

Sprawozdanie

1. Omówienie wykorzystanych algorytmów i struktur danych

- Algorytm Dijkstry
Tworzę zbiór Q nieodwiedzonych wierzchołków, następnie wybieram wierzchołek początkowy i usuwam go ze zbioru Q, po czym sprawdzam sąsiadów jego w kolejności od najbliższych i usuwam ich ze zbioru, następnie drogi od wierzchołka początkowego zapisuje w tablicy D[i], po czym sprawdzam sąsiadów ostatnio odwiedzonego wierzchołka, czynność poprawiam aż do znalezienia wierzchołka docelowego.
- Algorytm Bellmana-forda
W każdym obiegu pętli sprawdzamy wszystkie krawędzie i wypełniamy tablicę D[i], gdzie znajduje się droga od wierzchołka początkowego do wszystkich pozostałych, pętle powtarzamy N-1 razy, gdzie N – liczba wierzchołków
- Macierz sąsiedztwa
Budujemy tablice o rozmiarach V na V, gdzie V – liczba wierzchołków, następnie wypełniamy je zerami, jeśli dwa wierzchołki są połączone to umieszczamy jedynkę (w moich algorytmach umieszczam wagę krawędzi łączącej dane wierzchołki)
- Lista sąsiadów
Dla każdego wierzchołka trzeba utworzyć listę wierzchołków, które są połączone krawędzią z danym wierzchołkiem

2. Tabele pomiarowe

- Algorytm Dijkstry
V – l. wierzchołków
g – gęstość
t - czas

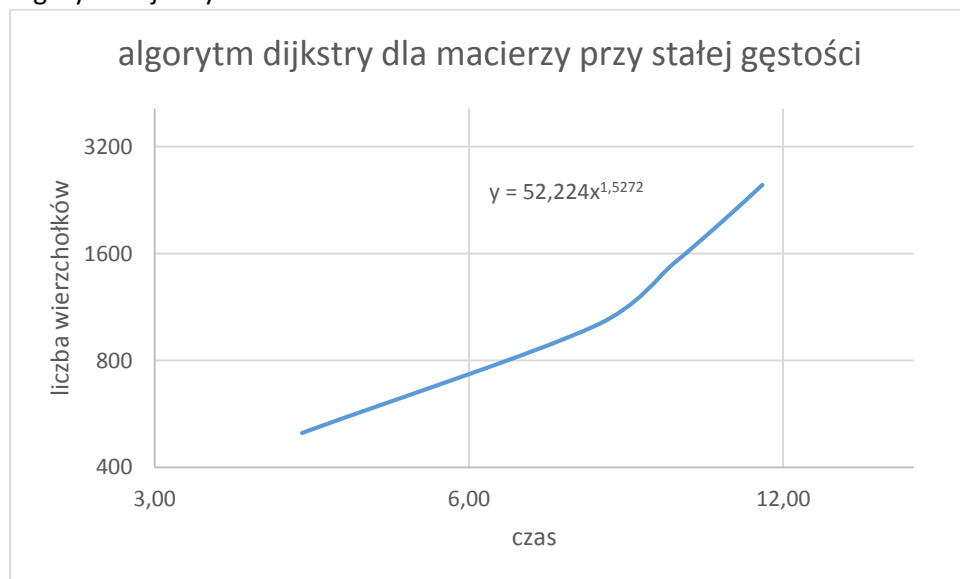
macierz			macierz			lista			lista		
V	g	t	V	g	t	V	g	t	V	g	t
500	50	4,15	1000	20	4,64	500	50	5,96	1000	20	4,67
1000		7,92		40	5,24	1000		8,61		40	5,30
1500		9,42		60	5,77	1500		10,23		60	5,58
2000		10,55		80	6,04	2000		12,66		80	5,98
2500		11,46		100	6,19	2500		13,68		100	6,32

