

# Projektowanie algorytmów i metody sztucznej inteligencji

## Projekt 3 - Grafy

Adam Kubiak 249480

5 maja 2020

### 1. Wstęp

Tematem projektu jest zbadanie efektywności działania algorytmu znajdowania drogi o najmniejszym koszcie w grafie spójnym, o różnej gęstości oraz reprezentacji. W programie zostały wykorzystane dwie reprezentacje grafu: na podstawie macierzy sąsiedztwa oraz listy sąsiedztwa. Użyty przez mnie algorytmem był algorytm Dijkstry. W projekcie przyjąłem założenie iż waga krawędzi znajduje się w przedziale od 1 do 10.

### 2. Reprezentacja w postaci macierzy sąsiedztwa

Reprezentacja grafu w postaci macierzy sąsiedztwa polega na stworzeniu macierzy  $n \times n$ , gdzie  $n$  to ilość wierzchołków grafu. W odpowiednich miejscach macierzy umieszczone zostają krawędzie od wierzchołka  $x$  do wierzchołka  $y$ . Zaletą tej implementacji jest natychmiastowy dostęp do krawędzi między dwoma danymi wierzchołkami. Wadą natomiast jest ilość potrzebnej pamięci, ponieważ niezależnie od ilości krawędzi, potrzebujemy takiej samej macierzy dla danej ilości wierzchołków.

### 3. Reprezentacja w postaci listy sąsiedztwa

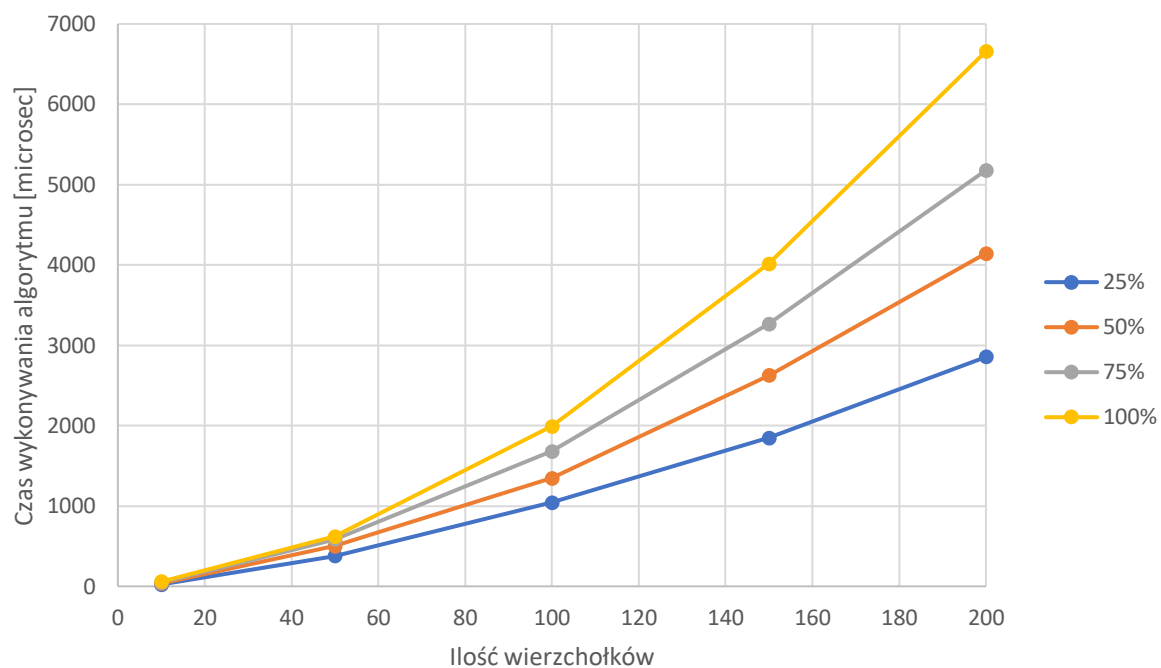
Reprezentacja grafu w postaci listy sąsiedztwa polega na utworzeniu listy list. Implementacja ta ma sporą złożoność obliczeniową w momencie gdy potrzebujemy uzyskać dostęp do krawędzi znajdujących się w jednym z środkowych wierzchołków, ponieważ musimy przeiterować przez maksymalnie połowę listy. Zaletą implementacji grafu w postaci listy sąsiedztwa jest mniejsza ilość potrzebnego miejsca, ponieważ potrzebujemy go tyle, ile jest krawędzi w grafie. Wadą tej implementacji jest potrzeba iterowania w liście krawędzi dla danego wierzchołka, w celu znalezienia interesującej nas krawędzi.

