

Sprawozdanie z ćwiczenia laboratoryjnego
z
Projektowania Algorytmów i Metod Sztucznej Inteligencji

Sprawozdanie z 7-tego i 8-tego laboratorium

Sprawozdanie wykonał:

Rafał Januszewski

Data wykonania sprawozdania: 12.05.2016 r.

Data wykonywanego ćwiczenia: 25.04.2016 r.

Termin zajęć: poniedziałek 08:15

1. Wprowadzenie

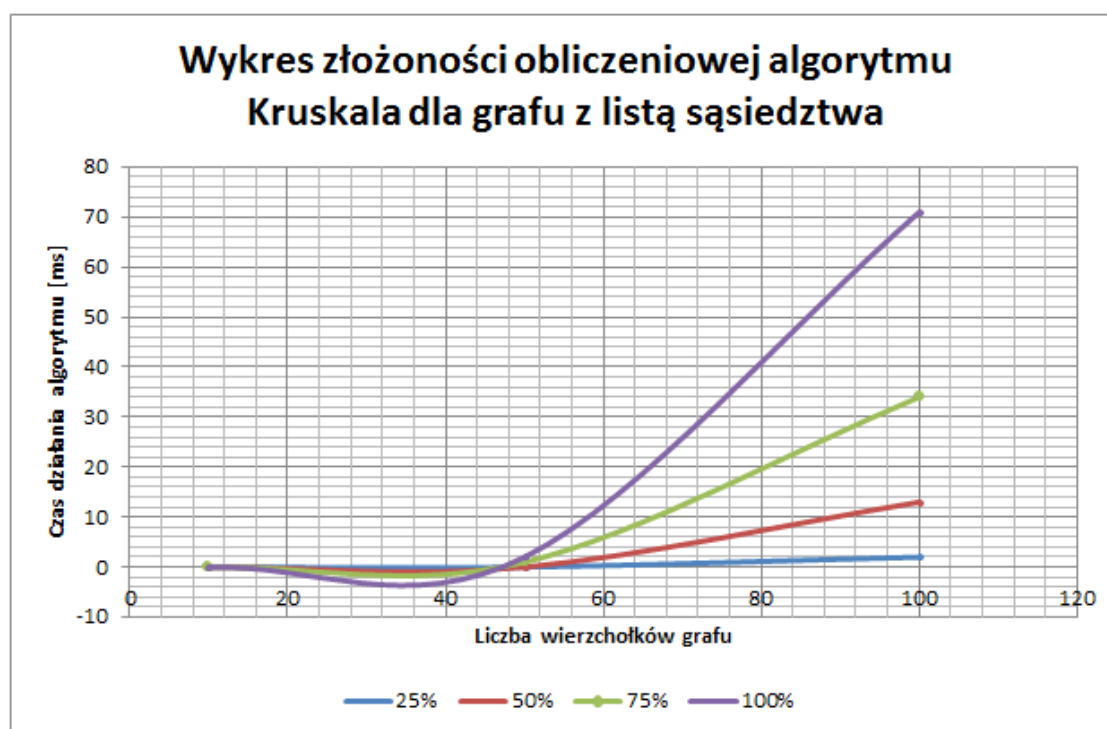
Celem ćwiczenia laboratoryjnego zaimplementowanie grafów nieskierowanych opartych na liście sąsiedztwa i macierzy sąsiedztwa oraz zbadanie efektywności dwóch algorytmów wyznaczania minimalnego drzewa rozpinającego: Kruskala i Prima. Testy efektywności algorytmów: Kruskala i Prima polegały na wygenerowaniu stu tablic o liczbie wierzchołków 10, 50, 100, 500 i 1000 zawierających liczby losowe typu całkowitoliczbowego i gęstości odpowiednio 25%, 50%, 75%, 100%. Powyższe algorytmy grafowe były testowane na systemie operacyjnym Linux Mint 17.3 64-bit. Parametry sprzętowe były następujące: model procesora i częstotliwość taktowania: Intel Core i3 M380 2,53GHz, pamięć RAM 2,8GiB.

2. Badanie algorytmu Kruskala

Algorytm Kruskala jest algorytmem zachłannym wyznaczania minimalnego drzewa rozpinającego. Działanie algorytmu polega na dodawaniu do krawędzi o najmniejszej wartości wagi do lasu. Algorytm wykorzystuje rozłączne zbiory wierzchołków. Złożoność obliczeniowa algorytmu Kruskala powinna wynosić $E \cdot \log(V)$.

