

Projektowanie Algorytmów i Metody Sztucznej Inteligencji - Grafy

Robert Krzyżoś 241541

March 2019

Prowadzący: dr inż. Krzysztof Halawa

Zajęcia: wtorek 13

1 Wstęp

Celem projektu było wyznaczenie najkrótszej drogi (ścieżki) w grafie między dwoma wierzchołkami, czyli znalezieniu najkrótszego połączenia pomiędzy tymi punktami. Każda krawędź posiadała swoją wagę, zatem za najkrótszą drogę uznano połączenie, o najmniejszym koszcie zsumowanych wag. W zadaniu za cel postawiono sobie wyznaczenie najkrótszych ścieżek łączących wszystkie wierzchołki z wierzchołkiem startowym. W tym celu w programie wykorzystano algorytm Dijkstry. Grafy występujące w projekcie to grafy skierowane, spójne reprezentowane przez macierz sąsiedztwa oraz listę sąsiedztwa. Kolejka priorytetowa została zaimplementowana w formie kopce.

2 Badanie efektywności

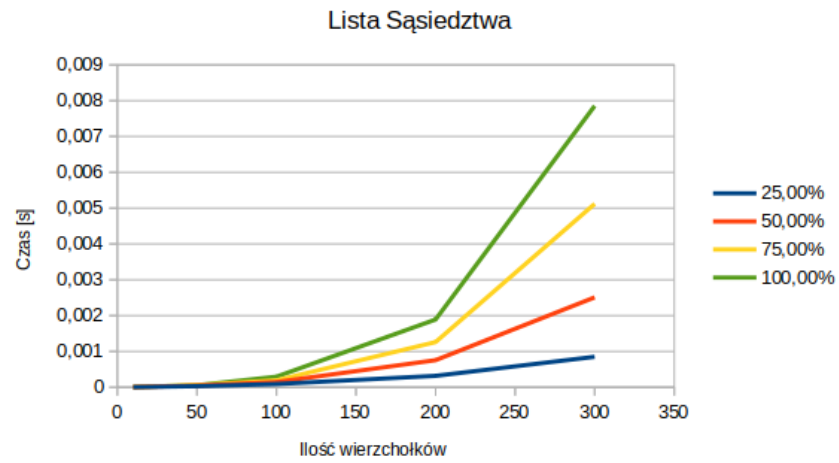
Badania wykonano dla 5 różnych ilości wierzchołków [10, 50, 100, 200, 300], kolejno dla różnych gęstości: 25%, 50%, 75% oraz dla grafu pełnego. Wyniki przedstawiono poniżej:

ListaSasiedztwa					
gest\rozmiar	10	50	100	200	300
25,00%	0,00000271	0,00002842	0,00008957	0,00031729	0,00084861
50,00%	0,00000312	0,00004339	0,00014454	0,00075411	0,00250884
75,00%	0,00000386	0,00005717	0,00019046	0,00126028	0,00511726
100,00%	0,00000383	0,00005380	0,00029299	0,00188880	0,00785335

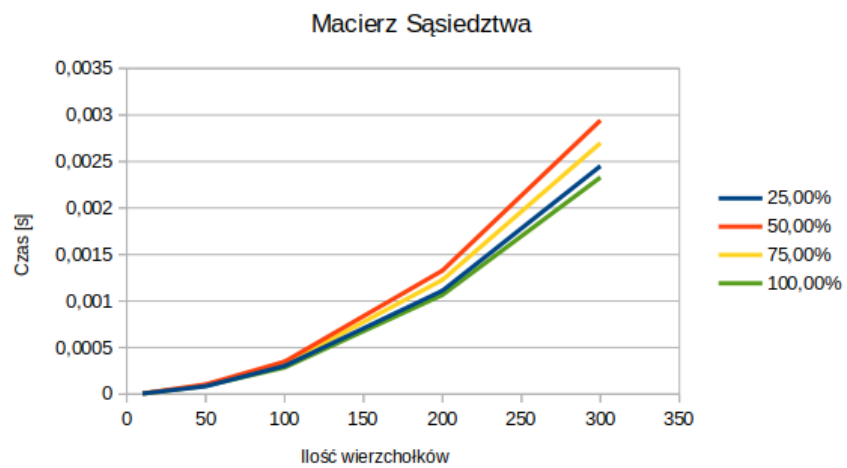
Rysunek 1: Tabela z pomiarami czasów dla listy sąsiedztwa.