#### 4. Wyniki pomiarów

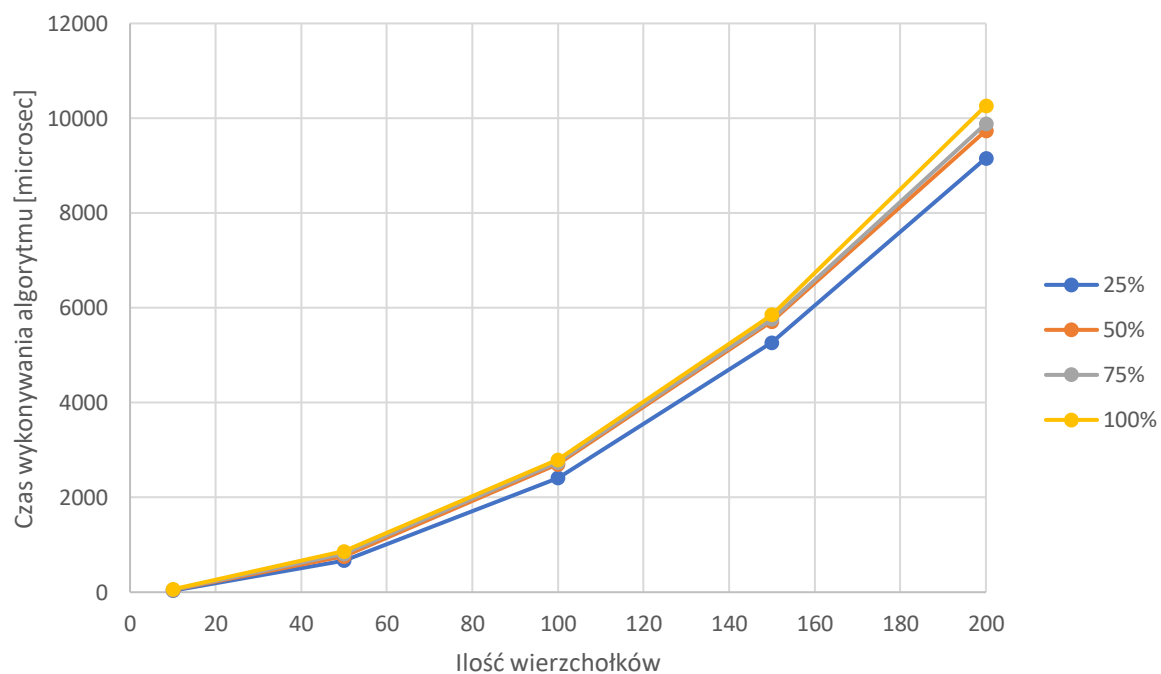
Lista sąsiedztwa				
Ilość wierzchołków	Gęstość			
	0,25	0,5	0,75	1
	Czas wykonywania algorytmu [microsec]			
10	24	34,9	47,5	58,1
50	377,6	501,7	584	622,2
100	1041,8	1345,3	1678,6	1991,8
150	1849,6	2624,2	3266,3	4015,6
200	2854,4	4144,9	5179,3	6658,8

