

13.05.2019r.

# **Projektowanie Algorytmów i Metody Sztucznej Inteligencji**

## **Projekt 2 – Implementacja Grafów**

Jacek Kopala

## 1. Wstęp

Tematem projektu jest implementacja i przetestowanie algorytmu najkrótszej drogi – algorytmu Dijkstry. Implementacja grafu została wykonana przy pomocy listy sąsiedztwa oraz macierzy sąsiedztwa .

Samo testowanie przebiegło na grafach ważonych, kierunkowych, w którym nie ma węzłów odizolowanych (do każdego węzła prowadzi chociaż jedna krawędź), a wagi krawędzi nie przyjmowały wartości ujemnych. Rozłożenie krawędzi oraz wagi, z zachowaniem powyższych założeń, są losowe.

## 2. Algorytm Dijkstry

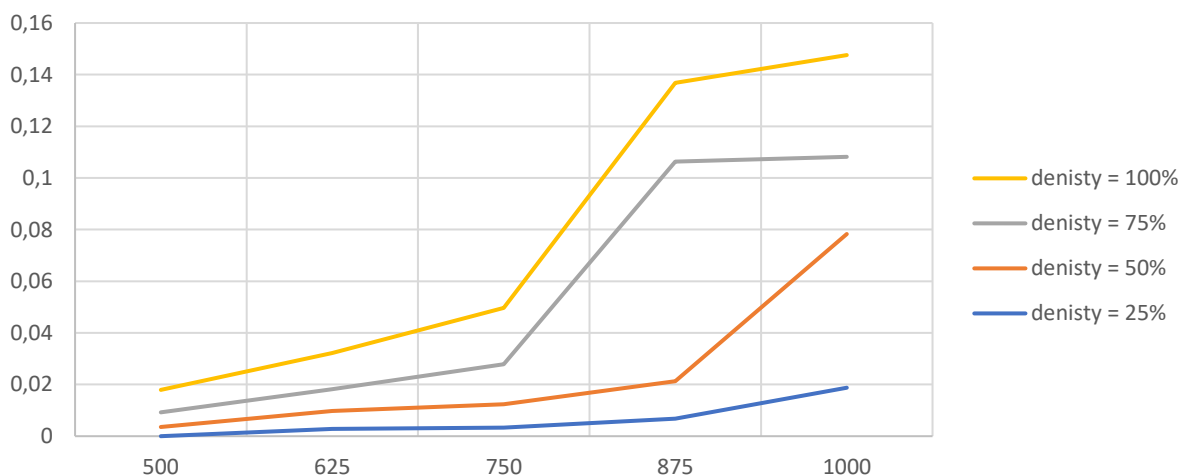
Celem realizacji funkcjonalności znajdowania najkrótszej ścieżki w grafie pomiędzy zadanim wierzchołkiem a resztą wierzchołków występujących w grafie, został wykorzystany algorytm Dijkstry, należący do rodziny algorytmów zachłannych. W przypadku implementacji grafu w postaci listy sąsiedztwa oraz algorytmu opartego o kolejkę priorytetową bazowaną na kopcu złożoność obliczeniowa algorytmu sprowadza się do postaci  $O((n + m) \log n)$ .

## 3. Pomiary

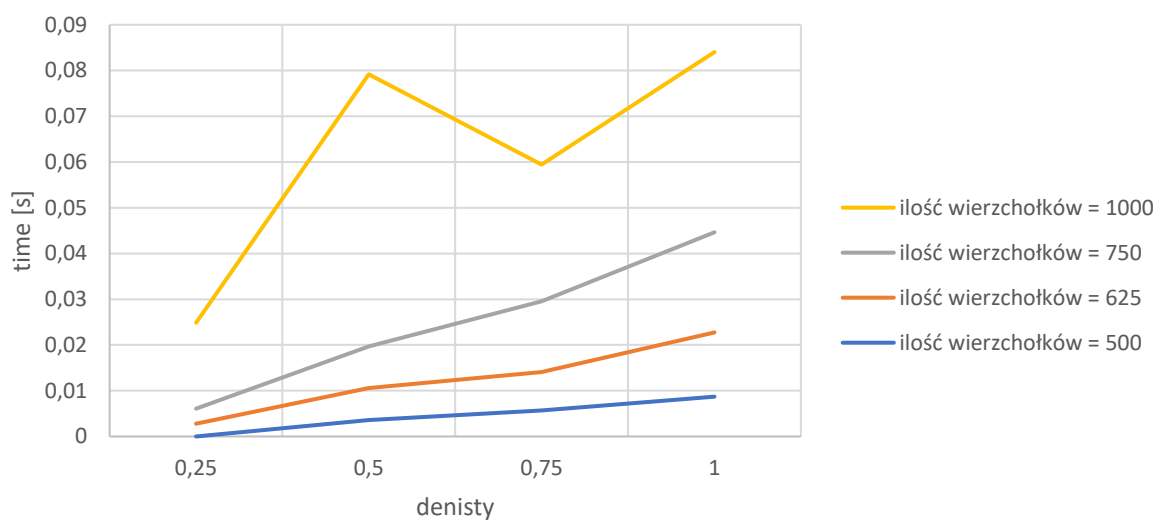
IMPLEMENTACJA ZA POMOCĄ LISTY				
V	denisty	min	max	śr
500	0,25	0	0,016	0
	0,5	0	0,016	0,00357
	0,75	0	0,016	0,00567
	1	0	0,016	0,0087
625	0,25	0	0,018	0,00279
	0,5	0	0,016	0,007
	0,75	0	0,016	0,00843
	1	0	0,025	0,01404
750	0,25	0	0,02	0,00329
	0,5	0	0,023	0,0091
	0,75	0	0,016	0,01546

