

Projektowanie algorytmów i metody sztucznej inteligencji
Prowadzący: Mgr inż. Marcin Ochman
Wt. 15:15-16:55

PROJEKT 3

Autor:

SIERACKA WERONIKA

5 maja 2020

1 Wstęp:

Zadania polegało na znalezieniu najkrótszej ścieżki od wybranego wierzchołka do wszystkich pozostałych wierzchołków. Należało przy tym uwzględnić reprezentacje grafu w postaci macierzy sąsiedztwa oraz listy sąsiedztwa, a badania wykonać dla 5 różnych liczb wierzchołków w grafie oraz następujących gęstości grafu: 25%, 50%, 75% oraz dla grafu pełnego. Dla każdego zestawu parametrów: algorytm, reprezentacja grafu, liczba wierzchołków i gęstość grafu należało wygenerować po 100 losowych instancji, a w sprawozdaniu umieszczono wyniki uśrednione.

2 Algorytm Bellmana-Forda

Algorytm służący do wyznaczania najkrótszych ścieżek w grafie. W odróżnieniu od algorytmu Dijkstry, algorytm Bellmana-Forda dopuszcza krawędzie o ujemnych wagach, nie mogą istnieć jednak ujemne cykle osiągalne z wierzchołka źródłowego. Na początku stworzono dwie tablice o rozmiarze odpowiadającym ilości wierzchołków. Jedna z tablic zawiera numery wierzchołków grafu które znajdują się w najkrótszej ścieżce dojścia do i-tego wierzchołka, a druga koszty dojścia z wierzchołka startowego do i-tego wierzchołka w grafie. Koszt dojścia do wierzchołka startowego wynosi 0, natomiast resztę tablicy wypełnia się nieskończonością, a tablice zawierającą numery wierzchołków wypełnia się wartością -1, ponieważ w grafie nie ma wierzchołka o takim numerze. Następnie są wykonywane obiegi pętli, podczas których ustala koszt dojścia do wierzchołka. Przegląda się po kolei wszystkie krawędzie grafu, jeśli znajdzie się drogę którą dojście od jednego wierzchołka do drugiego jest tańsze niż poprzednio znalezione drogi to ustawiamy koszt na koszt tańszej ścieżki, a także dodajemy ten wierzchołek do tablicy zawierającej drogę. Po wykonaniu wszystkich obiegów pętli, których jest o jeden mniej niż wierzchołków w grafie otrzymujemy znalezione wszystkie drogi.

3 Złożoność czasowa

Dla grafu liczącego n wierzchołków i e krawędzi złożoność pesymistyczna jest równa $O(en)$. Biorąc pod uwagę, że w przypadku braku krawędzi wielokrotnych liczba krawędzi jest zawsze mniejsza od n^2 , można powiedzieć, że złożoność czasowa algorytmu to $O(n^3)$.

