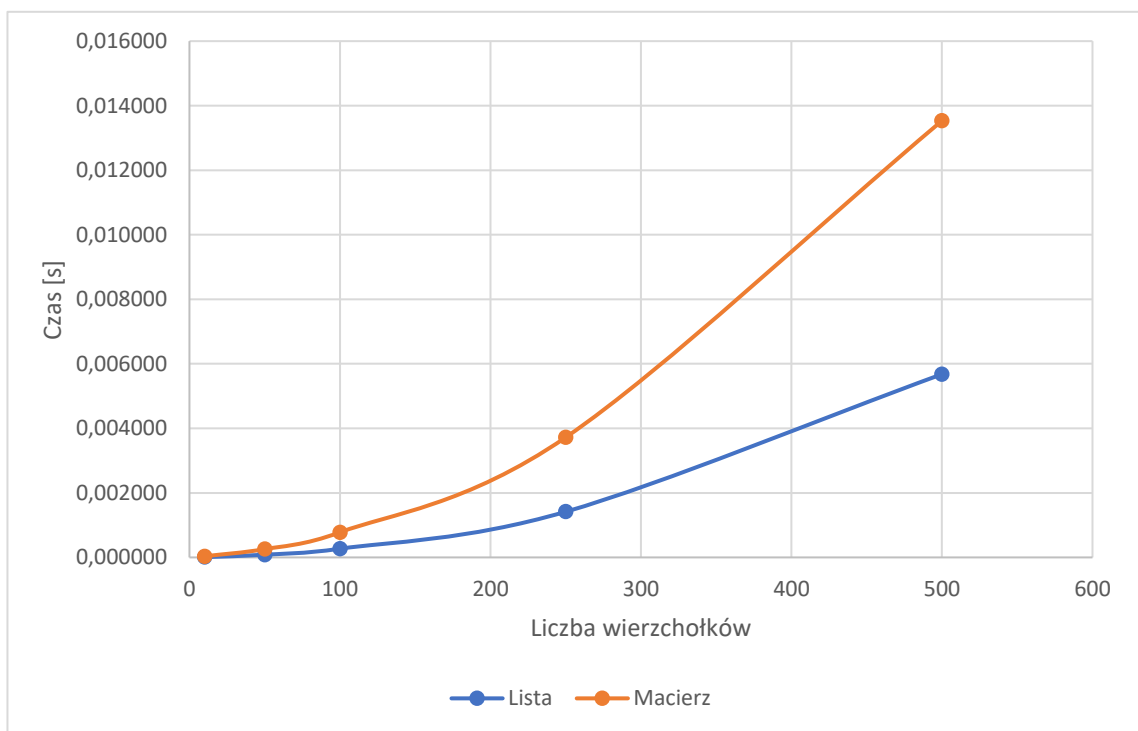


b) Z wykorzystaniem macierzy sąsiedztwa

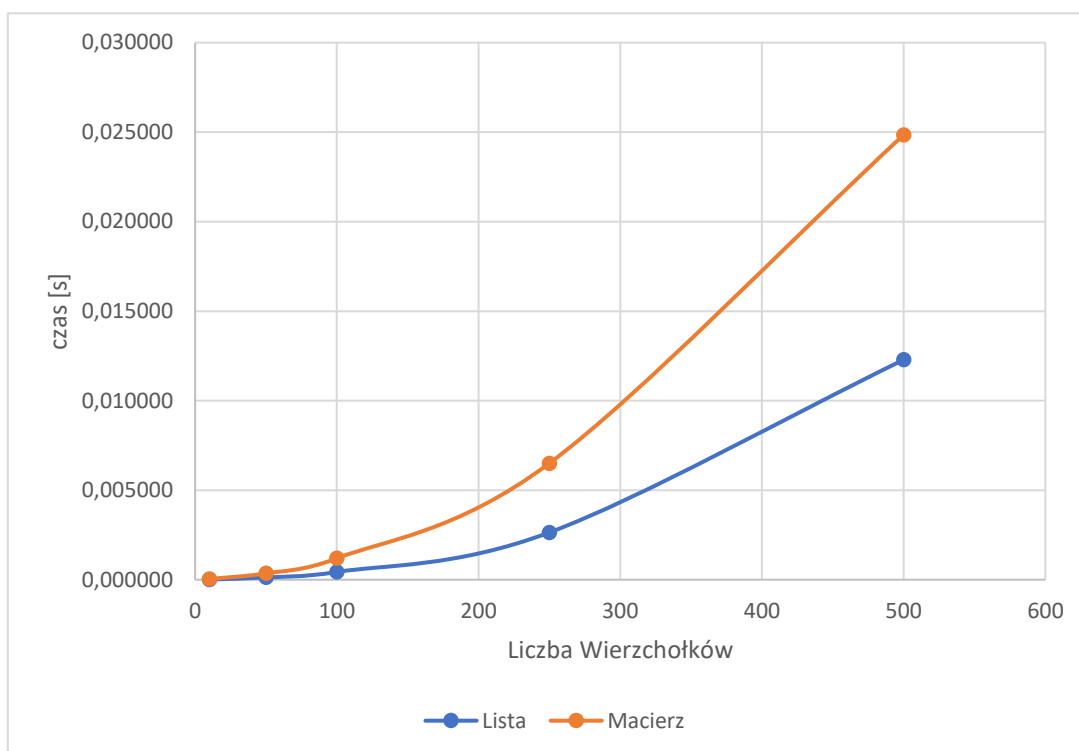


2. Typ2 (1 gęstość, dwie implementacje)

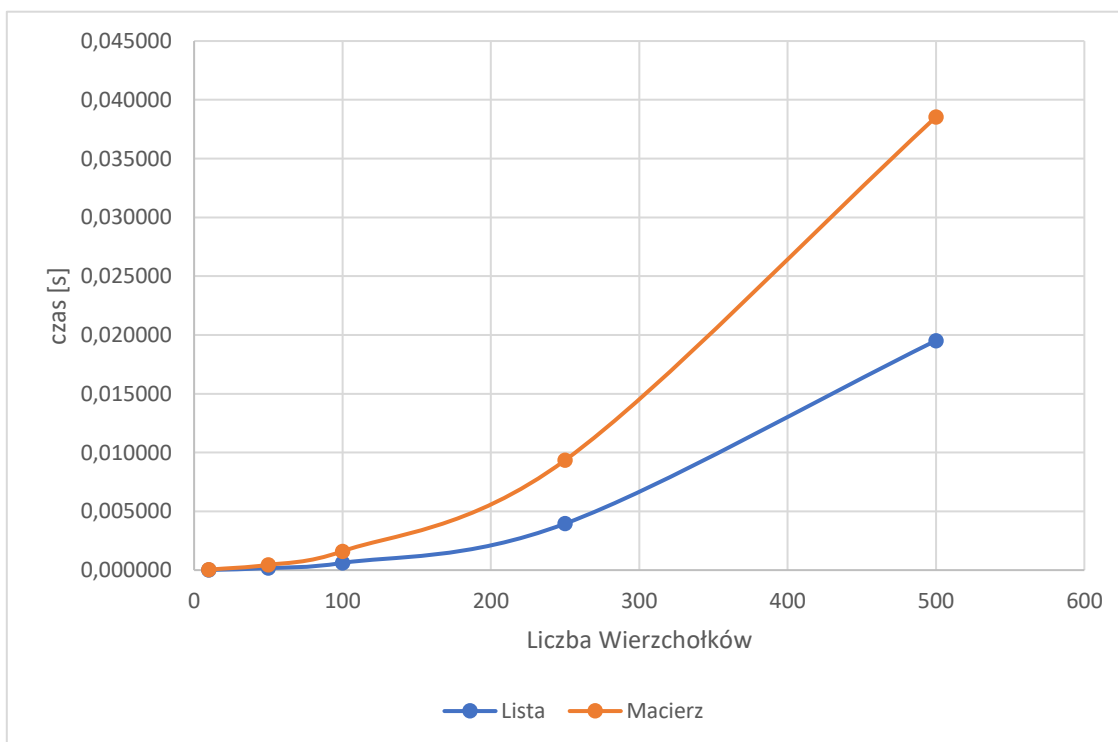
a) 25 %



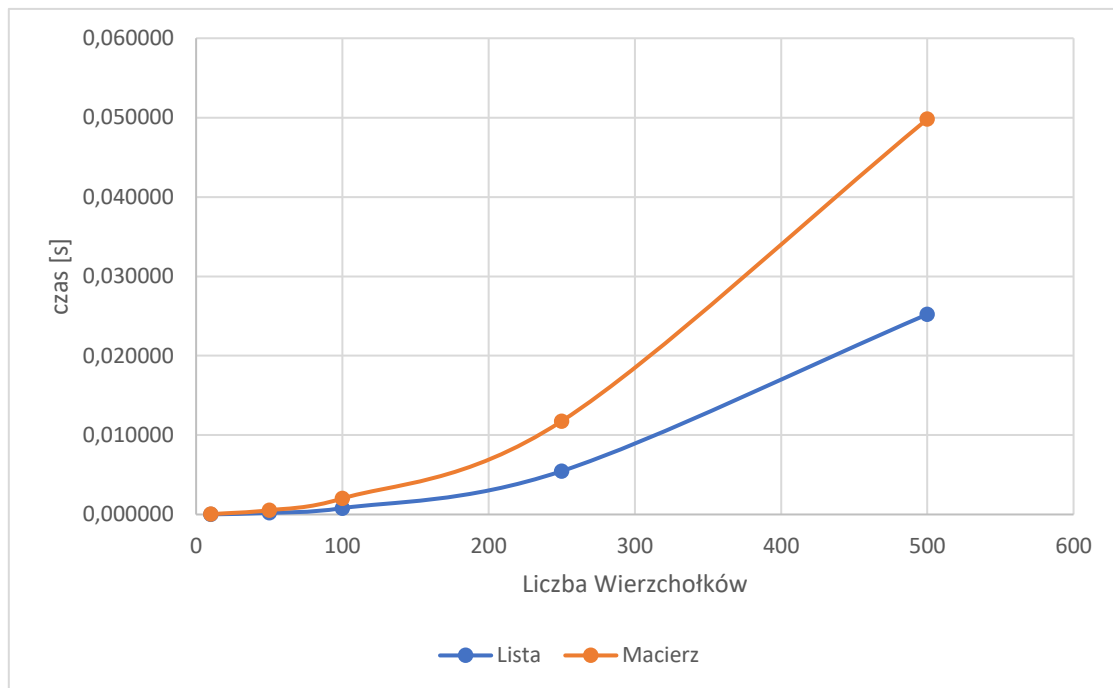
b) 50 %



c) 75%



d) 100%



5. Wnioski

Na podstawie przeprowadzonych testów można wywnioskować, że reprezentacja grafu za pomocą listy sąsiedztwa jest bardziej optymalna niż za pomocą macierzy sąsiedztwa. Dużą przewagę listy sąsiedztwa widać na wykresach typu 2, gdzie przyrost czasu zależny od gęstości jest szybszy niż dla listy. Ponadto widać też zależność, że im większa liczba wierzchołków w grafie tym większy czas wykonywania algorytmu. To samo tyczy się gęstości. Zmiana liczby wierzchołków ma większy wpływ gdy jest ona mała, im większa całkowita liczba wierzchołków tym mniej zmiana ich liczby wpływa na czas wykonania algorytmu.

6. Bibliografia

1. Materiały z wykładów
2. https://pl.wikipedia.org/wiki/Algorytm_Dijkstry [dostęp 03.05.2020]
3. https://pl.wikipedia.org/wiki/Reprezentacja_grafu [dostęp 30.04.2020]
4. https://zasoby1.open.agh.edu.pl/dydaktyka/matematyka/c_badania_operacyjne/krok/krok2_01.html [dostęp 03.05.2020]