	1	0,015	0,032	0,02189
825	0,25	0	0,017	0,0068
	0,5	0	0,017	0,01449
	0,75	0,015	1,438	0,0851
	1	0,015	0,046	0,03043
1000	0,25	0	0,236	0,01877
	0,5	0,015	0,644	0,05952
	0,75	0,015	0,037	0,02989
	1	0,031	0,053	0,03941

Czas wykonywania się algorytmu w zależności od ilości wierzchołków  
(lista sąsiedztwa)

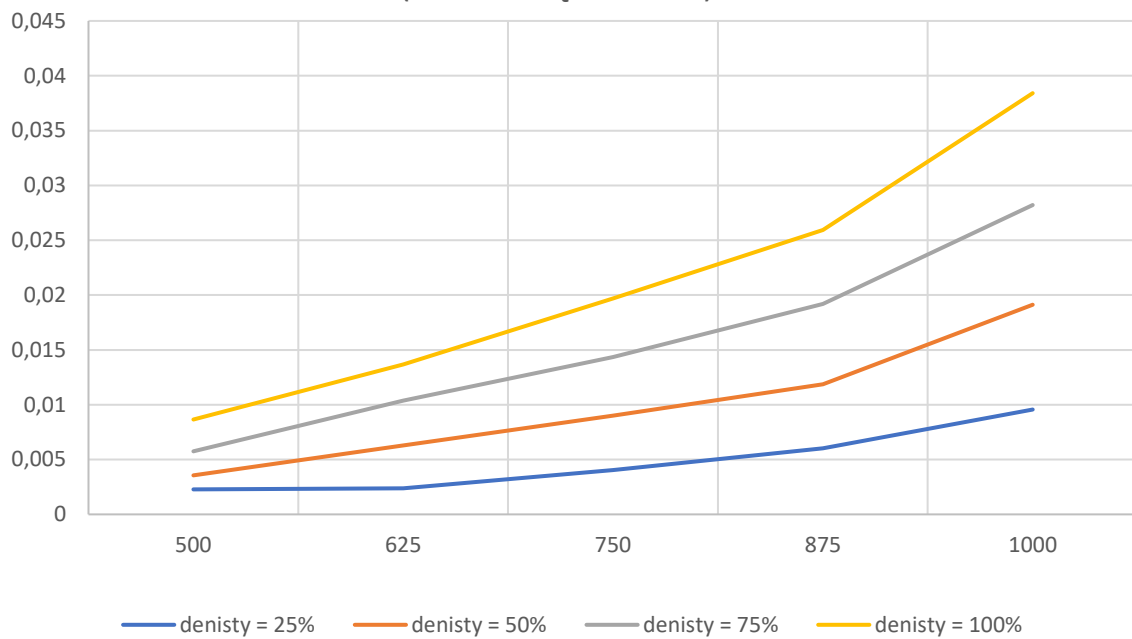


Czas wykonywania się algorytmu w zależności od gęstości grafu  
(lista sąsiedztwa)