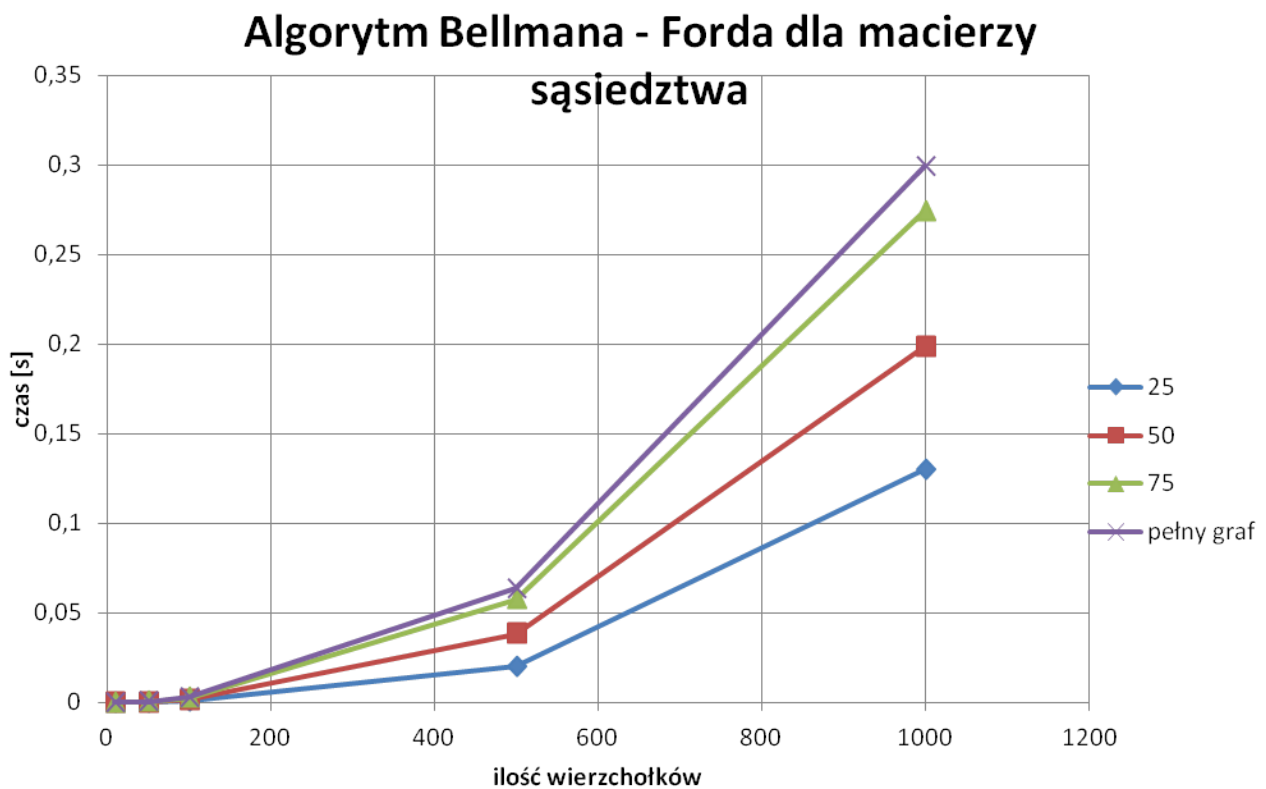
4 Wyniki macierz sąsiedztwa:

ilość wierzchołków	gęstość 25%	gęstość 50%	gęstość 75%	pełny graf
10	0,00000408s	0,00000828	0,0000135	0,0000174s
50	0,000151s	0,000318s	0,000501s	0,000566s
100	0.0011627s	0.0018599 s	0.0026496 s	0.00290819 s
500	0.020372 s	0.038369 s	0.057613 s	0.063932 s
1000	0.129988 s	0.198483 s	0.274282 s	0.299485 s