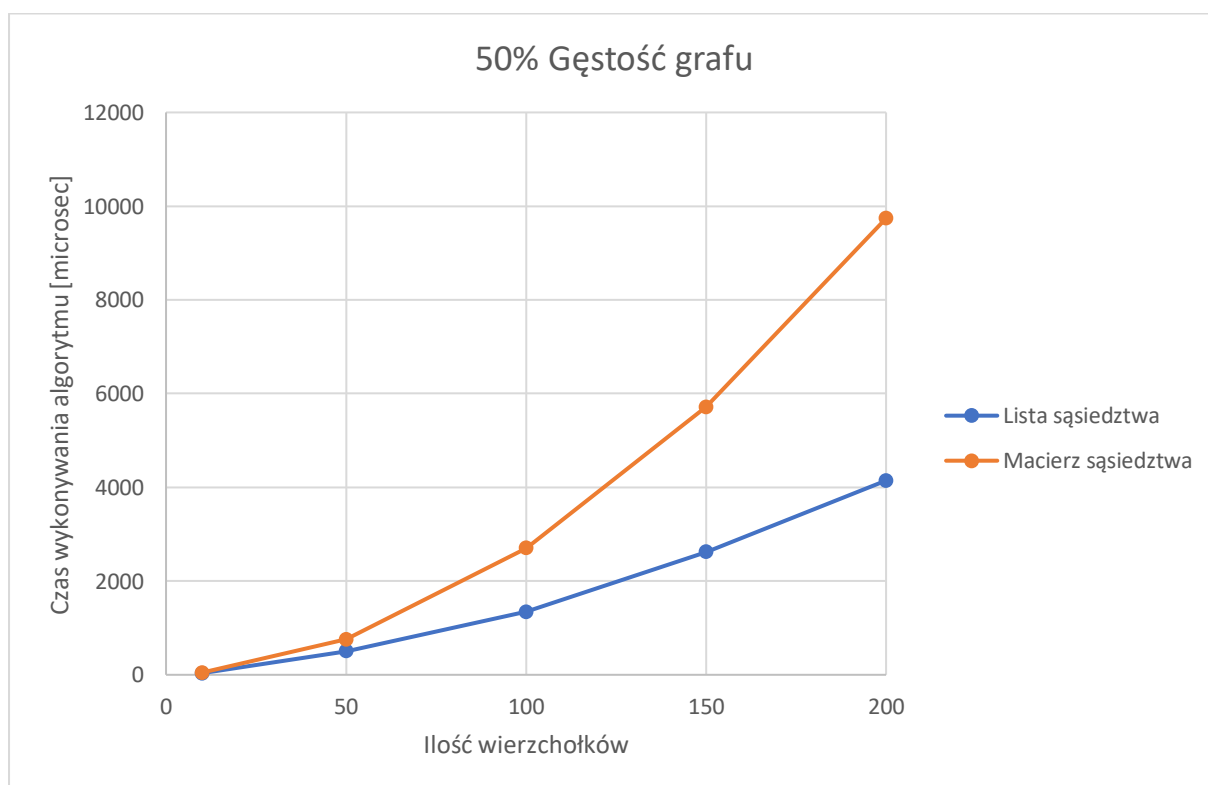
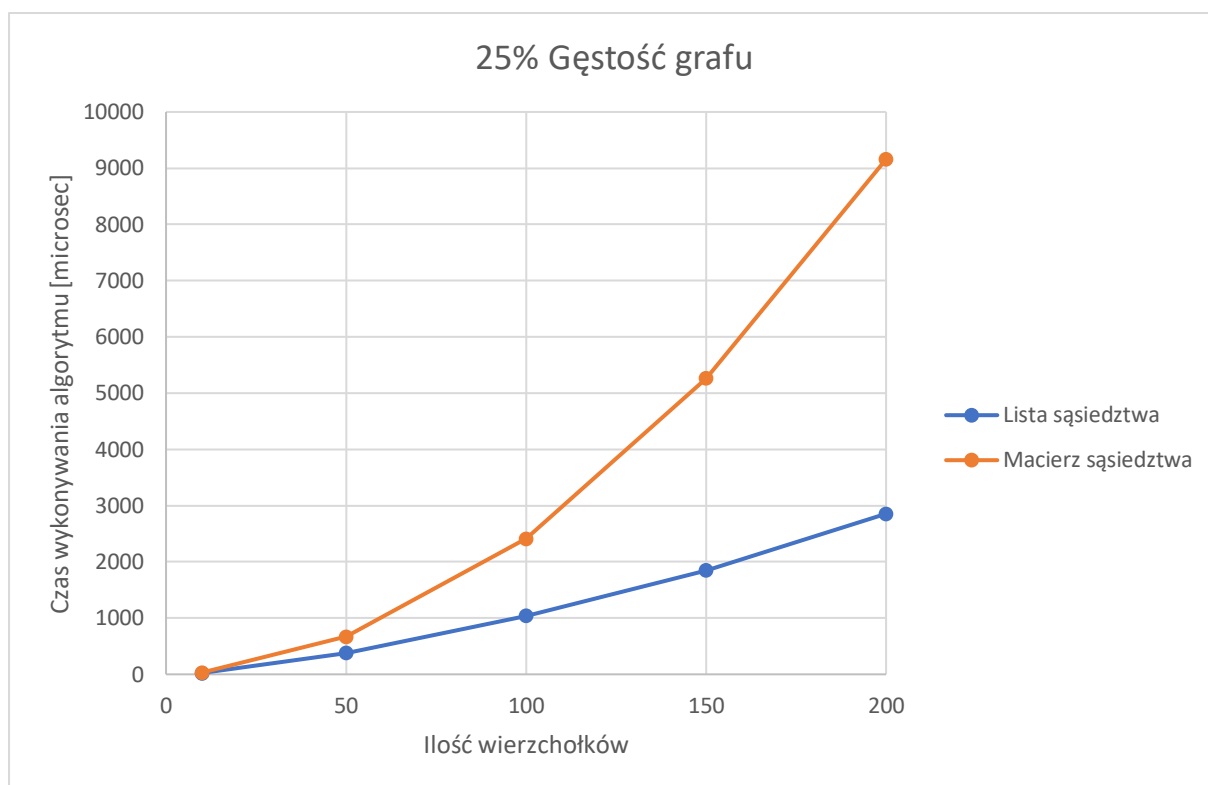
Macierz sąsiedztwa				
Ilość wierzchołków	Gęstość			
	0,25	0,5	0,75	1
	Czas wykonywania algorytmu [microsec]			
10	31,8	47,5	51,3	61,3
50	668,2	757,7	813	865
100	2414,1	2704,8	2740,7	2800,1
150	5266,1	5713,5	5762,4	5856,9
200	9158,9	9739,9	9887,6	10267

