

# Programowanie algorytmów i metody sztucznej inteligencji

## Laboratorium 9 i 10

Prowadzący: dr inż. Łukasz Jeleń

Sprawozdanie wykonał: Rafał Januszewski

Data wykonania ćwiczenia: 09.05.2016 r.

Data wykonania sprawozdania: 18.05.2016 r.

### 1 Wprowadzenie

Celem ćwiczenia laboratoryjnego było zaimplementowanie grafów nieskierowanych opartych na liście sąsiedztwa i macierzy sąsiedztwa oraz zbadanie efektywności algorytmu Dijkstry - algorytmu wyznaczania najkrótszych ścieżek w grafie. Testy efektywności algorytmu Dijkstry polegały na wygenerowaniu stu tablic o liczbie wierzchołków 10, 50, 100, 500 i 1000 zawierających liczby losowe typu całkowitoliczbowego i gęstości grafów odpowiednio: 25%, 50%, 75%, 100%.

Jako wierzchołek początkowy został wybrany wierzchołek grafu o indeksie 0. Wszystkie wczytywane krawędzie grafów miały nieujemne wagi. Kolejka priorytetowa, wykorzystywana w algorytmie została zbudowana na kopcu. Powyższe algorytmy grafowe były testowane na systemie operacyjnym Ubuntu 14.04 64-bit. Parametry sprzętowe były następujące: model procesora i częstotliwość taktowania: Intel Core i3 M380 2,53GHz, pamięć RAM 2,8GiB.

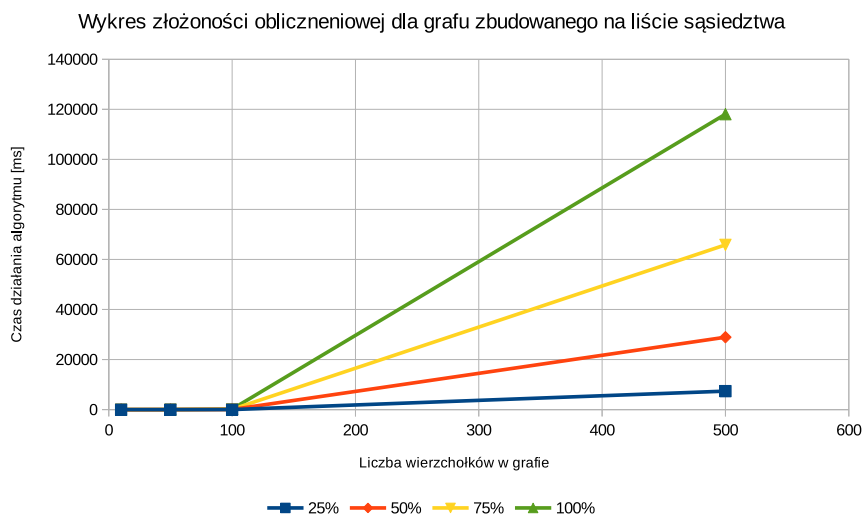
### 2 Badanie algorytmu Dijkstry

Algorytm Dijkstry wyznacza najkrótsze ścieżki od zadanego wierzchołka startowego w grafie, pod warunkiem że wszystkie krawędzie mają nieujemne wagi. Działanie algorytmu Dijkstry polega na wpisaniu wierzchołków wraz z odległościami (wagami) do kolejki priorytetowej. Następnie zostają usuwane wierzchołki o najmniejszej odległości od wierzchołka startowego. Dla każdego z tych wierzchołków algorytm oblicza najkrótszą ścieżkę od wierzchołka startowego. Działanie algorytmu kończy się wraz z usunięciem wszystkich elementów z kolejki priorytetowej. Algorytm zwraca tablicę najkrótszych ścieżek od zadanego wierzchołka starto-

wego. Czas działania algorytmu Dijkstry przy kolejce priorytetowej zaimplementowanej na kopcu wynosi  $O(E \cdot \log V)$ , a w wersji pesymistycznej  $O((V + E) \log V)$ .

Tablica 1: Badanie efektywności dla grafu opartego na liście sąsiedztwa.

