[AISDI] Raport.

Konstantin Panov. Grupa 105.

# Wyjaśnienie implementacji:

Algorytm jest zaimplementowany w funkcji ***dijkstra(filename)****, argumentem jest nazwa pliku z danymi wejściowymi.*

## Wczytywanie danych:

Funkcja wczytuje tekst z pliku, dzieli tekst względem nowych linii i zapisuję to w tablice ***data.*** Po czym zapamiętuje wysokość (Ilość wierszy) oraz szerokość (długość dowolnego wiersza) planszy. Pola planszy są przechowywane w tablice ***nodes*** jako obiekty klasy ***Node***, mające następujące parametry:

* ***index*** – Kolejność pola planszy w tablice (0 – góra lewo, ostatni – dół prawo).
* ***cost*** – Koszt przejścia do pola.
* ***distance*** – Najkrótsza droga do tego pola od pola początkowego. Najpierw ma wartość ***sys.maxsize –*** Największa wartość liczbowa w systemie (odpowiednik nieskończoności).
* ***neighbors –*** Lista indexów sąsiednich pól (od dwóch do czterech sąsiedzi).
* ***previous\_node –*** Pole leżące najbliżej początku.

Przechodząc po każdemu symbolu w każdym elemencie tablicy ***data*** funkcji tworzy obiekt klasy ***Node*** dla każdego pola, obliczając index pola i pól sąsiednich na podstawie pozycji aktualnego pola oraz ogólnej szerokości planszy, i wyszukuje pole początkowe i końcowe i zapisuje je w tablice ***start\_end***. Obiekty klasy ***Node*** są zapisane w tablice nodes. Na podstawie tablicy ***nodes*** tworzy się set ***unvisited***, przechowywujący jeszcze nie sprawdzone pola. Parametr ***distance*** pola początkowego ustalona na 0.

## Implementacja algorytmu:

Póki set ***unvisited*** nie jest pusty:

* wartość zmiennej jest równa indeksu pola o minimalnej odległości od pola początkowego.
* liczy się odległość do sąsiadów tego pola i zapisuje się, jeśli jest mniejsza od obliczonej wcześniej.
* Pole o najmniejszej odległości od początku jest usuwane ze set’u ***unvisited.***

# Przykładowe plansze:

## Pierwsza plansza:

## 

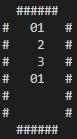
## Wynik:



## Druga plansza:



## Wynik:



## Trzecia plansza:



## Wynik:

