

Autor: Remigiusz Mielcarz 252887

Data wykonania: 10.05.2021r

Grupa: E12-99f Wt. 15:15

Projektowanie algorytmów i metody sztucznej inteligencji

Grafy

1 Cel ćwiczenia

Celem projektu jest zapoznanie się z strukturami danych zwanymi jako grafy, ich zasadnością oraz możliwościami. Są one reprezentowane za pomocą listy oraz macierzy sąsiedztwa.

Kolejnym celem jest poznanie algorytmu Bellmana-Forda, który służy do rozwiązania problemu najkrótszej ścieżki w grafie między dwoma wierzchołkami.

Następnym celem jest zbadanie kiedy powinno stosować się implementację na liście, a kiedy implementację na macierzy.

Algorytm został poddany testom efektywności 100 grafów o różnych ilościach wierzchołków:

10, 50, 100, 500, 100 oraz z różną gęstością: 25%, 50%, 75%, 100%.

2 Algorytm Bellmana-Forda

Algorytm Bellmana-Forda służy do wyszukiwania najkrótszych ścieżek w grafie ważonym z wierzchołką źródłowego do wszystkich pozostałych wierzchołków. Idea algorytmu opiera się na metodzie relaksacji (dokładniej następuje relaksacja $|V| - 1$ dla każdej krawędzi).

W odróżnieniu od algorytmu Dijkstry algorytm Bellmana Forda działa poprawnie także dla grafów z wagami ujemnymi (nie może jednak wystąpić cykl o łącznej ujemnej wadze osiągalny ze źródła). Za tę ogólność płaci się jednak wyższą złożoność czasową. Algorytm Bellmana-Forda działa w czasie $O(|V| * |E|)$, gdzie złożoność czasowa algorytmu Dijkstry to $O(E * \log_2 V)$. Złożoność pamięciowa algorytmu Bellmana-Forda to $O(|V|)$.

3 Przebieg testów

Po otwarciu programu następuje proces generowania grafów do pliku o nazwie dane.txt.