Tabela 1. Badanie efektywności algorytmu Kruskala dla grafu opartego na liście sąsiedztwa.

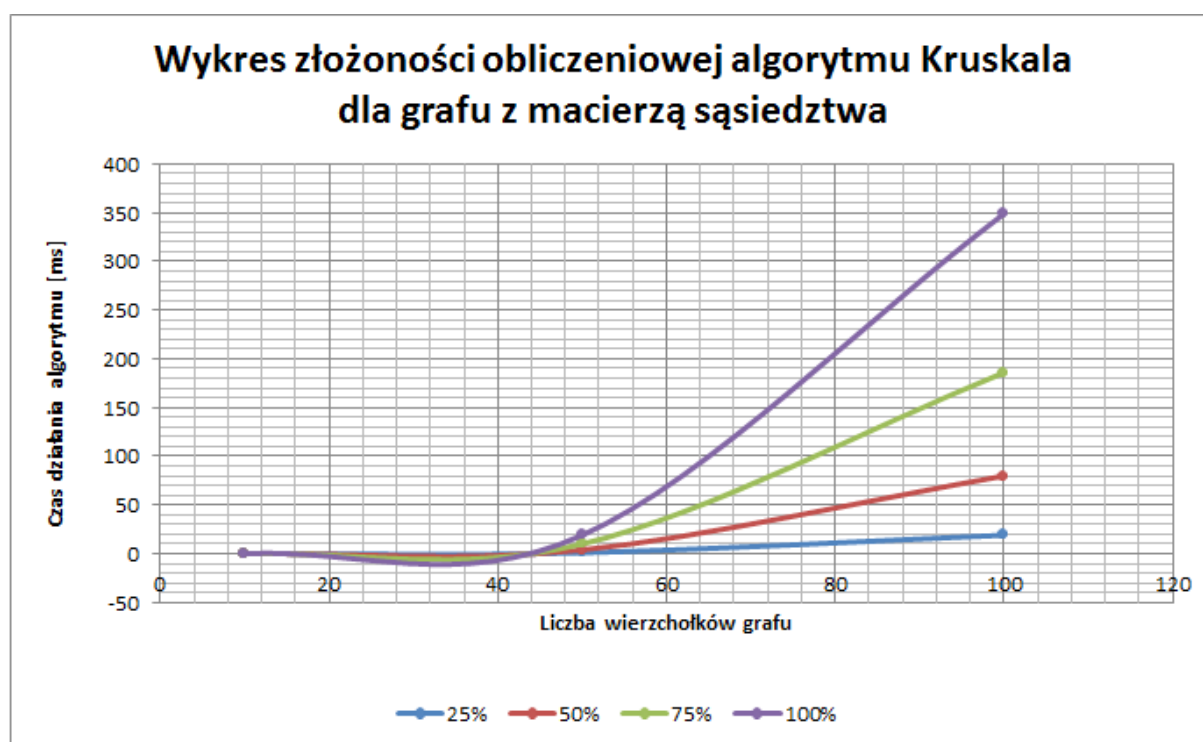
liczba wierzchołków	czas działania[ms]			
	Gęstość grafu [%]			
	25	50	75	100
10	0	0	0	0
50	0	0	1	2
100	2	13	34	71
500	12943	b. dużo	b. dużo	b. dużo
1000	b. dużo	b. dużo	b. dużo	b. dużo



Rys.1. Wykres złożoności obliczeniowej algorytmu Kruskala dla grafu opartego na liście sąsiedztwa.

Tabela 2. Badanie efektywności algorytmu Kruskala dla grafu opartego na macierzy sąsiedztwa.

liczba wierzchołków	czas działania[ms]			
	Gęstość grafu [%]			
	25	50	75	100
10	0	0	0	0
50	1	4	10	19
100	19	80	186	349
500	17582	b. dużo	b. dużo	b. dużo
1000	b. dużo	b. dużo	b. dużo	b. dużo



