

Projekt 2 - Grafy

Nazwa kursu	
Projektowanie Algorytmów i Metody Sztucznej Inteligencji	
Autor	Data
Dominik Koperkiewicz 254023	18.05.2021
Prowadząca	Termin zajęć
Mgr inż. Marta Emirsajłow	Wt 15:15

Wprowadzenie

Problemem postawionym w zadaniu był tak zwany Problem najkrótszej ścieżki. Polega on na znalezieniu takiego połączenia między wierzchołkami grafu, aby suma wag zawartych w ścieżce była jak najkrótsza. Do rozwiązania tego problemu posłużono się algorytmem Dijkstry. Dla wybranego wierzchołka startowego podawane są najkrótsze znalezione ścieżki do każdego wierzchołka, oraz ścieżka jaką należy poruszać się po grafie.

Reprezentacja Grafów

Algorytm został zaimplementowany i przetestowany dla dwóch reprezentacji grafów:

- Macierzy Sąsiedztwa
- Listy Sąsiedztwa

Macierz sąsiedztwa jest tablicą dwu wymiarową gdzie indeksy tablicy reprezentują wierzchołki grafu, a pola reprezentują krawędzie łączące wierzchołki o indeksach tego pola.

Lista Sąsiedztwa jest listą/tablicą w której każdy element jest oddzielną listą przechowującą wszystkie wierzchołki sąsiadujące z wierzchołkiem o indeksie tego elementu.

Testy wydajności

Testy zostały przeprowadzone dla czterech różnych gęstości grafów oraz pięciu różnych ilości wierzchołków, dla obu reprezentacji grafów. Przy każdym teście użyto 100 grafów, a następnie uśredniono czas (podany w mikrosekundach). Wyniki prezentują się następująco:

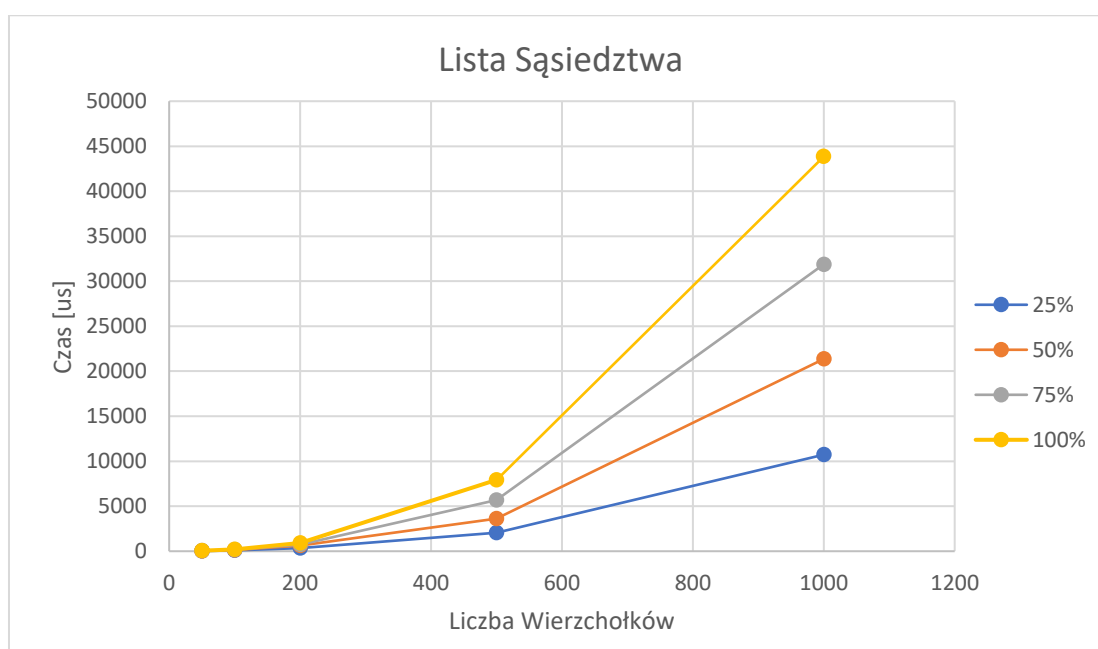
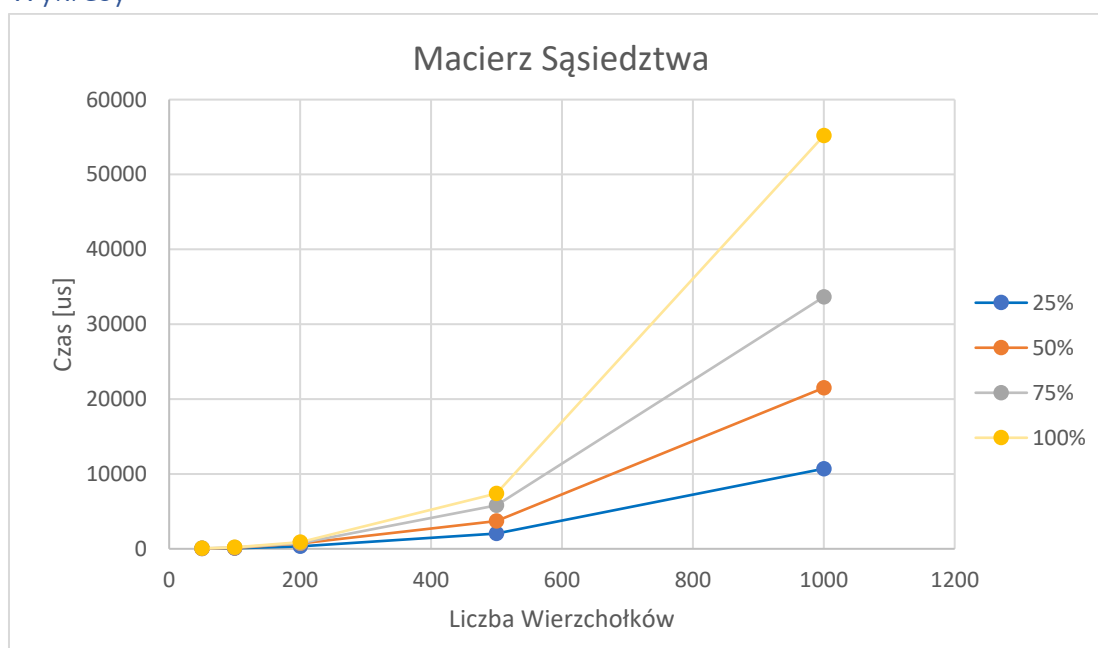
Macierz Sąsiedztwa