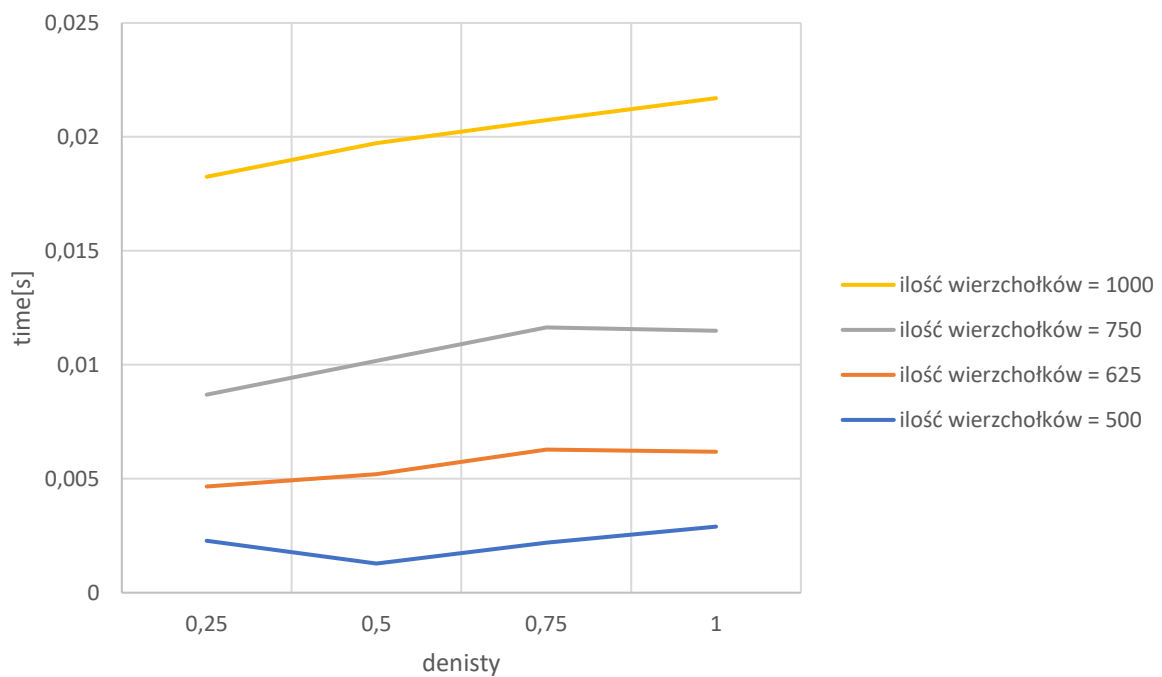


IMPLEMENTACJA GRAFU ZA POMCĄ MACIERZY				
V	denisty	min	max	śr
500	0,25	0	0,016	0,00228
	0,5	0	0,016	0,00128
	0,75	0	0,016	0,00219
	1	0	0,016	0,0029
625	0,25	0	0,016	0,00238
	0,5	0	0,016	0,00392
	0,75	0	0,017	0,00409
	1	0	0,016	0,00328
750	0,25	0	0,016	0,00403
	0,5	0	0,016	0,00497
	0,75	0	0,016	0,00536
	1	0	0,016	0,00532
825	0,25	0	0,016	0,00604
	0,5	0	0,019	0,00581
	0,75	0	0,016	0,00733
	1	0	0,018	0,00675
1000	0,25	0	0,023	0,00956
	0,5	0	0,016	0,00956
	0,75	0	0,019	0,00909
	1	0	0,022	0,0102

Czas wykonywania się algorytmu w zależności od ilości wierzchołków  
(macierz sąsiedztwa)



Czas wykonywania się algorytmu w zależności od gęstości grafu  
(macierz sąsiedztwa)



#### 4. Wnioski

Zgodnie z oczekiwaniami, długość wykonywania się algorytmu rosła wraz ze wzrostem wielkości grafu oraz jego gęstości. Zależy to bezpośrednio od ilości połączeń między wierzchołkami grafu. Pewnie odchylenia od normy (w niektórych przypadkach średnia dla większej gęstości jest większa niż ta dla mniejszej) mogą wynikać z tego, że w części przypadków algorytm wykonywał się tak szybko, że program testujący notował czas jako 0, pewne subtelne różnice w czasie nie zostały więc zarejestrowane. Przy większej próbie badawczej prawdopodobnie błąd by zniknął.

#### 5. Literatura

<https://www.geeksforgeeks.org/dijkstras-algorithm-for-adjacency-list-representation-greedy-algo-8/> [dostęp 10.05.2019]

[https://pl.wikipedia.org/wiki/Algorytm\\_Dijkstry](https://pl.wikipedia.org/wiki/Algorytm_Dijkstry) [dostęp 9.05.2019]

Data Structures & Algorithms M.T.Goodrich R.Tamassia D.Mount Second Edition