- Algorytm Bellmana-forda
V – l. wierzchołków
g – gęstość
t - czas

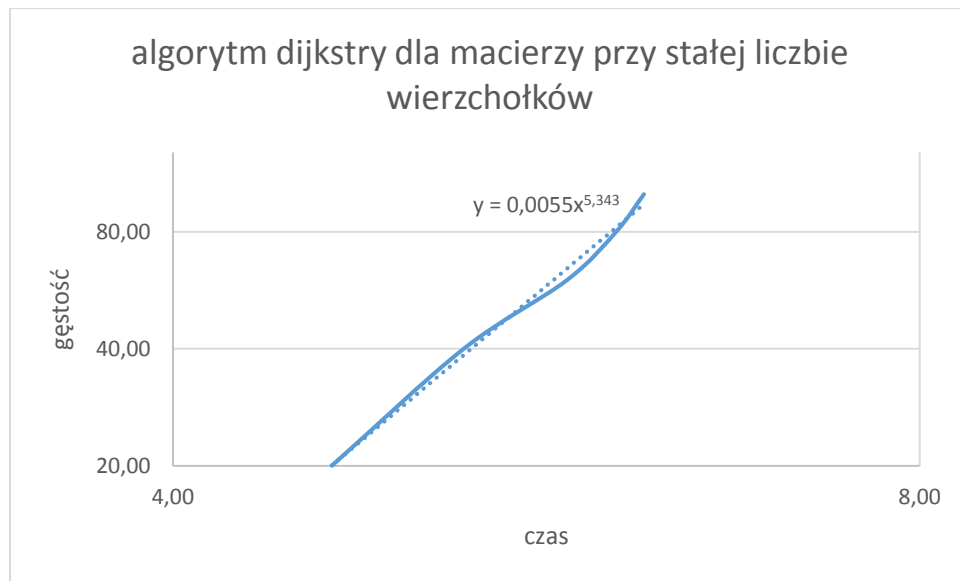
macierz			macierz			lista			lista		
V	g	t	V	g	t	V	g	t	V	g	t
500	50	7,85	1000	20	8,56	500	50	5,09	1000	20	7,61
1000		15,76		40	9,35	1000		13,13		40	9,22
1500		20,17		60	9,98	1500		18,54		60	9,81
2000		23,82		80	10,41	2000		20,62		80	10,32
2500		25,42		100	10,87	2500		22,90		100	10,90