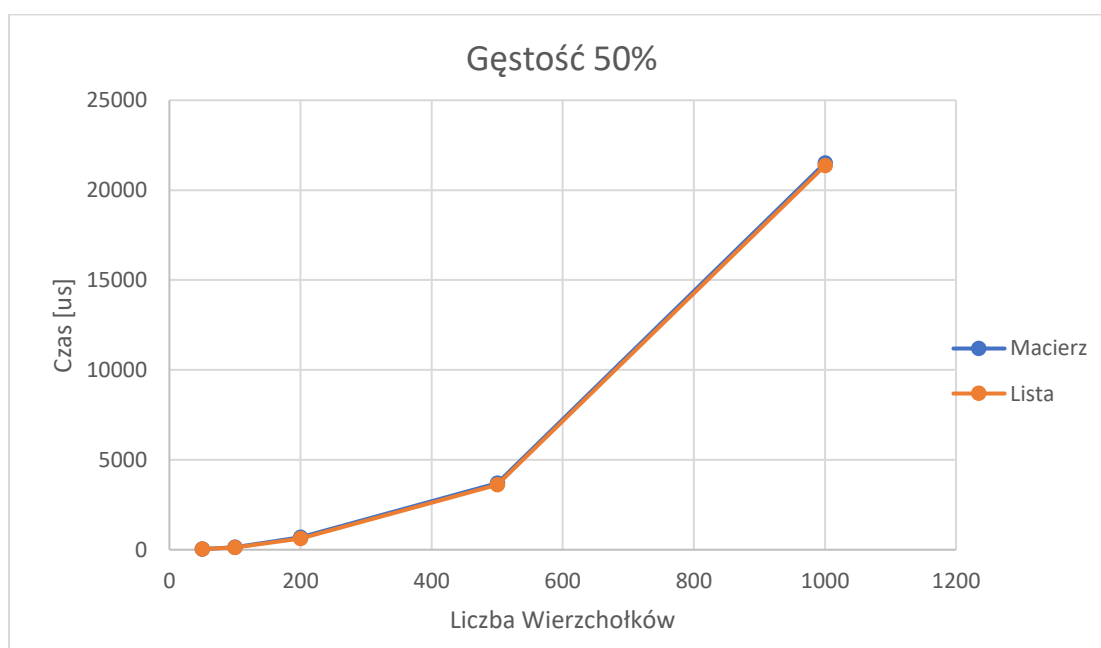
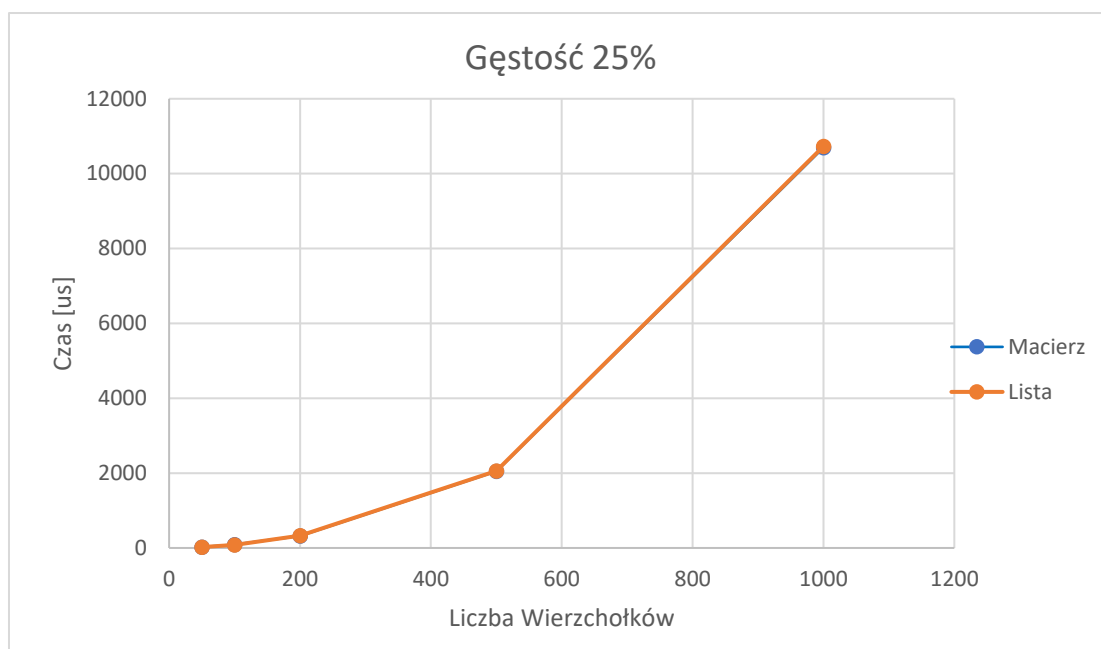
*	50	100	200	500	1000
25%	23,83	87,54	317,55	2050,06	10695,57
50%	34,06	131,46	694,82	3693,53	21501,32
75%	52,64	173,17	761,8	5799,54	33645,82
100%	51,67	199,52	867,33	7380,63	55184,41

