

Projektowanie algorytmów i metody sztucznej inteligencji

Projekt 2

Prowadzący: Dr inż. Andrzej Rusiecki

Wtorek 13:15-15:00

Imię Nazwisko: Mateusz Bartkowiak

Numer indeksu: 249467

1. Wykonanie zadania

Projekt polegał na przetestowaniu działania algorytmów wyznaczających najkrótszą drogę z danego wężła grafu do pozostałych węzłów.

Algorytmy użyte w zadaniu:

a) Bellmana-Forda – stosować można przy założeniu, że w grafie nie ma wag ujemnych. Pesemistyczna złożoność obliczeniowa algorytmu Dikstry wynosi $(E+V\log V)$

b) Dijkstry – dopuszczalne są wagi ujemne, natomiast niedopuszczalne jest istnienie cyklu o koszcie ujemnym. Pesemistyczna złożoność obliczeniowa algorytmu wynosi $O(VE)$

Zbadana została efektywność powyższych algorytmów w zależności od gęstości grafu: 25%, 50%, 75% oraz dla grafu pełnego dla różnej ilości węzłów. Dla każdego takiego zestawu wygenerowano 100 prób by móc bardziej sprecyzować wyniki.

Program również umożliwia wczytywanie grafów z plików tekstowych oraz możliwość zapisu wyniku działania algorytmu .