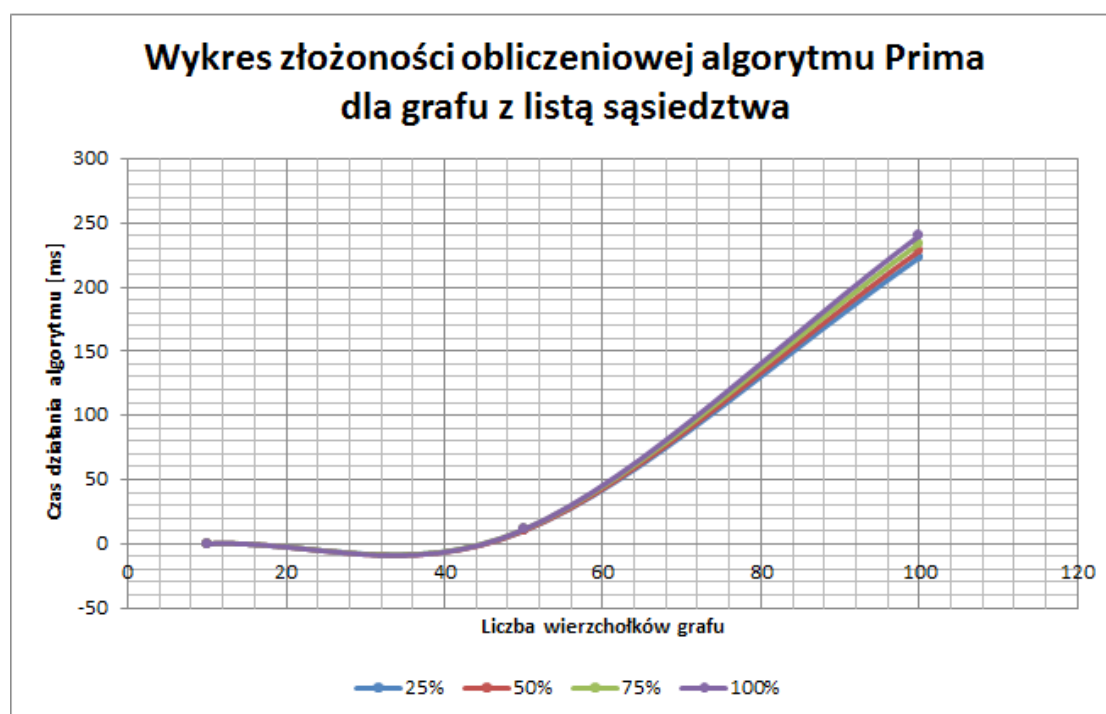
Rys.2. Wykres złożoności obliczeniowej algorytmu Kruskala dla grafu opartego na macierzy sąsiedztwa.

3. Badanie algorytmu Prima

Algorytm Prima jest algorytmem zachłannym wyznaczania minimalnego drzewa rozpinającego. Działanie algorytmu polega na dodawaniu do krawędzi, tak aby połączone wierzchołki nie tworzyły zamkniętego cyklu. W przypadku zastosowanie kolejki priorytetowej złożoność obliczeniowa algorytmu Prima powinna wynosić $E \cdot \log(V)$.

Tabela 3. Badanie efektywności algorytmu Prima dla grafu opartego na liście sąsiedztwa.

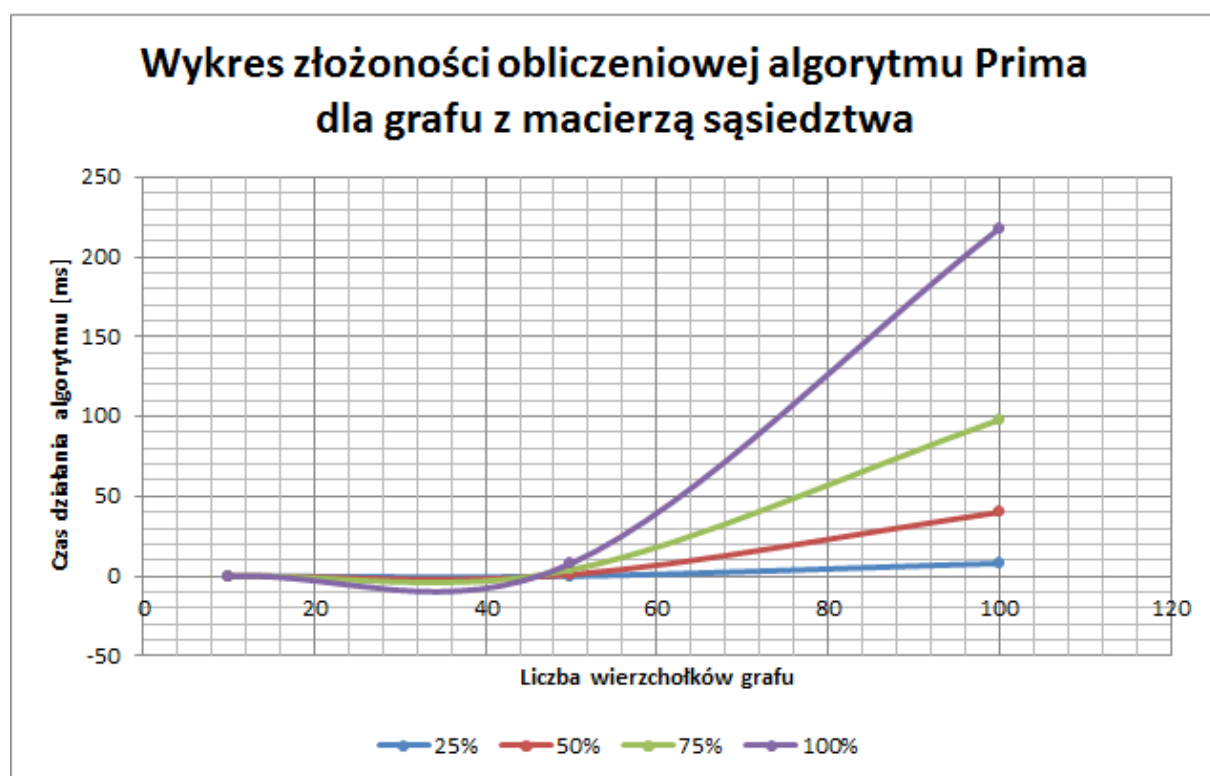
liczba wierzchołków	czas działania[ms]			
	Gęstość grafu [%]			
	25	50	75	100
10	0	0	0	0
50	10	10	11	11
100	223	228	234	240
500	b. dużo	b. dużo	b. dużo	b. dużo
1000	b. dużo	b. dużo	b. dużo	b. dużo



