

Projektowanie Algorytmów i Metody Sztucznej Inteligencji

Projekt 2

Termin	poniedziałek 13:15
Prowadzący projekt	Mgr inż. Marta Emirsajłow
Temat	Badanie efektywności algorytmu Bellmana-Forda
Imię i Nazwisko	Dominik Polak

1. Wstęp

Graf – powszechnie stosowana struktura danych we współczesnej informatyce. Składa się ona z dwóch zbiorów: zbioru wierzchołków nazywanych też węzłami oraz krawędzi łączących te wierzchołki. Grafy można reprezentować na wiele sposobów. W naszym przypadku będzie graf reprezentować w postaci macierzy oraz list sąsiedztwa.

Macierz sąsiedztwa – jest to macierz kwadratowa o stopniu n , gdzie n oznacza liczbę wierzchołków w naszym grafie. Jest ona reprezentowana przez tablicę dwuwymiarową, gdzie w poszczególnych komórkach są zapisane wagi poszczególnym krawędzi grafu i jeśli taka krawędź nie istnieje to przypisujemy wartość 0 albo NULL.

Lista sąsiedztwa – często okazuje się, że jest najefektywniejszą metodą reprezentacji grafu. Dla każdego wierzchołka zapamiętywana jest lista sąsiadujących ze sobą węzłów. Do jej utworzenia wykorzystujemy n -elementową tablicę, gdzie n to liczba wierzchołków, a każdy element tej tablicy jest listą.

Gęstość grafu – jest to stosunek liczby krawędzi do największej możliwej liczby krawędzi.

Algorytm Bellmana-Forda – algorytm służący do wyszukiwania najkrótszych ścieżek w grafie ważonym z wierzchołka źródłowego do wszystkich pozostałych wierzchołków. Idea tego algorytmu opiera się na metodzie relaksacji, dokładniej następuje relaksacja $|V|-1$ razy dla każdej z krawędzi. Algorytm ten także działa poprawnie dla grafów z wagami ujemnymi, jednak nie może wystąpić cykl o łącznej ujemnej wadze. Za tę niedogodność płaci się jednak wyższą złożonością czasową gdyż działa on w czasie $O(|V| * |E|)$.

2. Wyniki

Test efektywności przeprowadzono dla 100 instancji grafów ważonych i skierowanych dla różnej liczby węzłów: 10, 50, 100, 500, 1000 oraz dla różnych gęstości grafu: 25%, 50%, 75% oraz grafu pełnego.

Na podstawie tych testów policzono średni czas działania algorytmu Bellmana-Forda dla jednego zaimplementowanego grafu.