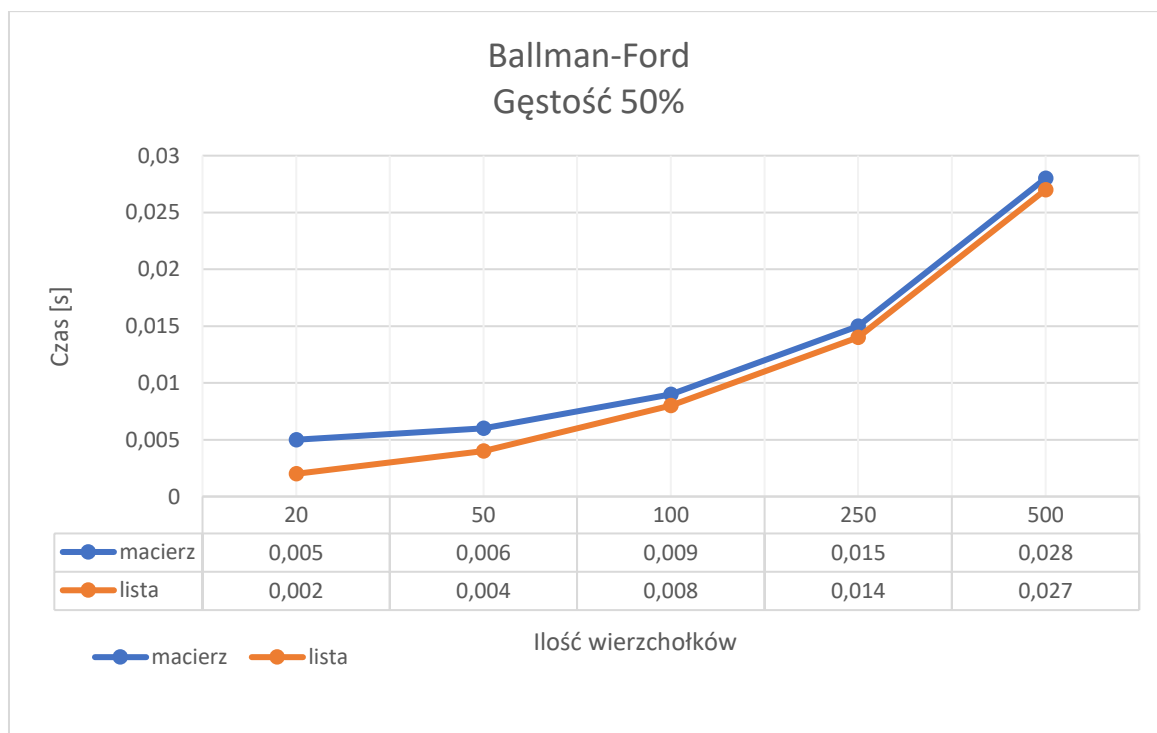
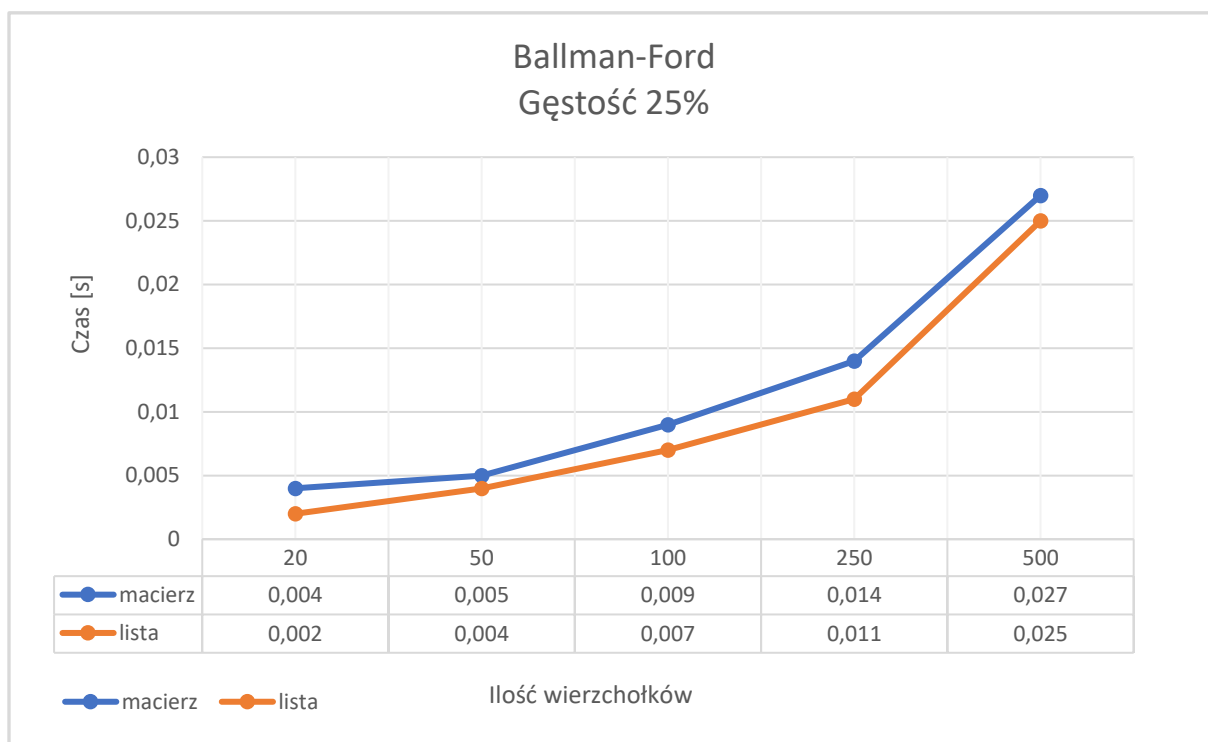
Sprzęt:

Pamięć: 8GB RAM