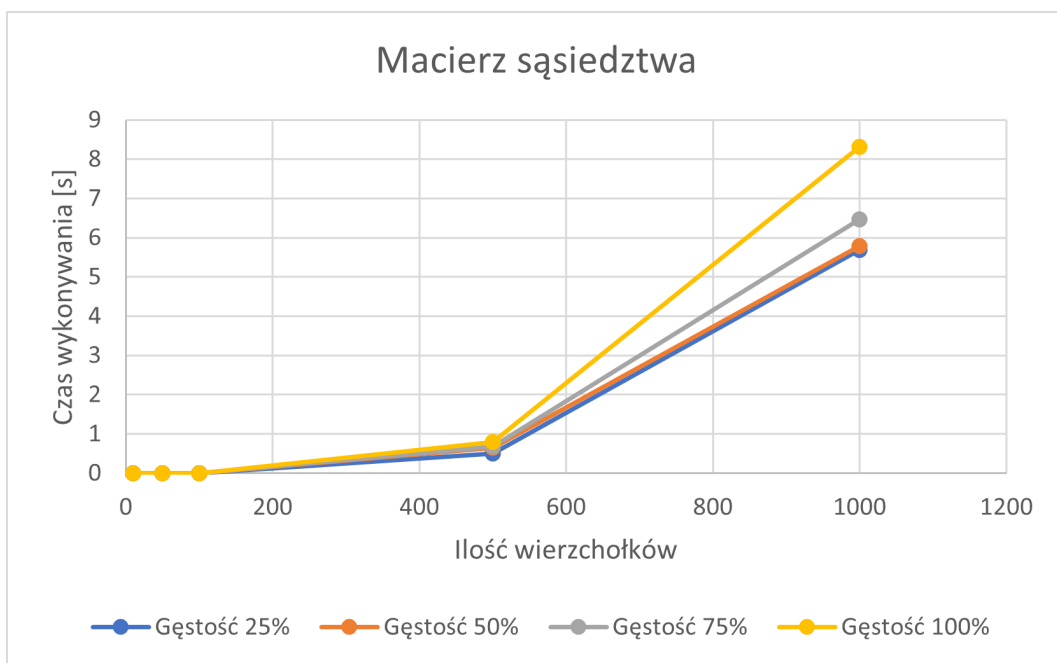
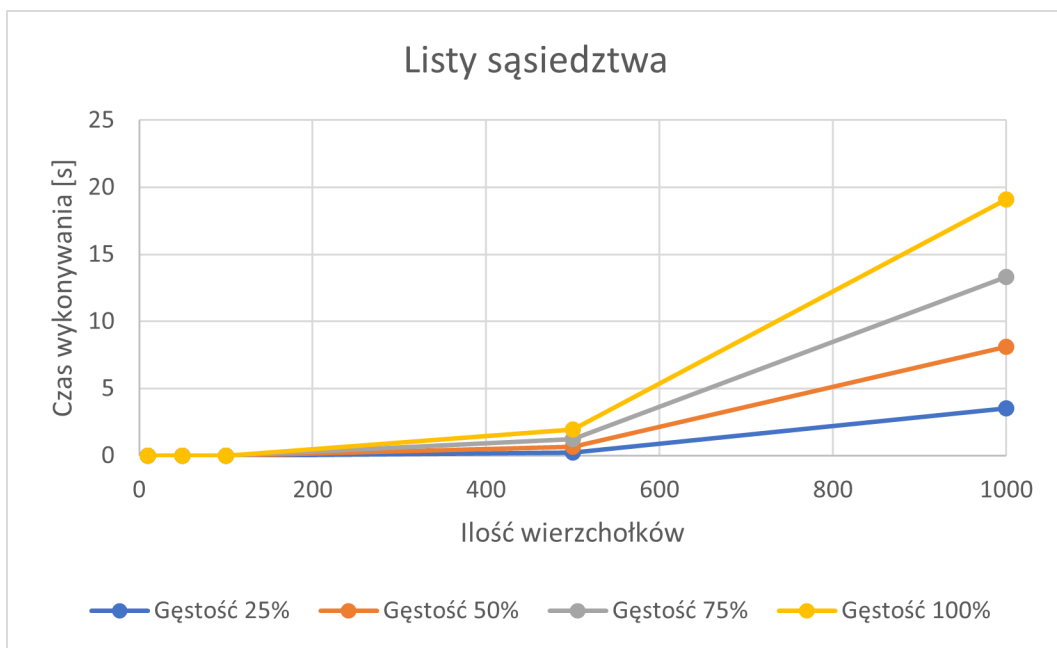
Wygenerowane grafy mają: 10, 50, 100, 500, 1000 wierzchołków, w tym: 25%, 50%, 75%, 100% wypełnienia. Generujemy 100 grafów dla każdej sytuacji. Następnie rozpoczyna się proces wczytywania grafów i poszukiwania najkrótszej ścieżki poprzez algorytm Bellmana-Forda. Wyniki pomiarów zapisują się do pliku o nazwie pomiar.txt.