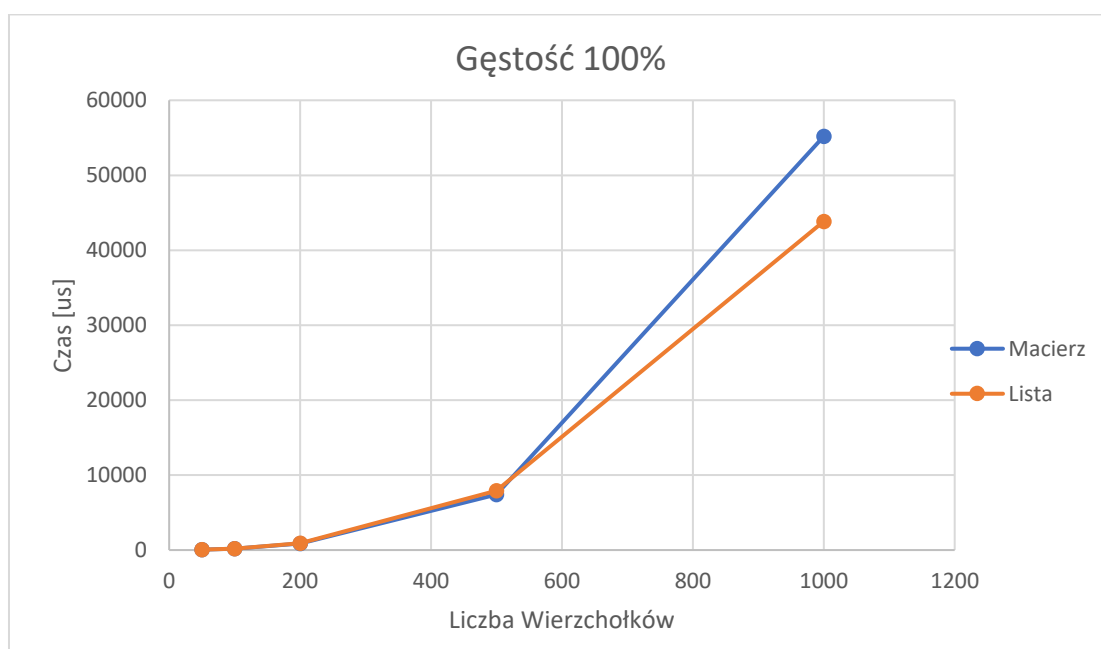
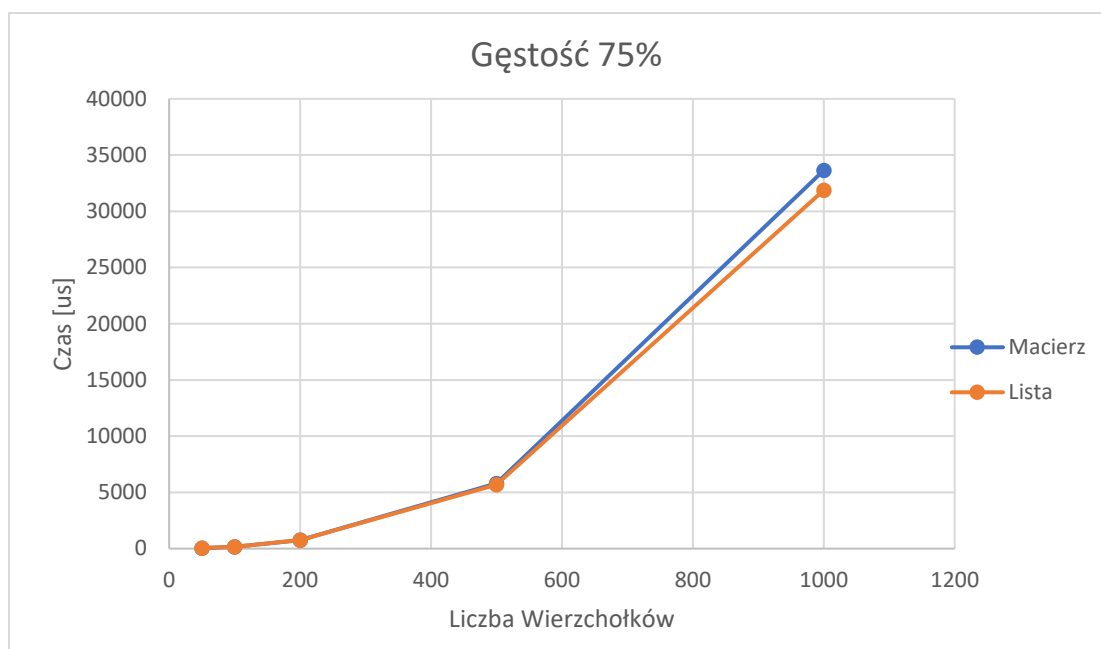
Lista Sąsiedztwa

