

# Projektowanie algorytmów i metody sztucznej inteligencji

## Projekt 3 - Grafy

Anna Bernaś 241613

8 maja 2019

### 1 Wstęp

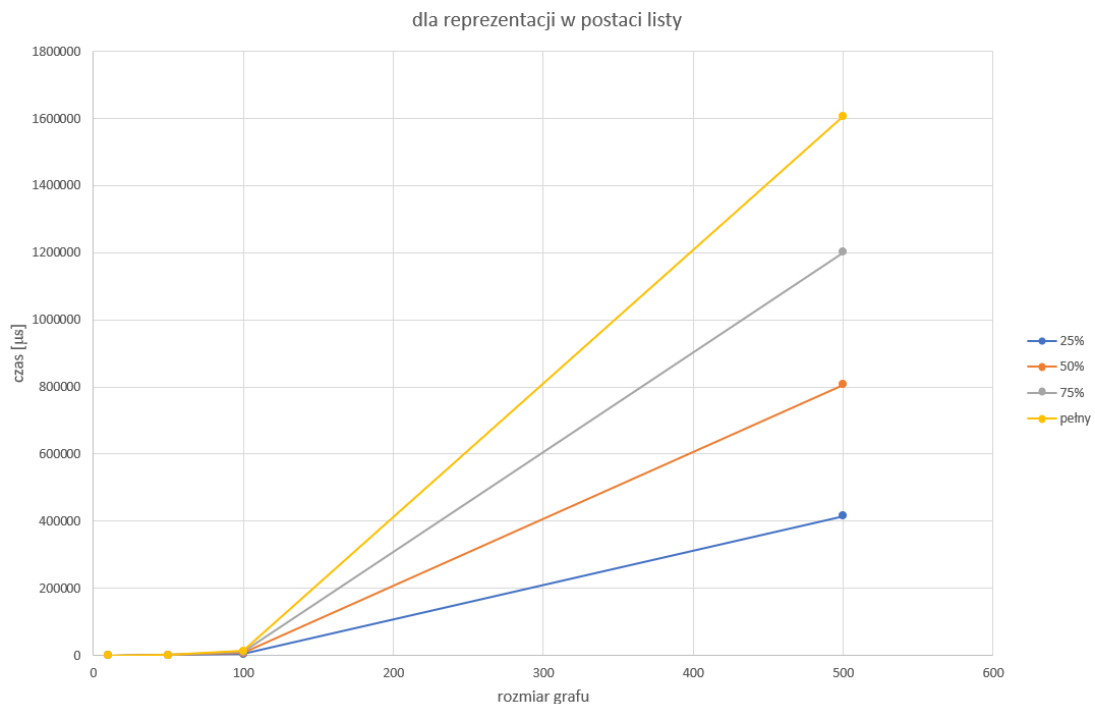
Zbadano efektywność algorytmów : Bellmana-Forda oraz Dijkstry dla dwóch reprezentacji grafów: macierzy sąsiedztwa oraz listy sąsiedztwa. Badano je dla rozmiarów grafów: 10, 50, 100, 500 oraz dla gęstości grafów 25%, 50%, 75% oraz dla grafu pełnego. Czasy trwania każdego z algorytmów są podawane w mikrosekundach.

### 2 Algorytm Bellmana-Forda

Złożoność czasowa tego algorytmu wynosi  $O(V \cdot E)$ .

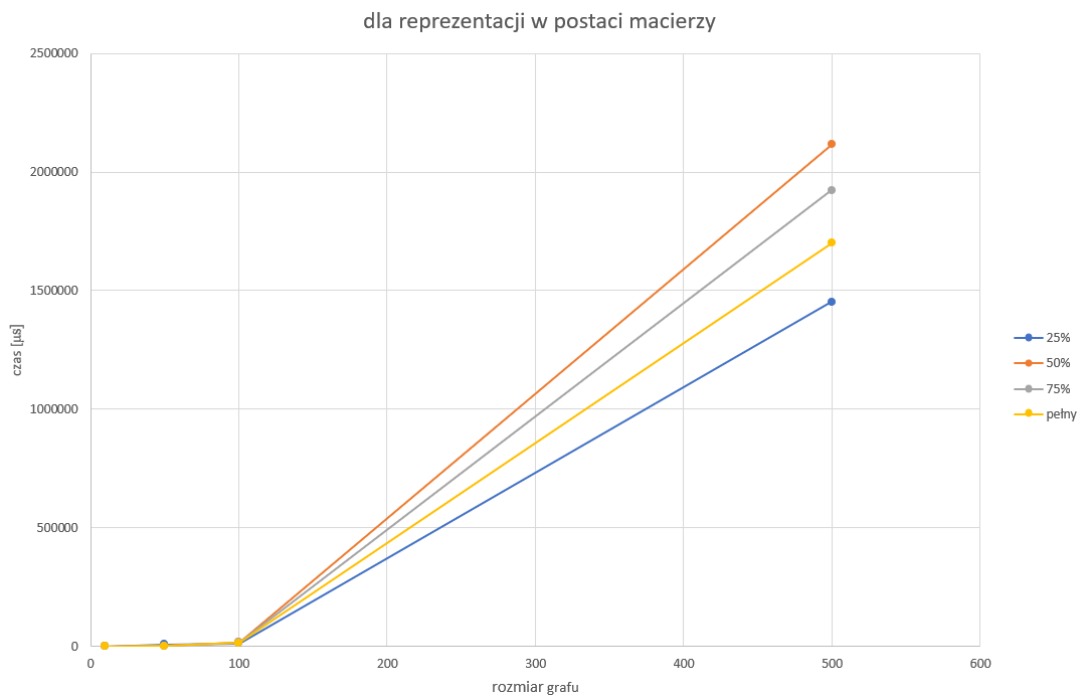
#### 2.1 Czas trwania algorytmu w zależności od gęstości i rozmiaru grafu w postaci listy sąsiedztwa