4 Wyniki testów

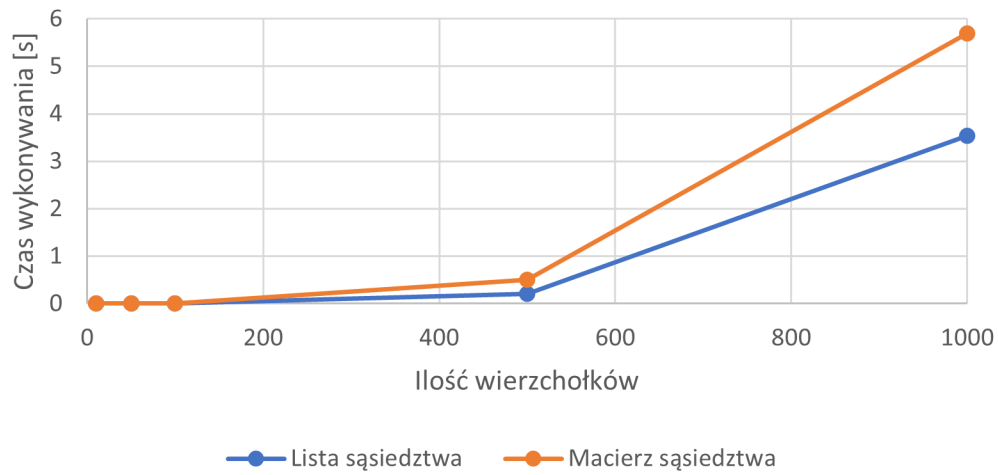
Lista	10	50	100	500	1000
25%	0,000002	0,000128	0,001148	0,204354	3,529977
50%	0,000003	0,000312	0,002224	0,672085	8,095631
75%	0,000003	0,000461	0,003618	1,231669	13,297278
100%	0,000004	0,000613	0,004980	1,935260	19,115243

Macierz	10	50	100	500	1000
25%	0,000004	0,000441	0,003247	0,503792	5,691570
50%	0,000004	0,000488	0,003703	0,643618	5,787913
75%	0,000004	0,000543	0,004175	0,678702	6,462724
100%	0,000005	0,000608	0,004497	0,795100	8,317489

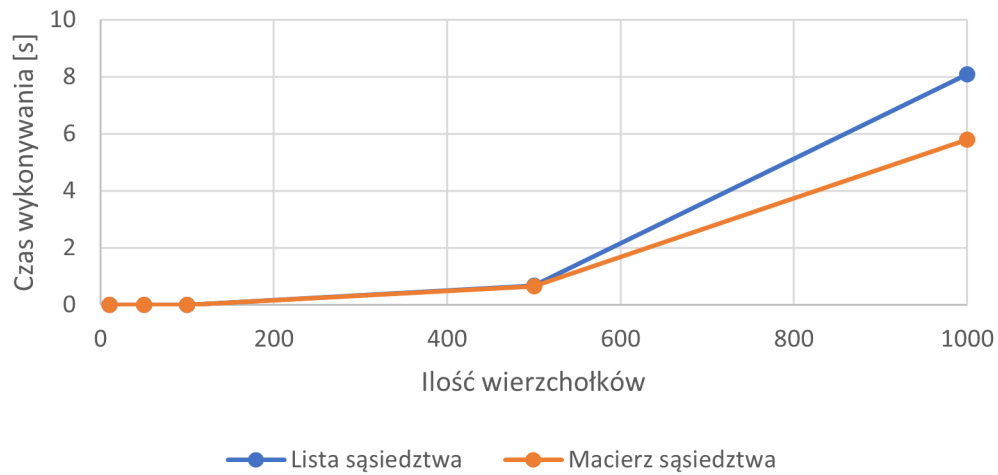
5 Wykresy



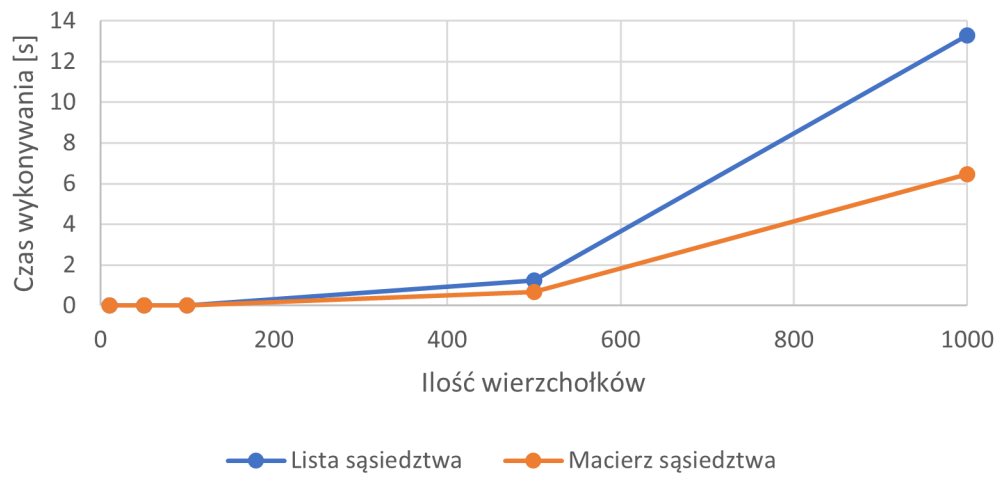
Efektywność algorytmu Bellmana-Forda dla gęstości 25%