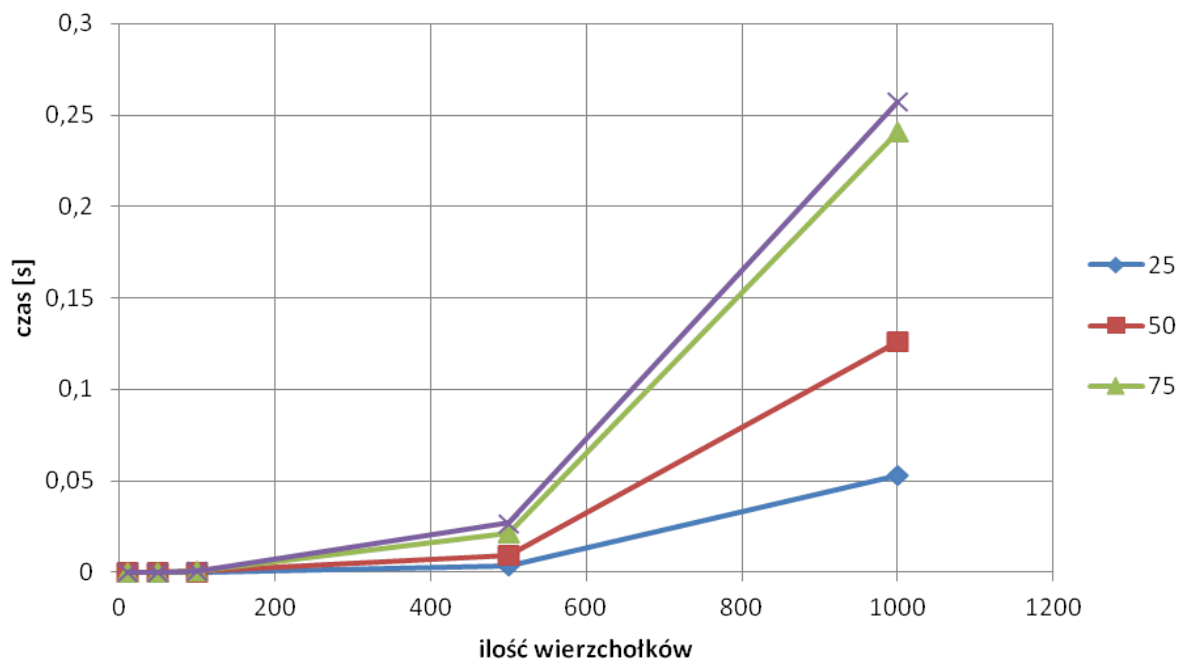
5 Wyniki lista sasiedztwa

ilość wierzchołków	gęstość 25%	gęstość 50%	gęstość 75%	pełny graf
10	0,00000128s	0,00000828s	0,00000467s	0,00000625s
50	0,0000265s	0,00004549s	0,00007713s	0,00009617s
100	0.00015091s	0.00024849s	0.00038609s	0.00046042s
500	0.00329111s	0.00921314s	0.0215267s	0.0268037s
1000	0.0532206s	0.12589s	0.240636s	0.257s

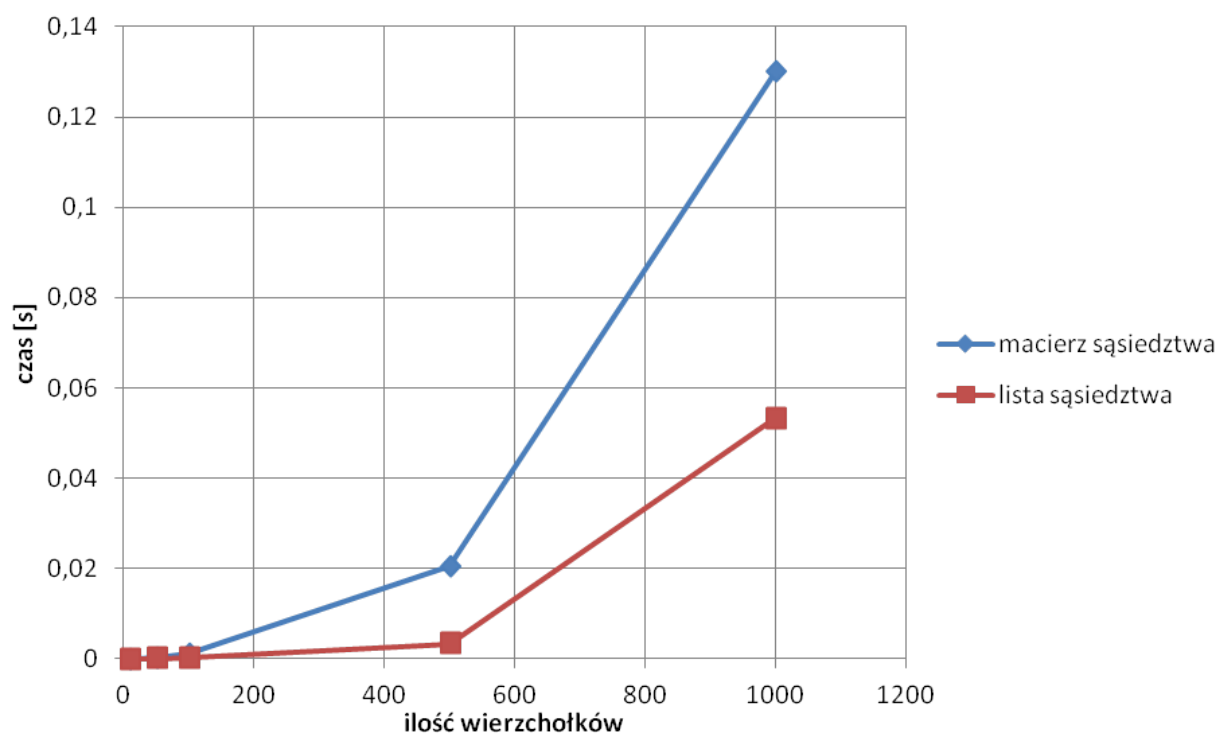
6 Wykresy:



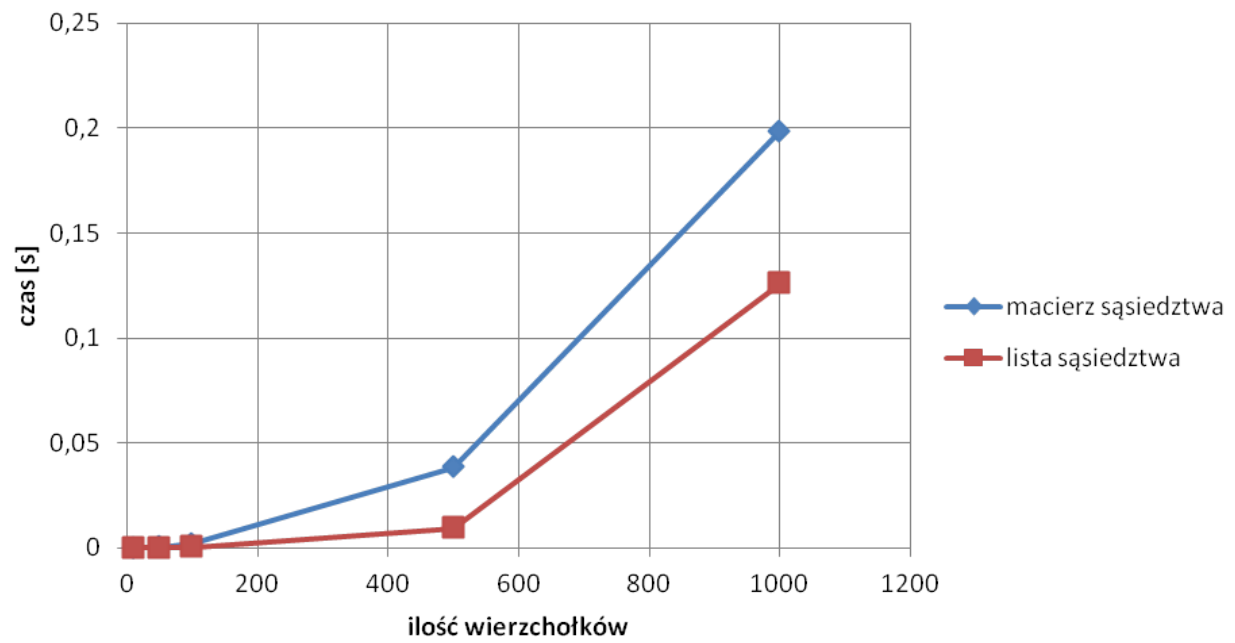
Algorytm Bellmana - Forda dla listy sąsiedztwa