3. Wykresy

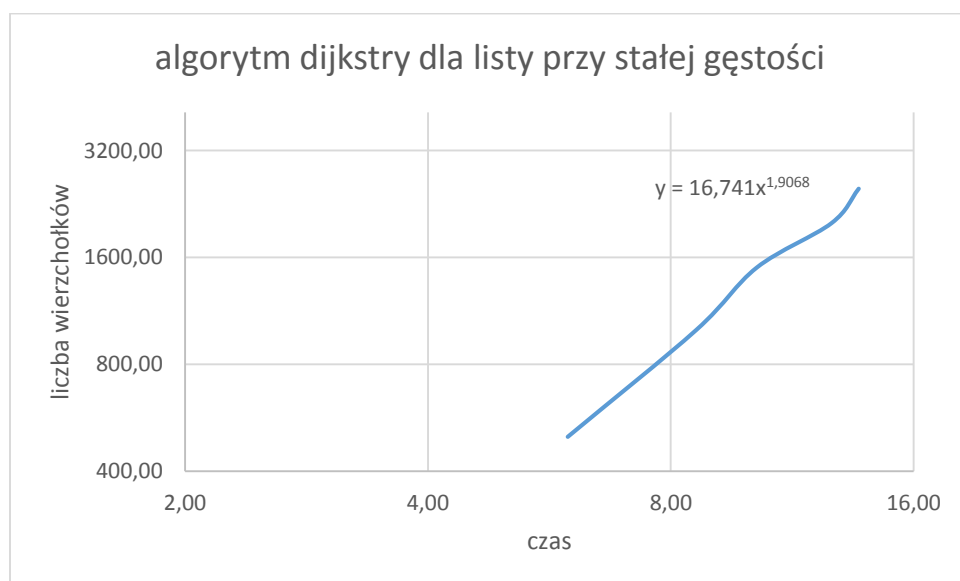
- Algorytm Dijkstry



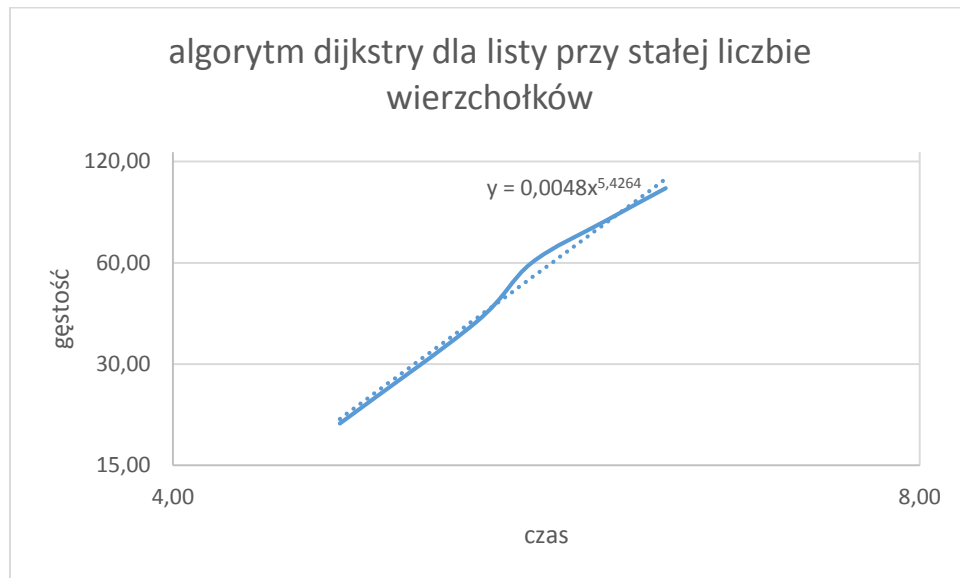
Złożoność obliczeniowa: $O(n^{1,5})$



Złożoność obliczeniowa:

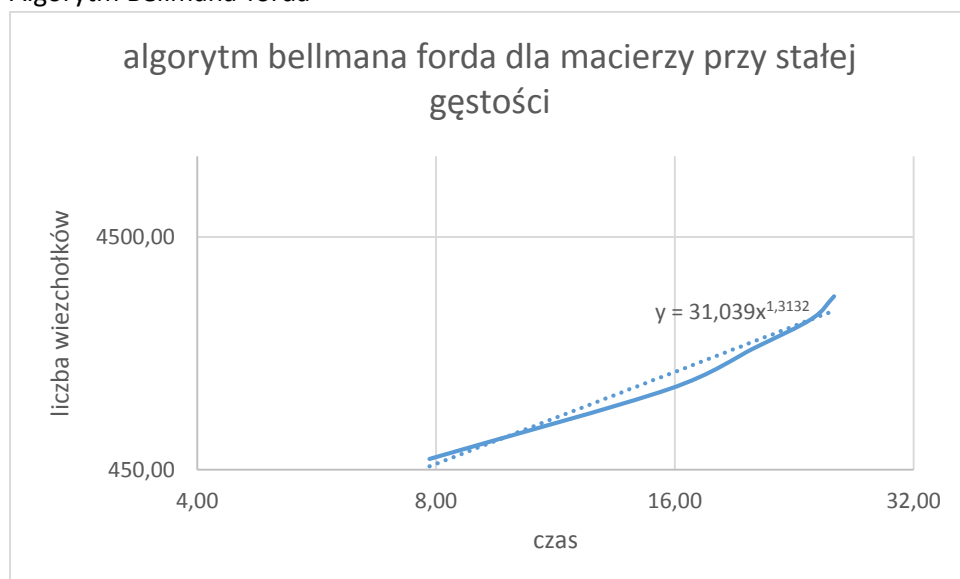


Złożoność obliczeniowa: $O(n^{1.9})$

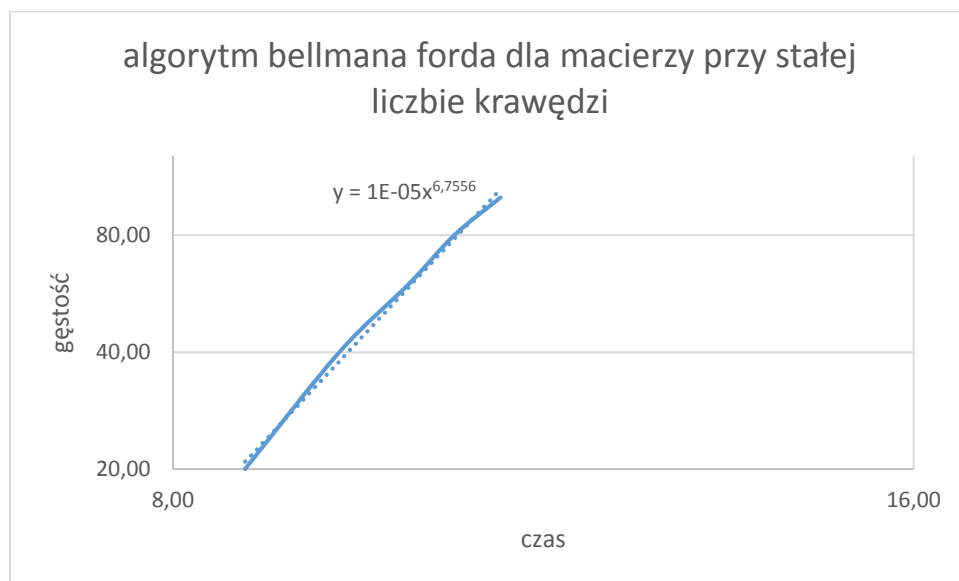


Złożoność obliczeniowa

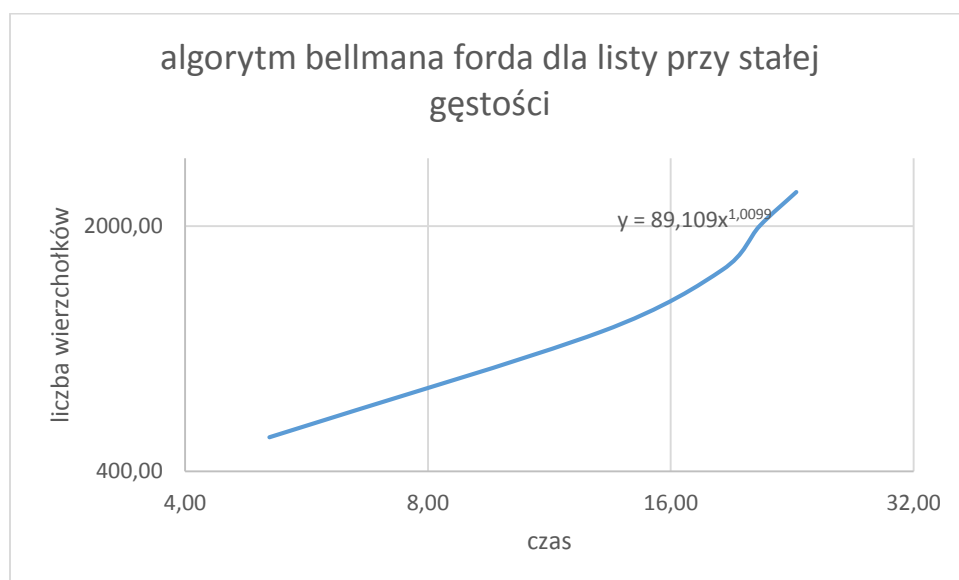
- Algorytm Bellmana-forda



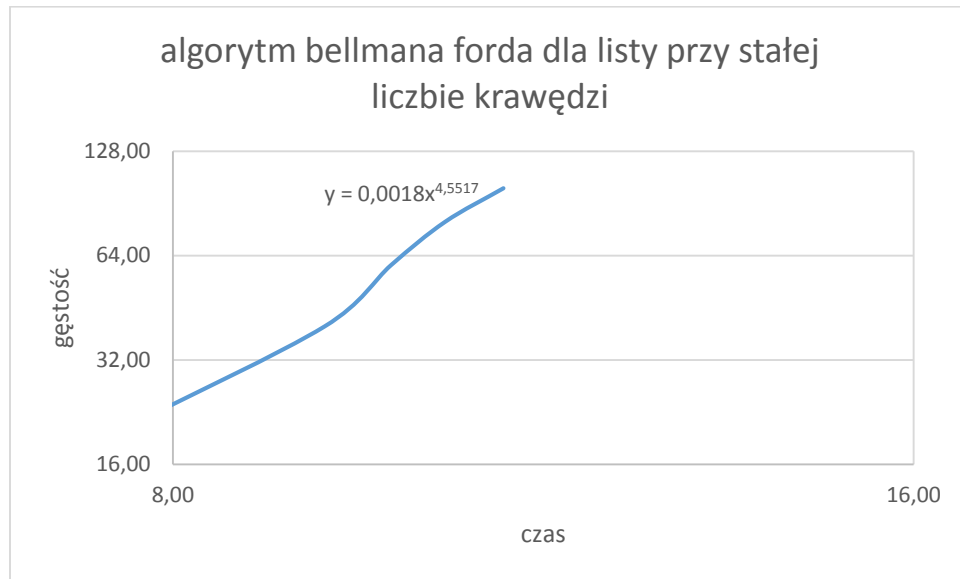
Złożoność obliczeniowa: $O(n^{1,3})$



Złożoność obliczeniowa:



Złożoność obliczeniowa: $O(n)$



Złożoność obliczeniowa:

4. Wnioski

- Algorytm Dijkstry szybciej radzi sobie na macierzach sąsiedztwa,
- Algorytm Bellmana-forda dużo lepiej radzi sobie na listach sąsiedztwa.
- Algorytm Dijkstry jest szybszy od algorytmu Bellmana-forda

5. Parametry komputera

Procesor: Intel Core i5-2430M

Pamięć ram: 4 GB

Karta graficzna: AMD Radeon HD 6470M 1GB