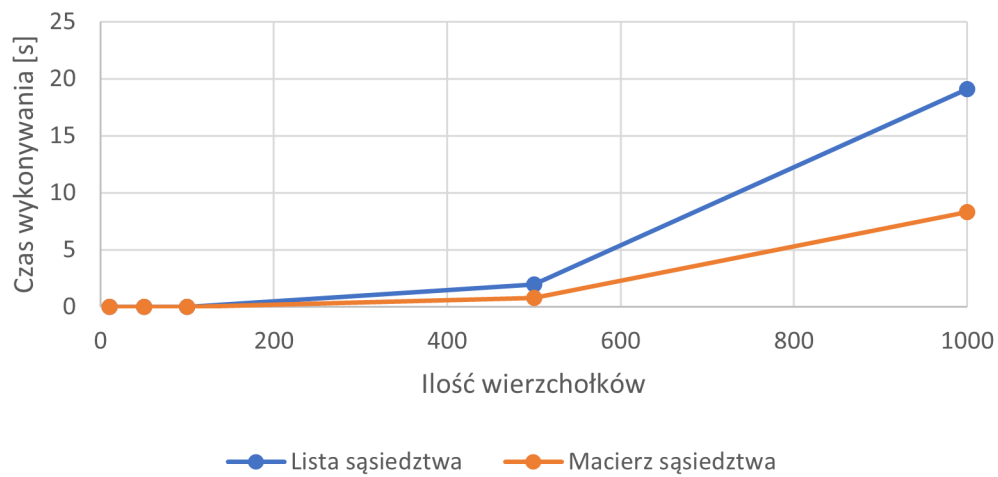
Efektywność algorytmu Bellmana-Forda dla gęstości 50%



Efektywność algorytmu Bellmana-Forda dla gęstości 75%



Efektywność algorytmu Bellmana-Forda dla gęstości 100%



6 Wnioski

- Algorytm Bellmana-Forda wykonuje się tym dłużej, im więcej mamy wierzchołków w naszym grafie.
- Algorytm Bellmana-Forda jest efektywniejszy dla grafów o mniejszej gęstości.
- Znajdowanie najkrótszej ścieżki przy reprezentacji grafu za pomocą listy sąsiedztwa wykonuje się krócej niż przy reprezentacji za pomocą macierzy sąsiedztwa dla gęstości równej 25%.
- Dla gęstości 50% czasy są zbliżone.
- Dla gęstości grafu równej 75% i 100% bardziej efektywna jest reprezentacja grafu za pomocą macierzy sąsiedztwa.
- Czasy znajdowania najkrótszej ścieżki przy takiej samej liczbie wierzchołków w reprezentacji macierzy sąsiedztwa są bardzo podobne, gęstość grafu ma na nie mały wpływ.
- Czasy znajdowania najkrótszej ścieżki przy takiej samej liczbie wierzchołków w reprezentacji listy sąsiedztwa różnią się znacząco, a zatem gęstość grafu ma na nie wpływ.

7 Literatura

- https://pl.wikipedia.org/wiki/Algorytm_Bellmana-Forda
- <https://eduinf.waw.pl/inf/alg/001/search/0138a.php>
- <https://www.youtube.com/watch?v=mV4KpPqKqLA>

Wykonano w systemie L^AT_EX