*	50	100	200	500	1000
25%	23,28	84,13	332,33	2055,3	10728,51
50%	37,64	128,63	624,93	3619,01	21358,79
75%	53,27	169,85	754,69	5684,63	31877,59
100%	59,42	198,24	923,18	7909,45	43847,09

Wykresy







Złożoność obliczeniowa

Zaimplementowany algorytm do wyszukiwania najmniejszego elementu wykorzystywał przeszukiwanie liniowe w wyniku czego jego pesymistyczna złożoność obliczeniowa wynosi $O(V^2)$. Stosując kolejkę priorytetową w formie kopca można by zoptymalizować algorytm tak że jego złożoność obliczeniowa wynosiła by $O(E + V \cdot \log V)$.

Jak można zauważyć na wykresach, czas wykonywania algorytmu wzrasta wraz z gęstością grafów. Dla małych oraz mało gęstych grafów, algorytm dla obu reprezentacji radził sobie podobnie. Natomiast w przypadku większych i gęstszych grafów, reprezentacja poprzez listę sąsiedztwa była zdecydowanie lepsza.

Literatura

- https://pl.wikipedia.org/wiki/Algorytm_Dijkstry
- https://eduinf.waw.pl/inf/alg/001_search/0138.php
- <https://www.youtube.com/watch?v=pVfj6mxhdMw>