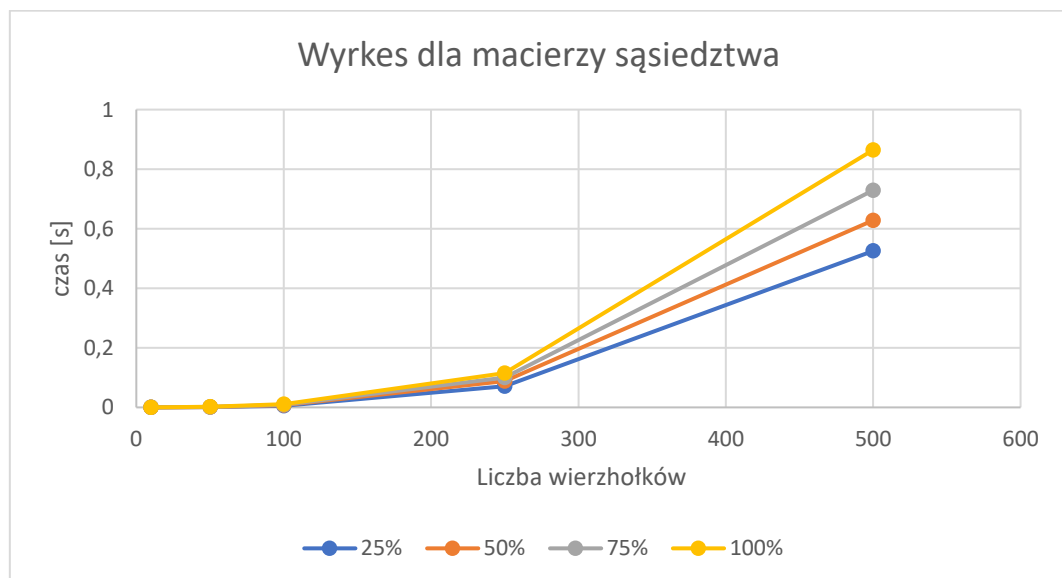
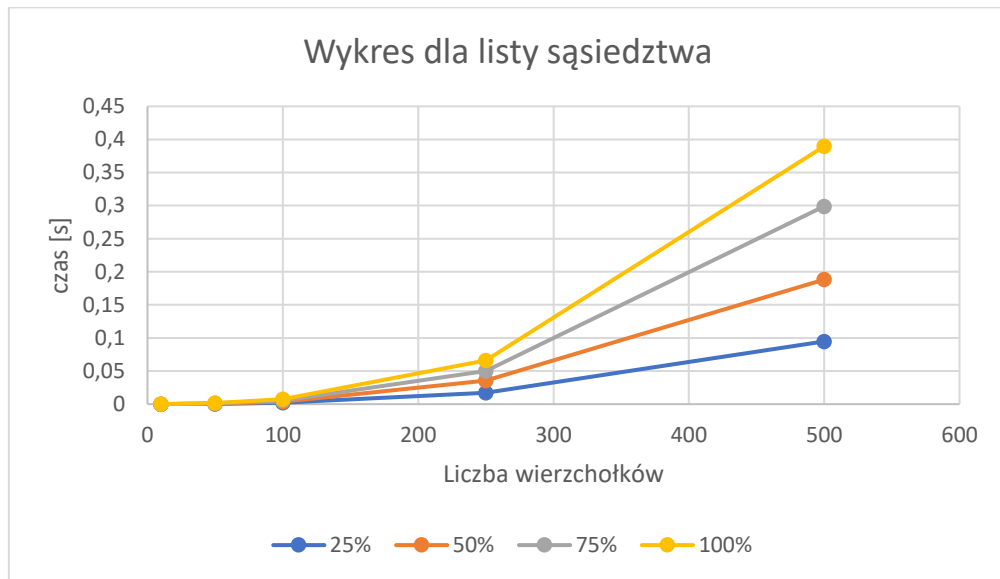
a. Implementacja grafu za pomocą listy sąsiedztwa