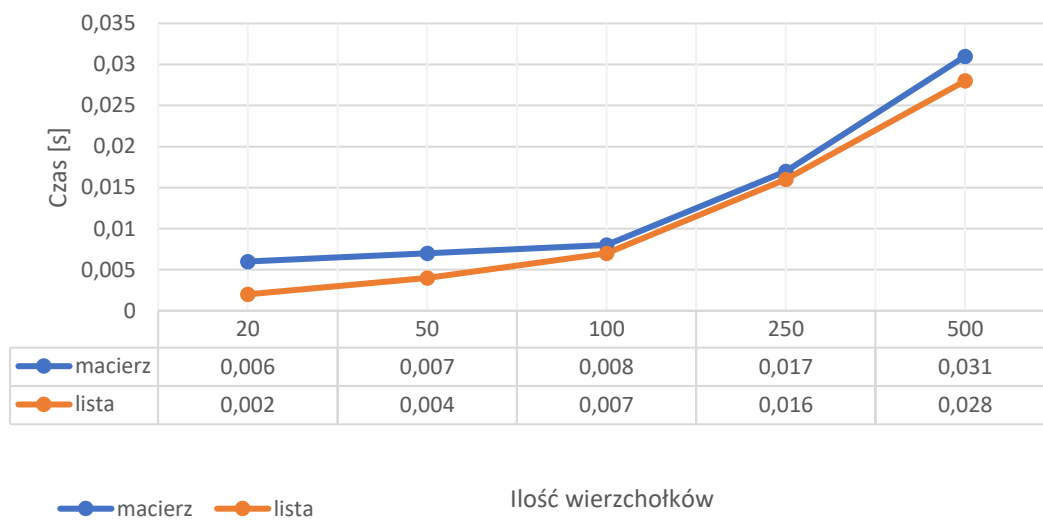
Procesor: Intel Core i5-8250U CPU 1.60Ghz (8 CPUs)