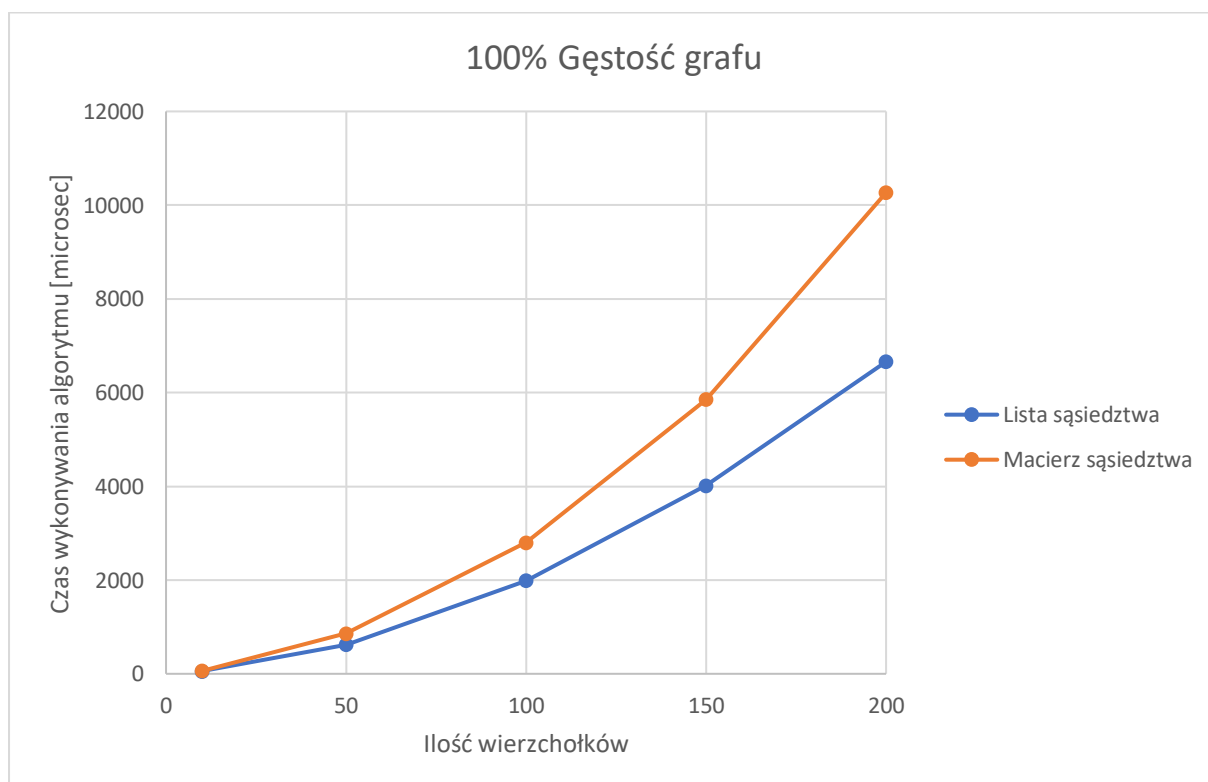
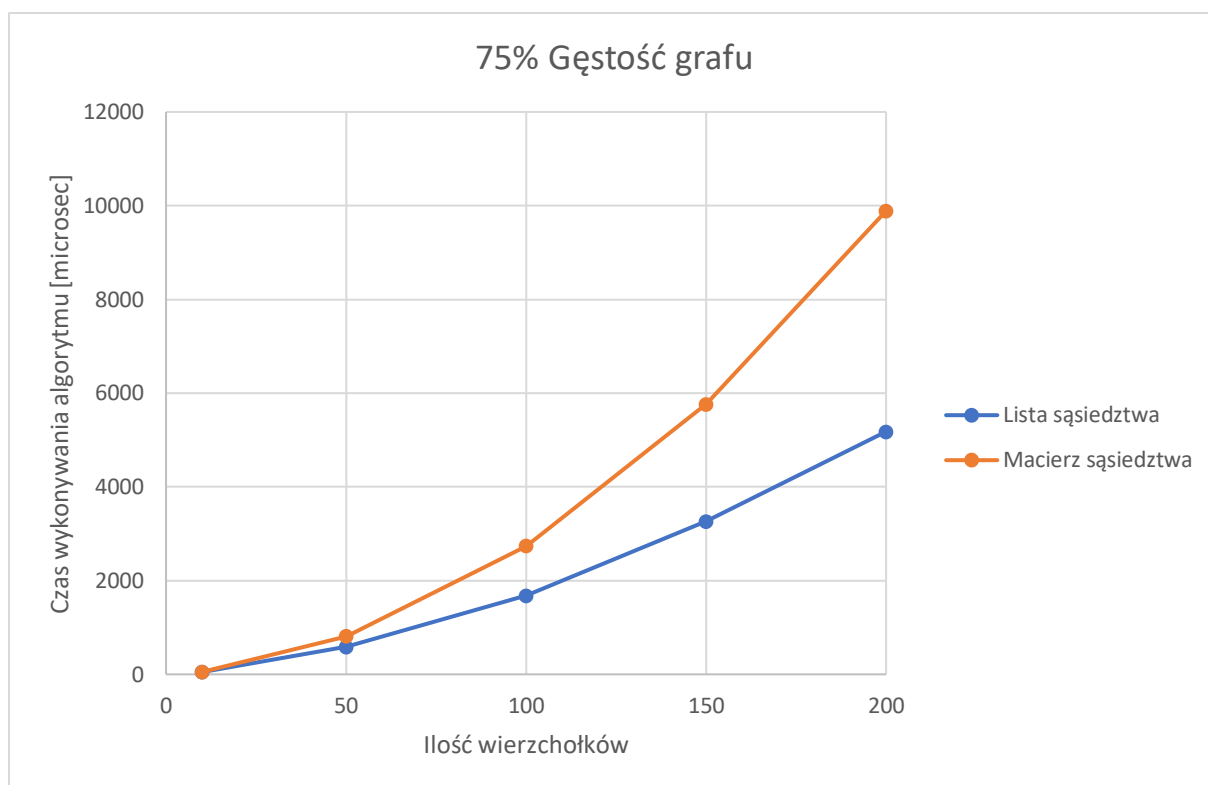
Reprezentacja grafu poprzez listę sąsiedztwa