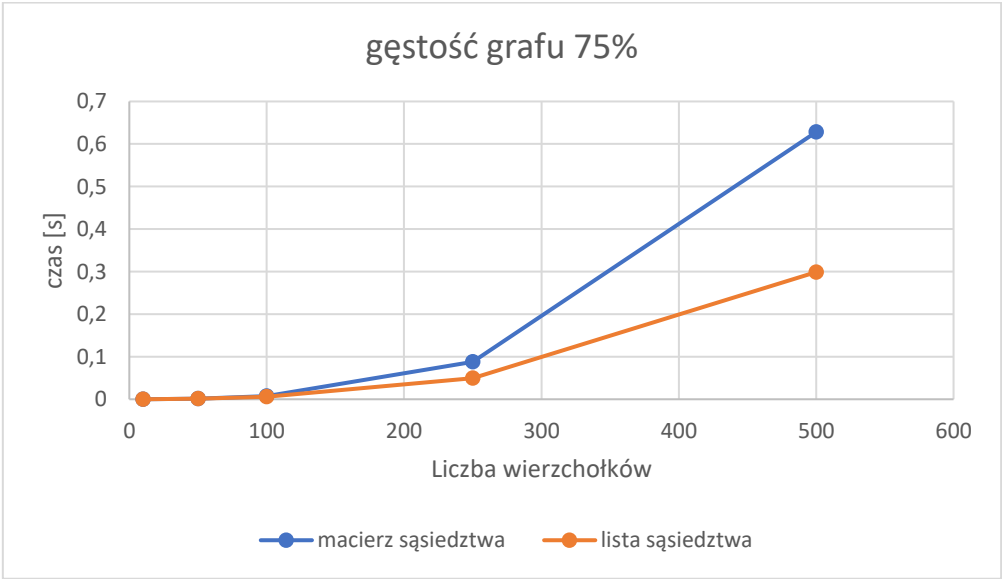
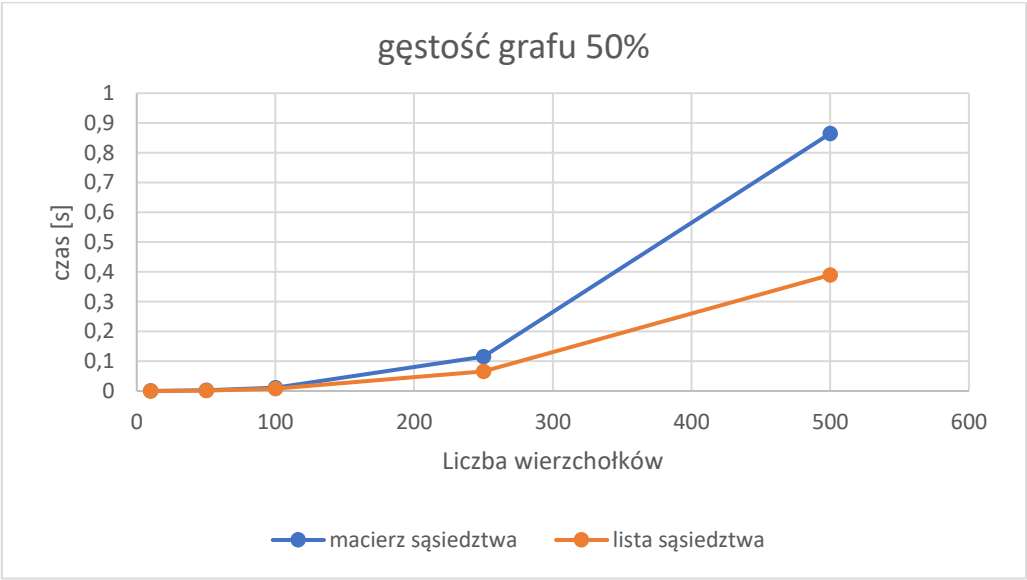
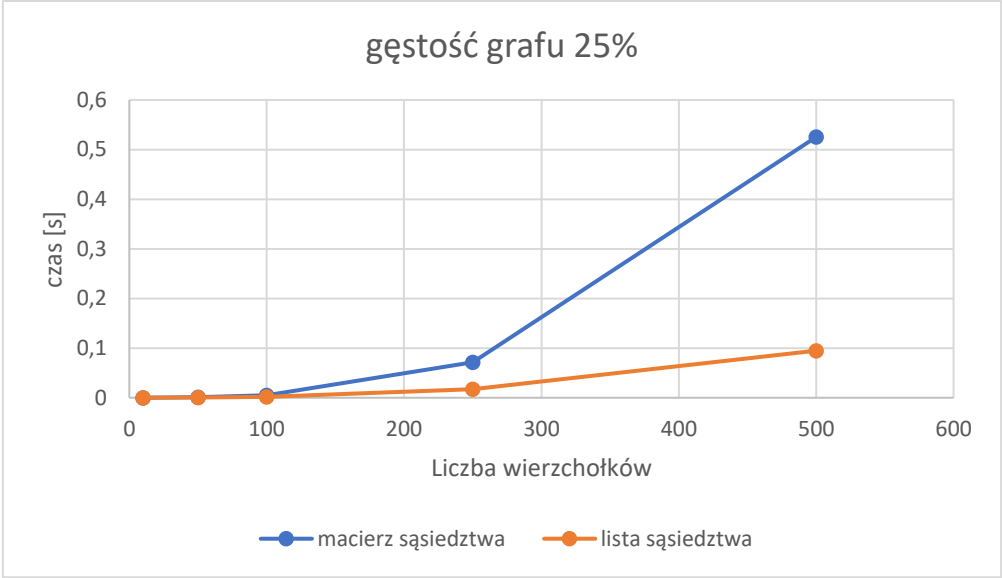
Gęstość/wierzchołki	10	50	100	250	500	Jednostka
25%	0.0000202	0.00041	0.0020	0.017	0.094	[s]
50%	0.0000302	0.00081	0.0038	0.035	0.19	[s]
75%	0.0000299	0.0012	0.0056	0.050	0.30	[s]
100%	0.0000601	0.0015	0.0075	0.066	0.39	[s]

