

# **Sprawozdanie**

*Autorzy: Angelika Ostrowska, Kacper Straszak*

*Data: 25.05.2023*

*AISDI Zad6*

## **Grafy**

Zaimplementowano program, który wczytuje plik zawierający planszę jako argument wywołania programu. Korzystając z algorytmu Dijkstry znajduje on we wczytanej planszy dowolną najmniej kosztowną trasę przejścia między polami z cyfrą 0.

Na koniec program wyświetla wynik - pola planszy, które tworzą najkrótszą, najmniej kosztowną trasę.

### *Algorytm Dijkstry:*

Jest to algorytm umożliwiający znalezienie najkrótszej/najmniej kosztownej ścieżki z jednego wybranego wierzchołka grafu do drugiego.

Przechowujemy tablicę odległości wszystkich wierzchołków od początkowego wierzchołka, początkowo ustawiając wszystkie odległości na nieskończone (poza odległością wierzchołka początkowego od samego siebie). Przechowujemy również tablicę poprzedników, którą na bieżąco aktualizujemy.

Algorytm wybiera wierzchołek  $v$  o najmniejszej odległości spośród nieodwiedzonych (zacznie od wybranego początkowego wierzchołka grafu) i przenosi go do wierzchołków odwiedzonych. Jego odległość nie będzie się już zmieniać, to finalna długość najkrótszej ścieżki od niego do wierzchołka początkowego.

Następnie algorytm sprawdza nieodwiedzone wierzchołki sąsiadujące z  $v$ . Jeśli ścieżka prowadząca od wierzchołka początkowego do któregoś z sąsiadów przez wierzchołek  $v$  jest krótsza niż aktualnie najmniej kosztowna, to aktualizuje koszt dotarcia do tego wierzchołka.

Teraz należy powtarzać powyższe punkty aż do dotarcia do ostatniego wierzchołka grafu.

Po zakończeniu algorytm Dijkstry posiada tablicę kosztów, która przechowuje najkrótsze znalezione odległości od wierzchołka startowego do wszystkich innych wierzchołków w grafie, oraz mapę przedstawiającą najkrótszą trasę pomiędzy dwoma wybranymi punktami.

### *Przedstawienie działania algorytmu na przykładach:*

111122	1	1	1
104122	0		1
942111		1	1
996411			1
990411		0	1
991111	1	1	1

graf1.txt

0113	0	1	1
3332			3
1110		1	0

graf2.txt

0144	0	1	4
6521		2	1
1891			1
8511	5	1	1
1195	1		
1107	1	0	

graf3.txt

123456789			0
123456089			7
123456789			6
666666666			1
191919191			4
222244444		4	4
112333455			
112358132	3	3	
919294959	5		
777707777	9		
111111111	0		
125472891			

graf4.txt