

# Specyfikacja funkcjonalna

## Generacja grafu i odnalezienie najkrótszej ścieżki pomiędzy węzłami - **graph**

Ulyana Petrashevich, Inga Maziarz

20.04.2022

### Cel projektu

Program **graph** ma na celu znalezienie najkrótszej ścieżki między dwoma wybranymi punktami. Program może wczytywać graf z pliku o określonym formacie lub wygenerować graf o zadanej liczbie kolumn i wierszy z losowymi wartościami wag krawędzi w określonym przedziale. Sprawdzana jest spójność grafu. Program działa w trybie wsadowym.

### Dane wejściowe

Program przyjmuje dane o grafie z pliku tekstowego, w którym są opisane parametry węzłów:

- liczba wierszy i kolumn węzłów: liczby całkowite  $> 0$ ;
- numer węzła, do którego prowadzi krawędź od węzła obecnego: numer obecnego węzła odpowiada numerowi linii w pliku - 1, nie licząc linii opisującej liczbę kolumn i wierszy. Przyjęta jest konwencja, że węzły przyjmują indeksy od 0 do  $n - 1$  (gdzie  $n$  to sumaryczna liczba węzłów) i są numerowane kolejno od lewej do prawej;
- waga krawędzi: liczba rzeczywista  $> 0$ .

W pierwszej linii pliku podana jest ilość wierszy i kolumn grafu. Parametry do jednego węzła są definiowane poniżej w jednej linii. Numer węzła docelowego jest oddzielany od wagi spacją i znakiem  $:$ . Parametry dla jednego węzła są oddzielane dwiema spacjami.

W przypadku generowania grafu przez program, parametry są przekazywane za pomocą argumentów wywołania programu. Jeżeli parametry nie zostaną określone, wygenerowany zostanie graf bazujący na domyślnych wartościach (10 wierszy, 10 kolumn, przedział wagowy krawędzi  $[0.01, 10.0]$ ).

Przykład danych wejściowych:

```
2 3
1 :0.8864916775696521  2 :0.2187532451857941
0 :0.2637754478952221  5 :0.5486221194511974  2 :0.25413610146892474
3 :0.5380334165340379
2 :0.5486221194511974  5 :0.25413610146892474
3 :0.31509645530519625  0 :0.40326574227480094  5 :0.8901867253885717
4 :0.44272335750313074
```

Wyżej jest zdefiniowany graf o rozmiarze 2 wierszy i 3 kolumn. Dla węzła numer 0 istnieją przejścia do węzła 1 o wadze 0.8865 i do węzła 2 o wadze 0.2188.

## Dane wyjściowe

Program jest wyposażony w interfejs graficzny.

Interfejs jest wyposażony w opcje:

- Size  
Pozwala na wyświetlenie rozmiaru podanego grafu lub wprowadzenie rozmiaru dla generacji grafu.
- Edge weight range  
Pozwala na wyświetlenie zakresu wartości wag krawędzi lub podanie zakresu dla generacji grafu.
- Connectivity  
Pozwala na wyświetlenie flagi spójności grafu lub wprowadzenie flagi dla generatora grafu (0 - niespójny, 1 - spójny, 2 - spójność losowa).
- Generate  
Przycisk uruchamia generator grafu z parametrami, wprowadzonymi w poprzednich okienkach.
- Read  
Przycisk pozwala na wybranie gotowego pliku z grafem i jego wczytanie.
- Save graph  
Przycisk pozwala na zapisanie danego grafu do pliku.
- Save path  
Przycisk pozwala na zapisanie wyliczonej ścieżki do pliku.
- Clear  
Przycisk pozwala na wyczyszczenie narysowanych ścieżek w grafie.

- Delete

Przycisk usuwa rysunek grafu.