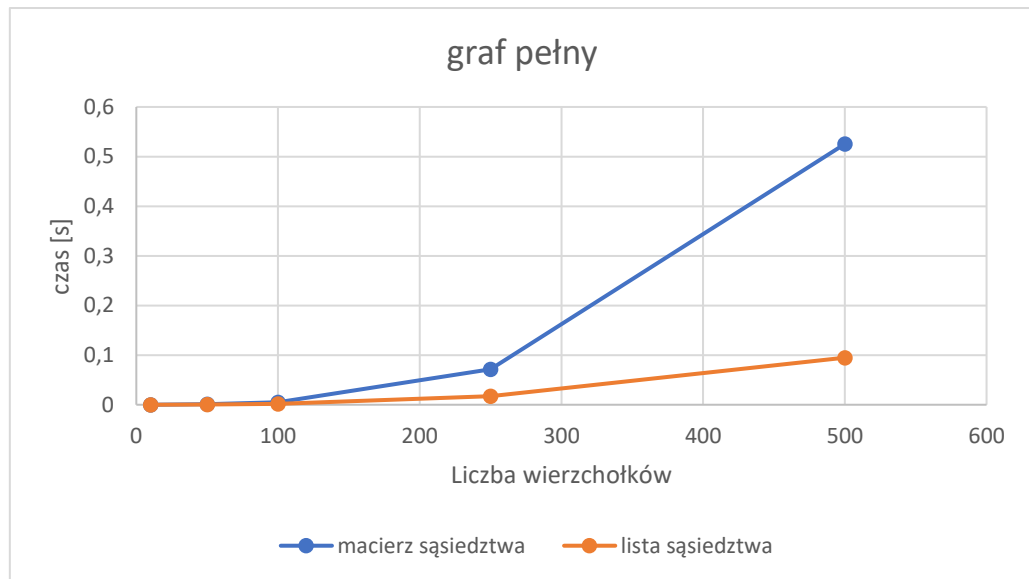
b. Implementacja grafu za pomocą macierzy sąsiedztwa

Gęstość/wierzchołki	10	50	100	250	500	Jednostka
25%	0.000020	0.00087	0.0053	0.072	0.53	[s]
50%	0.000030	0.0012	0.0072	0.088	0.63	[s]
75%	0.000040	0.0016	0.0091	0.099	0.73	[s]
100%	0.000050	0.0021	0.011	0.11	0.86	[s]

3. Wykresy







4. Wnioski

- Dla naszego przypadku grafu ważonego, skierowanego implementacja w postaci list jest efektywniejszym sposobem jego reprezentacji. Jak możemy zauważyć na wykresach średni czas wyszukiwania ścieżek od wierzchołka startowego do wszystkich innych przy pomocy algorytmu Bellmana_Forda dla reprezentacji grafu w postaci listy sąsiedztwa jest znacznie krótszy niż dla macierzy sąsiedztwa.
- Krzywe na wykresach w przybliżeniu pokrywają się ze złożonością obliczeniową algorytmu. Czas wykonania algorytmu zwiększał się wraz ze wzrostem liczby krawędzi tym samym algorytm działał najdłużej dla grafu pełnego.

5. Bibliografia

- Drozdek A., C++. Algorytmy i struktury danych, Helion
- https://eduinf.waw.pl/inf/alg/001_search/0138a.php
- https://eduinf.waw.pl/inf/alg/001_search/0124.php