System operacyjny: Windows 10

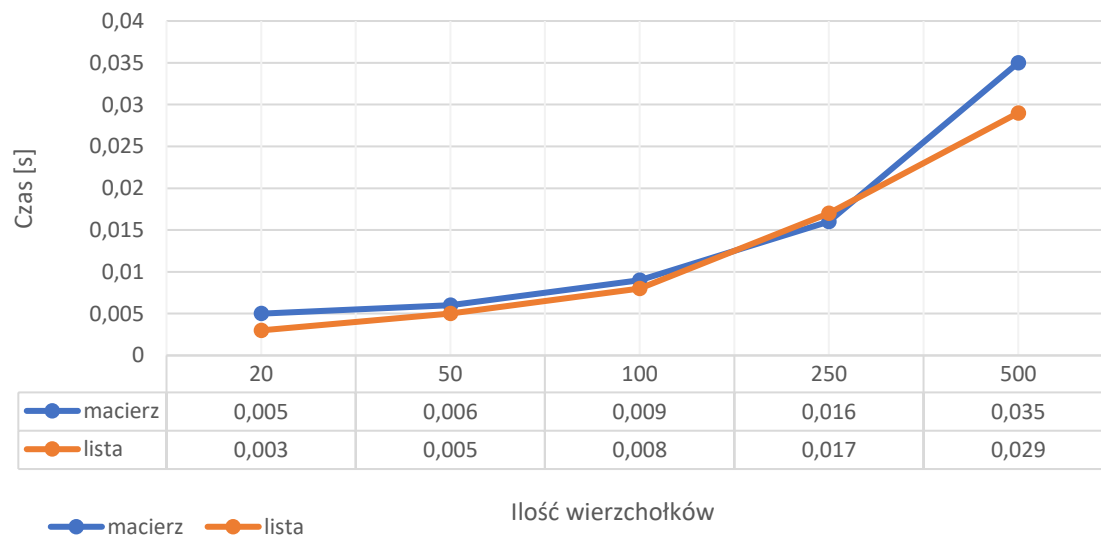
2. Otrzymane wyniki



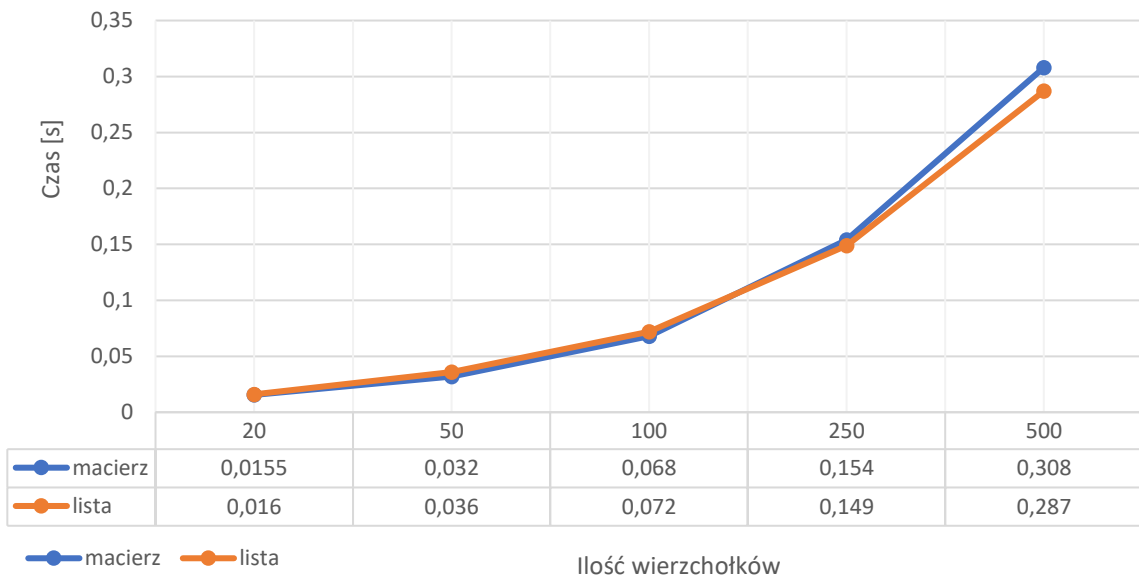
Ballman-Ford Gęstość 75%



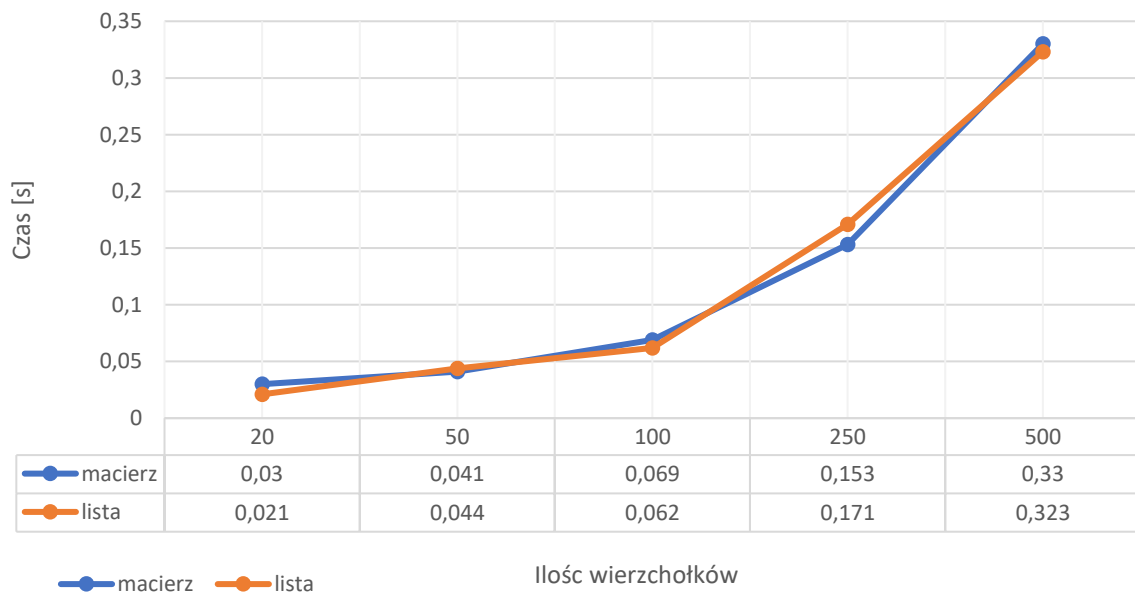
Ballman-Ford Gęstość 100%



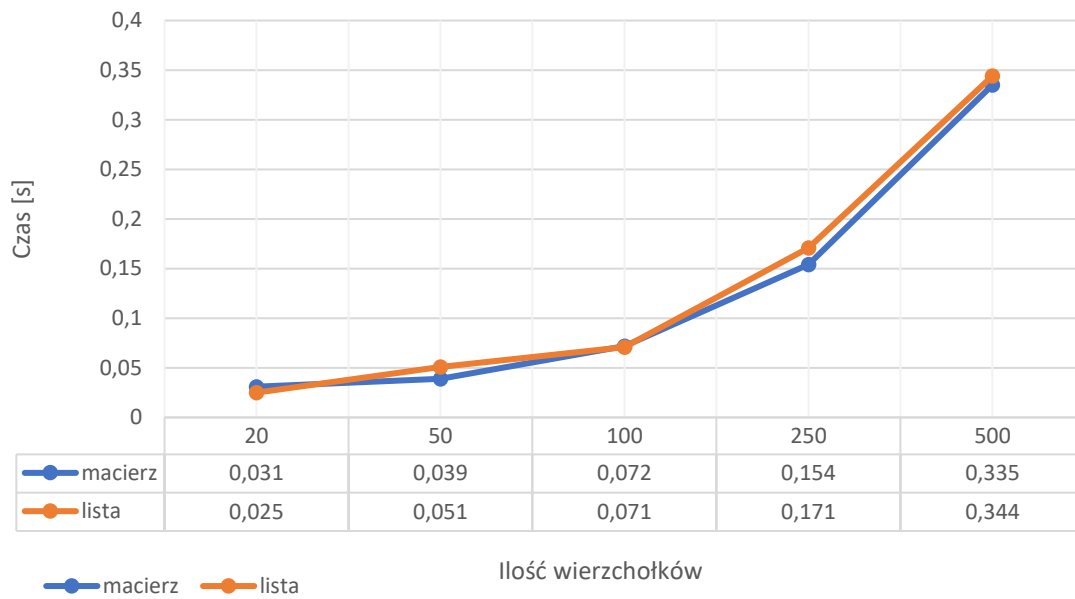
Dijkstry Gęstość 25%



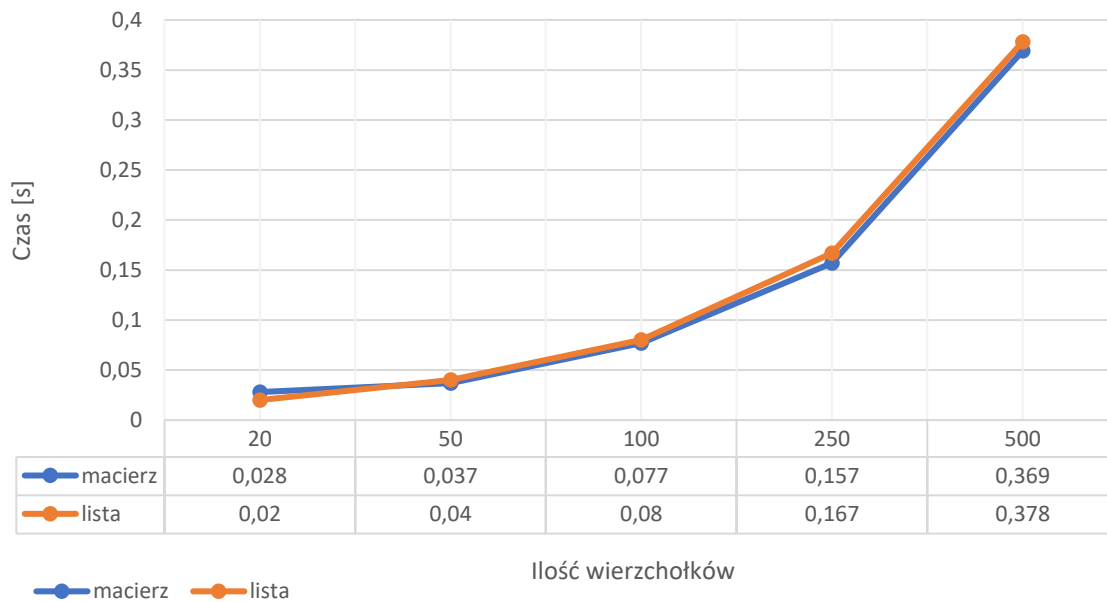
Dijkstry Gęstość 50%



Dijkstry Gęstość 75%



Dijkstry Gęstość 100%



3. Wnioski

Z wykresów wynika, że lista w większości przypadkach jest szybsza niż macierz. Może to być spowodowane tym, że przeszukiwanie całej tablicy jest wolniejsze niż dostęp do danych w zaimplementowanej liście. Kolejnym powodem może być fakt, że macierz potrzebuje więcej pamięci niż lista.