Rys.3. Wykres złożoności obliczeniowej algorytmu Prima dla grafu opartego na liście sąsiedztwa.

Tabela 4. Badanie efektywności algorytmu Kruskala dla grafu opartego na macierzy sąsiedztwa.

liczba wierzchołków	czas działania[ms]			
	Gęstość grafu [%]			
	25	50	75	100
10	0	0	0	0
50	0	1	4	8
100	8	40	98	218
500	23509	b. dużo	b. dużo	b. dużo
1000	b. dużo	b. dużo	b. dużo	b. dużo



Rys.4. Wykres złożoności obliczeniowej algorytmu Prima dla grafu opartego na macierzy sąsiedztwa.

5. Wnioski

- Algorytm Kruskala

Na podstawie wykresów złożoności obliczeniowej algorytmu Kruskala można stwierdzić, że dla małej liczby wierzchołków (od 10 do 100) złożoność obliczeniowa wynosi w przybliżeniu $E \cdot \log(V)$, co jest zgodne z założeniami. Dla grafu opartego na macierzy sąsiedztwa czas działania algorytmu Kruskala przy liczbie wierzchołków równej 100 jest zdecydowanie większy niż dla działania tego samego algorytmu na grafie zbudowanym w oparciu o listę sąsiedztwa. Dla wierzchołków mniejszych niż 100 czas działania algorytmu na grafie z macierzą sąsiedztwa jest porównywalny dla czasu działania tego samego algorytmu na grafie z listą sąsiedztwa. Z wykresów można zauważyć, że wraz ze wzrostem liczby wierzchołków przy stałej gęstości grafu – większej od 50% rośnie złożoność obliczeniowa od $E \cdot \log(V)$ do $\log(V)$.

- Algorytm Prima

Z wykresu złożoności obliczeniowej algorytmu Prima działającego na liście sąsiedztwa można zauważyć, złożoność obliczeniowa nie zależy od gęstości grafu, tylko od liczby wierzchołków. Tak jak w przypadku algorytmu Kruskala złożoność obliczeniowa $E \cdot \log(V)$ występuje dla małej liczby wierzchołków (od 10 do 100). Algorytm Prima działający na grafie z macierzą sąsiedztwa ma nieznacznie większą złożoność obliczeniową niż ten sam algorytm działający na grafie z listą sąsiedztwa.

Bardzo duży czas wykonywania algorytmów: Kruskala i Prima przy dużej liczbie wierzchołków może wynikać ze sposobu implementacji ww. algorytmów w programie.

6. Literatura

Strony internetowe:

<https://pl.wikipedia.org/>

<http://www.algorytm.edu.pl>

Cormen T. H., Leiserson C. E., Rivest R. L., *Wprowadzenie do algorytmów*, WNT, Warszawa 2001