Reprezentacja grafu poprzez macierz sąsiedztwa







## 5. Wnioski

Z otrzymanych wyników można odczytać, iż zgodnie z założeniami ilość krawędzi nie wpłynęła w zbyt dużym stopniu na czas wykonywania algorytmu w przypadku reprezentacji za pomocą macierzy sąsiedztwa. Na wykresie „Reprezentacja grafu poprzez macierz sąsiedztwa” widać iż złożoność obliczeniowa w przypadku tego algorytmu wynosi  $O(V)$ , gdzie  $V$  to ilość wierzchołków, zgodnie z założeniami dla tej reprezentacji. Wraz ze wzrostem ilości wierzchołków czas wykonywania algorytmu dla grafy o reprezentacji w postaci listy sąsiedztwa wydłuża się stosunkowo więcej niż dla reprezentacji w postaci macierzy sąsiedztwa. Zgodnie z założeniami zwiększenie gęstości grafu, czyli zwiększenie ilości krawędzi miało bardzo duży wpływ na czas wykonywania algorytmu. Implementacja na podstawie listy sąsiedztwa jest mniej optymalna pod względem złożoności obliczeniowej, lecz jest lepsza pod względem ilości potrzebnej pamięci. Są to dwa różne kryteria, pod względem których powinno wybrać się odpowiednią implementację.