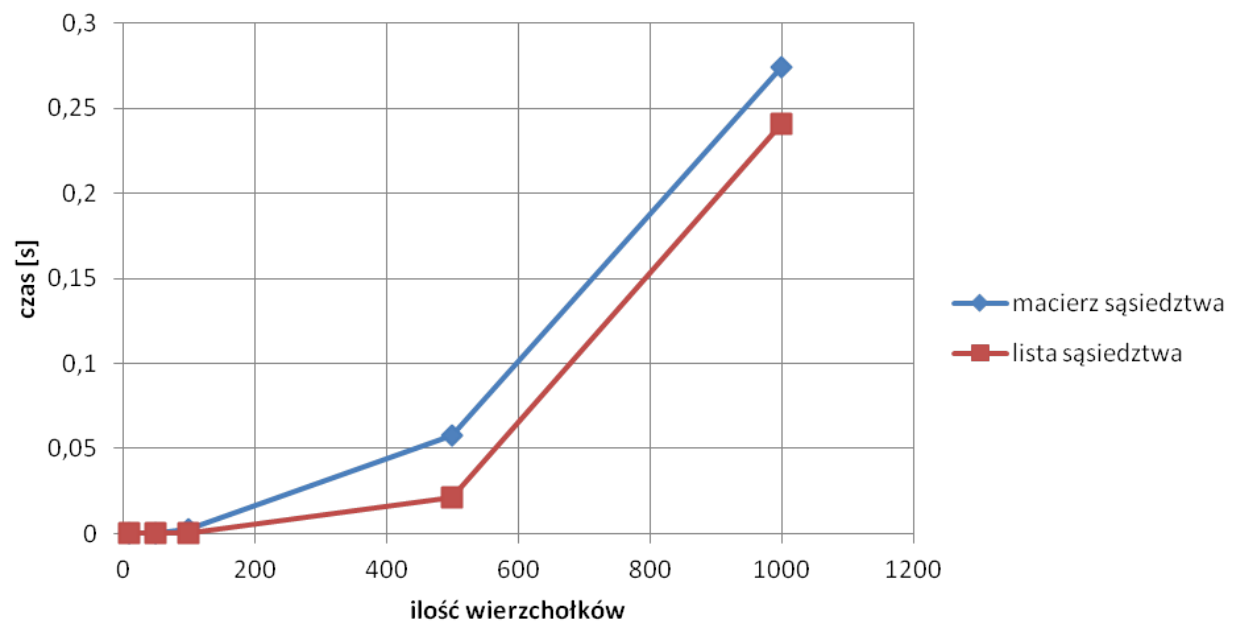
25%



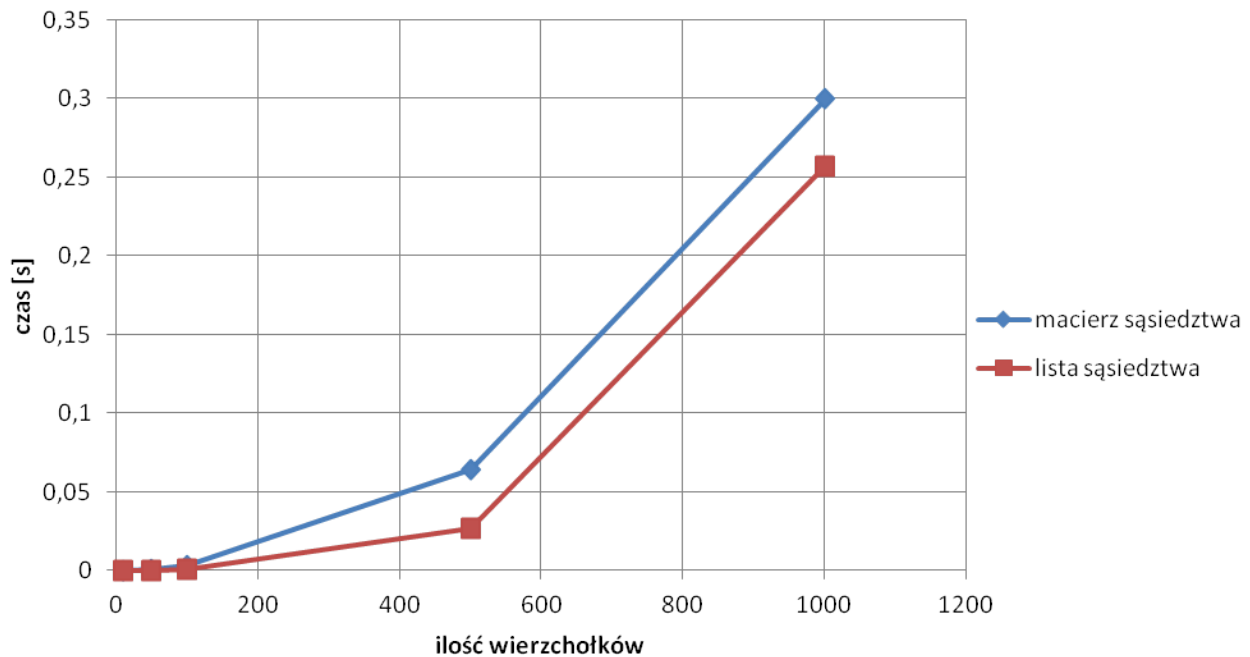
50%



75%



Pełny graf



7 Wnioski:

-Im większa jest gęstość grafu tym dłuższy jest czas trwania algorytmu.

-Implementacja za pomocą listy sąsiedztwa jest szybsza od implementacji za pomocą macierzy sąsiedztwa. Wynika to z faktu, że lista sąsiedztwa ma mniejszą złożoność pamięciową oraz szybsze przeszukiwanie krawędzi wychodzących z danego wierzchołka. Dla listy sąsiedztwa złożoność pamięciowa wynosi $O(E)$ a przejrzenie wszystkich krawędzi $O(E)$. Natomiast dla macierzy sąsiedztwa złożoność pamięciowa wynosi $O(V^2)$ a przejrzenie wszystkich krawędzi $O(V^2)$.

-sprawdzenie czy pojedyncza krawędź istnieje za pomocą macierzy sąsiedztwa jest szybsze, ponieważ w przypadku listy sąsiedztwa trzeba przejrzeć całą listę, a dla macierzy sprawdza się odpowiednie miejsce w tablicy

8 Literatura:

<http://www.algorytm.org/algorytmy-grafowe/algorytm-forda-bellmana.html>

https://pl.wikipedia.org/wiki/Algorytm_Bellmana-Forda

<https://eduinf.waw.pl/inf/alg/001search/0138a.php>

<http://smurf.mimuw.edu.pl/node/375>

https://pl.wikipedia.org/wiki/Reprezentacja_grafu

<https://eduinf.waw.pl/inf/utills/002roz/ol011.php>

<http://www.algorytm.org/klasyczne/grafy-i-ich-reprezentacje.html>

<https://eduinf.waw.pl/inf/alg/001search/0124.php>