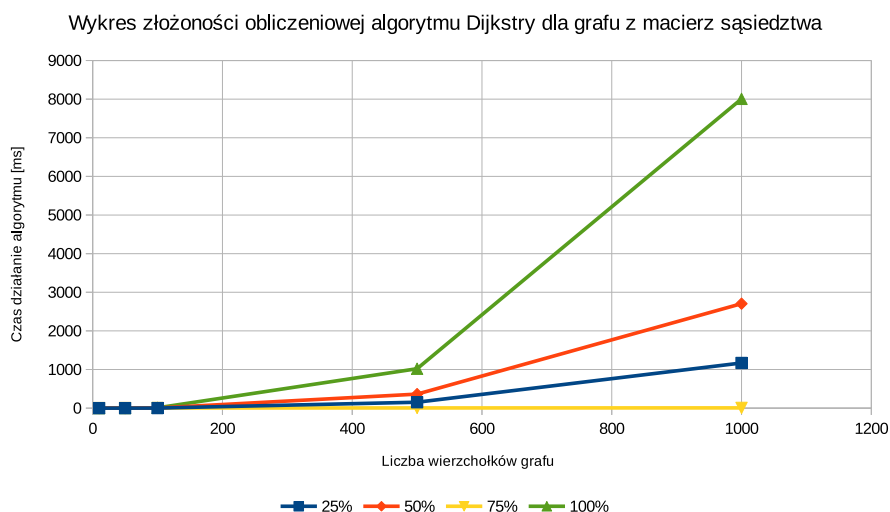
	czas działania [ms]			
	gęstość grafu [%]			
liczba wierzchołków	25	50	75	100
10	0	0	0	0
50	0	3	6	11
100	13	48	104	182
500	7380	28931	65846	118069
1000	116779	-	-	-



Rysunek 1: Wykres złożoności obliczeniowej algorytmu Dijkstry dla grafu opartego na liście sąsiedztwa.

Tablica 2: Badanie efektywności dla grafu opartego na macierzy sąsiedztwa.

	czas działania [ms]			
	gęstość grafu [%]			
liczba wierzchołków	25	50	75	100
10	0	0	0	0
50	0	0	0	1
100	1	3	5	8
500	152	362	643	1019
1000	1169	2704	4959	8008



Rysunek 2: Wykres złożoności obliczeniowej algorytmu Dijkstry dla grafu opartego na macierzy sąsiedztwa.

### 3 Wnioski

Z wykresu efektywności działania algorytmu Dijkstry działającego na liście sąsiedztwa można zauważyć, najgorsza złożoność obliczeniowa dla małej liczby wierzchołków (od 10 do 100) wynosi  $O(V)$ . Algorytm Dijkstry działający na grafie z listą sąsiedztwa ma znacznie większą złożoność obliczeniową dla dużej liczby wierzchołków ( $O(E \cdot \log V)$ ) niż ten sam algorytm działający na grafie z listą sąsiedztwa. Bardzo duży czas wykonywania algorytmu przy dużej liczbie wierzchołków może wynikać ze złożoności obliczeniowej metod grafu (opposite oraz incidentEdges).

Na podstawie wykresu efektywności algorytmu Dijkstry dla grafu zbudowanego na macierzy sąsiedztwa czasowa złożoność obliczeniowa dla dużej liczby wierzchołków (1000) wynosi w przybliżeniu  $O(V \cdot \log V)$ , a dla 500 wynosi  $O(V)$  - co jest zgodne z założeniami. Dla grafu opartego na macierzy sąsiedztwa czas działania algorytmu Dijkstry przy liczbie wierzchołków równej od 10 do 100 wynosi kilka milisekund. Z wykresu efektywności algorytmu Dijkstry można zauważyć, że algorytm jest najbardziej skuteczny dla grafów rzadkich i o najmniejszej liczbie wierzchołków.

W przeprowadzonych badaniach działania algorytmu Dijkstry wynika, że algorytm najskuteczniej wyznacza najkrótsze ścieżki dla grafu zbudowanego na macierzy sąsiedztwa.

## Literatura

- [1] T.H. Cormen, C.E. Leiserson, R.L. Rivest, *Wprowadzenie do algorytmów*. WNT, Warszawa 2001
- [2] J. Wałaszek, *Najkrótsza ścieżka w grafie ważonym Algorytm Dijkstry* [online]. Tarnów, I LO, [dostęp: 14 maja 2016], Dostępny w internecie:  
[http : //eduinf.waw.pl/inf/alg/001<sub>s</sub>earch/0138.php](http://eduinf.waw.pl/inf/alg/001_search/0138.php)
- [3] Instytut Automatyki i Robotyki Wydziału Mechatroniki Politechniki Warszawskiej, *Kolejki priorytetowe* [online]. [dostęp: 14 maja 2016], Dostępny w internecie:  
[http : //iair.mchtr.pw.edu.pl/bputz/aisd<sub>c</sub>pp/lekcja6/segment1/main.htm.php](http://iair.mchtr.pw.edu.pl/bputz/aisd_cpp/lekcja6/segment1/main.htm.php)