MacierzSasiedztwa					
gest\rozmiar	10	50	100	200	300
25,00%	0,00000575	0,00008379	0,00030294	0,00111184	0,00245166
50,00%	0,00000509	0,00010450	0,00034869	0,00132835	0,00294229
75,00%	0,00000556	0,00009368	0,00032321	0,00122833	0,00269910
100,00%	0,00000594	0,00008961	0,00028561	0,00106698	0,00232980

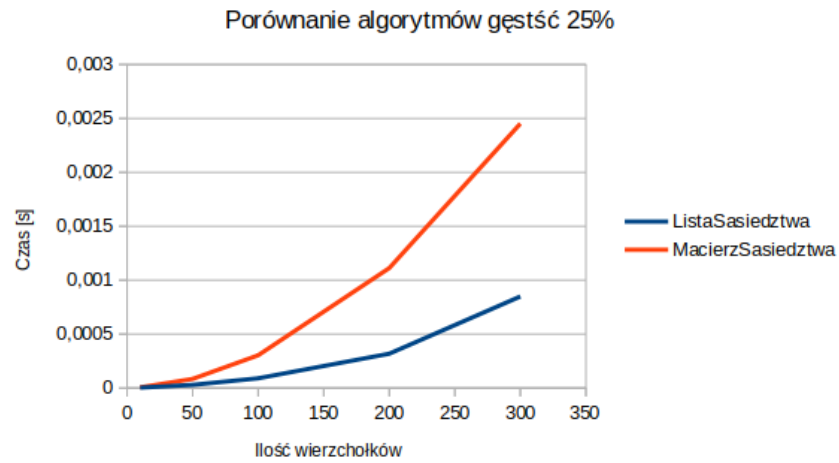
Rysunek 2: Tabela z pomiarami czasów dla macierzy sąsiedztwa.



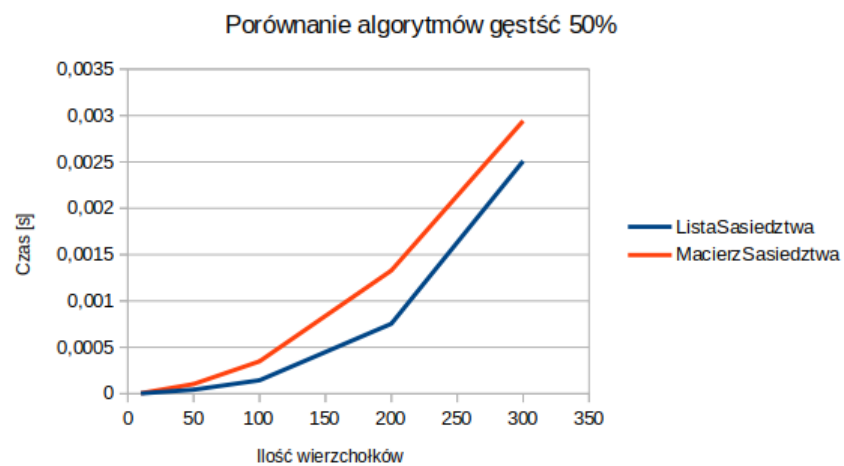
Rysunek 3: Wykres z wynikami pomiarów dla listy sąsiedztwa.