	25%	50%	75%	pełny
10	32	10	13	15
50	805	880	1263	1643
100	4139	6801	9853	12974
500	414715	806735	1202076	1606449



## 2.2 Czas trwania algorytmu w zależności od gęstości i rozmiaru grafu w postaci macierzy sąsiedztwa

	25%	50%	75%	pełny
10	39	11	13	13
50	7140	1931	1878	1682
100	12165	16792	15184	13695
500	1451822	2115671	1923930	1699645

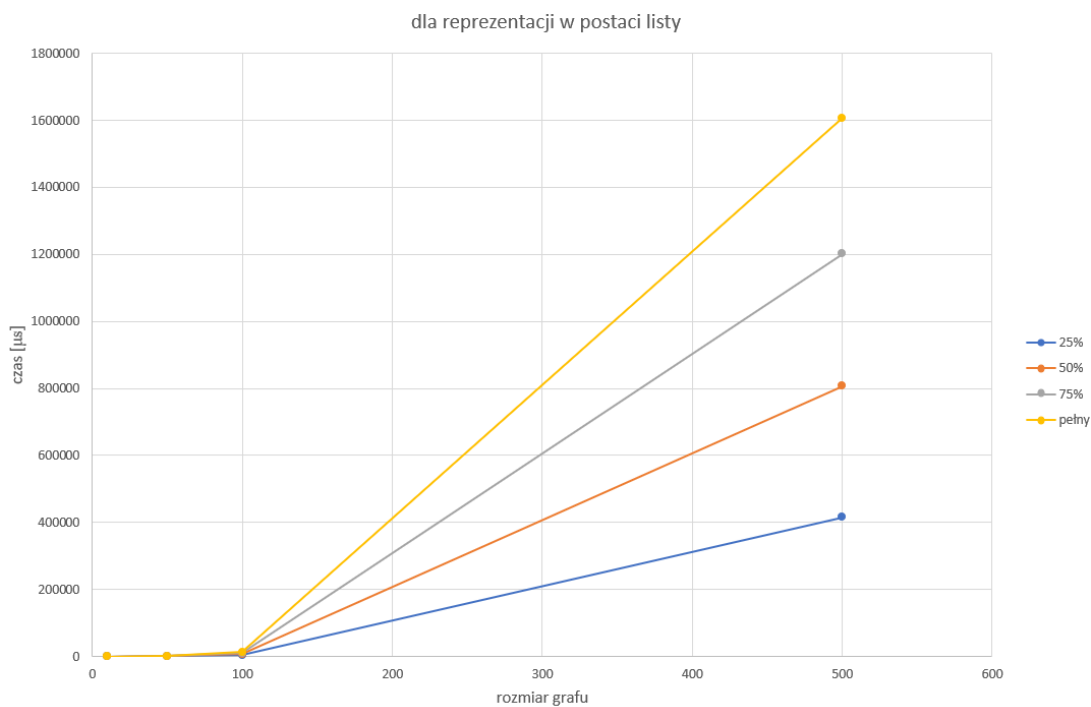


## 3 Algorytm Dijkstry

Złożoność czasowa tego algorytmu wynosi  $O(E \cdot \log(V))$ .

### 3.1 Czas trwania algorytmu w zależności od gęstości i rozmiaru grafu w postaci listy sąsiedztwa

	25%	50%	75%	pełny
10	22	9	12	11
50	158	129	170	215
100	409	460	595	758
500	4817	8310	11582	14989



### 3.2 Czas trwania algorytmu w zależności od gęstości i rozmiaru grafu w postaci macierzy sąsiedztwa

	25%	50%	75%	pełny
10	20	9	11	8
50	446	203	198	193
100	486	723	774	696
500	11097	17301	16083	14087



## 4 Wnioski

- Algorytm Dijkstry ( $O(E \cdot \log(V))$ ) jest szybszy od algorytmu Bellmana-Forda ( $O(V \cdot E)$ ).
- Dla grafu pełnego w postaci macierzy sąsiedztwa złożoność czasowa jest porównywalna dla reprezentacji w postaci listy, jednakże dla mniejszych gęstości jest ona większa.