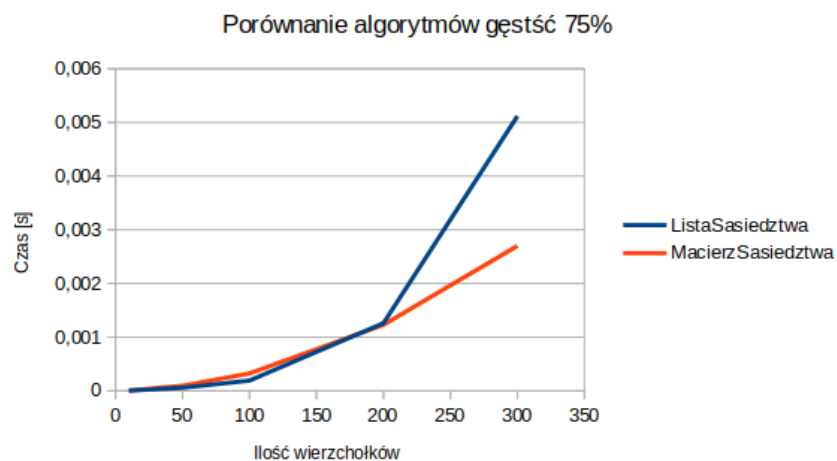
Rysunek 4: Wykres z wynikami pomiarów dla macierzy sąsiedztwa.



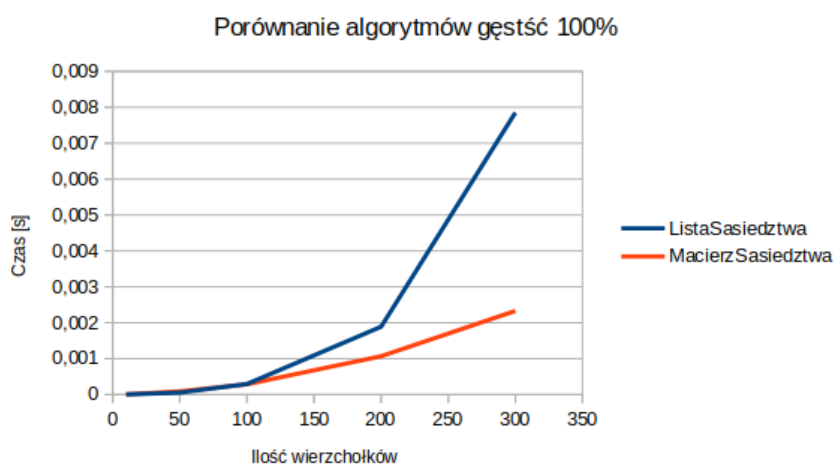
Rysunek 5: Porównanie czasów działania algorytmu w zależności od reprezentacji grafu przy gęstości 25%



Rysunek 6: Porównanie czasów działania algorytmu w zależności od reprezentacji grafu przy gęstości 50%



Rysunek 7: Porównanie czasów działania algorytmu w zależności od reprezentacji grafu przy gęstości 75%



Rysunek 8: Porównanie czasów działania algorytmu w zależności od reprezentacji grafu przy gęstości 100%

3 Wnioski

Algorytm działa dość szybko. Dzięki zastosowaniu implementacji kolejki priorytetowej w formie kopca złożoność obliczeniowa wynosi $O(V * \log V)$ dla macierzy sąsiedztwa oraz $O(E * \log V)$ dla listy sąsiedztwa. Jak się okazuje duże znaczenie ma także reprezentacja grafu w programie. Dla gęstości do ok. 60-70% nieco szybszy jest algorytm wykorzystujący listę sąsiedztwa. Później natomiast, gdy gęstość rośnie widać znaczącą różnicę czasu trwania zadania na korzyść reprezentacji z macierzą sąsiedztwa.

Literatura

- [1] Eduinf Grafy https://eduinf.waw.pl/inf/alg/001_search/0122.php
- [2] Wikipedia Algorytm Dijkstry
- [3] GeeksforGeeks
- [4] 4programmers.net forum