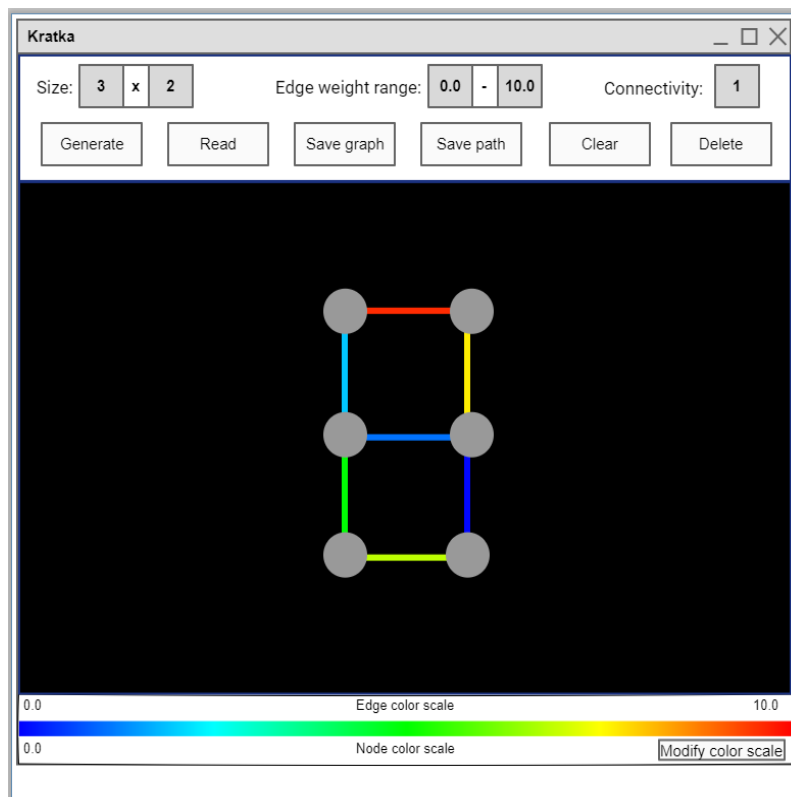
- Modify color range

Przycisk pozwala na dostosowanie spektrum kolorów używanych do oznaczania wag krawędzi.

Postać narysowanego grafu:

Wagi krawędzi między węzłami będą odwzorowane kolorem (wzrastająco od niebieskiego do czerwonego). Tuż po narysowaniu grafu węzły będą miały kolor szary. Po naciśnięciu na wybrany węzeł lewym przyciskiem myszy, kolor węzłów będzie odwzorowywał koszt dojścia od wybranego węzła. Po naciśnięciu na inny węzeł prawym klawiszem myszy zostanie narysowana ścieżka w kolorze białym. Możliwe jest wyznaczenie wielu najkrótszych ścieżek na tym samym rysunku poprzez wybór kolejnych węzłów początkowych i końcowych.

## Makieta interfejsu graficznego



## Teoria

Graf jest **spójny**, jeżeli dla każdej pary węzłów istnieje łącząca je ścieżka. Program sprawdza spójność grafu za pomocą algorytmu przeszukiwania grafu wszerek (BFS). Działanie algorytmu ma następane kroki:

- Zaczynamy od węzła początkowego. Zaznaczamy go jako odwiedzony i dodajemy do kolejki wszystkie węzły, z którymi jest powiązany, w kolejności od węzła z najmniejszym indeksem.
- Odwiedzamy następny węzeł w kolejce. Dodajemy do kolejki wszystkie węzły z nim powiązane i jeszcze nieodwiedzone.
- Powtarzamy poprzedni krok, aż kolejka będzie pusta. Jeżeli wszystkie węzły w grafie zostały odwiedzone, to można stwierdzić, że graf jest spójny.

Poszukiwanie najkrótszej ścieżki między zadanymi punktami program będzie realizował przez **algorytm Dijkstry**. Algorytm polega na odnajdowaniu najkrótszych ścieżek od zadanego węzła do każdego innego.

- Dla każdego węzła ustawiamy długość ścieżki na nieskończoność lub wartość, która do niej dąży; długości przy węźle początkowym nadajemy wartość 0.
- Oznaczamy węzeł jako odwiedzony. Dla każdego węzła połączonego z początkowym, przypisujemy długość równą wadze krawędzi ich łączących.
- Z nieodwiedzonych węzłów znajdujemy węzeł o najmniejszej przepisanej długości. Oznaczamy go jako odwiedzony. Dla każdego węzła sąsiadującego z obecnym liczymy wartość „długość przy obecnym węźle + waga krawędzi łączącej”. Jeżeli znaleziona wartość jest mniejsza niż przypisana do sąsiadującego węzła, podmieniamy ją.
- Powtarzamy poprzedni krok, aż zostaną odwiedzone wszystkie węzły. Po zakończeniu każdy węzeł będzie miał przypisaną długość najkrótszej ścieżki od węzła początkowego. Samą ścieżkę możemy odtworzyć od końca, jeżeli przy każdym przypisaniu węzłowi nowej długości będziemy zapamiętywali numer poprzedniego węzła.

## Komunikaty błędów

W przypadku, gdy znaczące dane nie są podane, program **graph** stara się nadawać im przypisane domyślne znaczenia. Lecz w przypadkach, gdy dane nie można zastąpić lub są podane błędnie, będą wyświetlane odpowiednie komunikaty.

1. Nie można otworzyć podanego pliku (np. przy wczytywaniu grafu): Nie udało się otworzyć pliku *filename*.
2. Nie podano liczby wierszy/kolumn: Nie podano liczby wierszy/kolumn. Wygenerowanie grafu nie jest możliwe.

3. Podany graf nie jest spójny: Podany graf nie jest spójny. Nie można wyznaczyć najkrótszych ścieżek.
4. Podana liczba wierszy węzłów generowanego grafu mniejsza/równa 0: Liczba wierszy musi być większa 0. Ustawiam wartość 10.
5. Podana liczba kolumn węzłów generowanego grafu mniejsza/równa 0: Liczba kolumn musi być większa 0. Ustawiam wartość 10.
6. Dolna granica zakresu losowanych wag mniejsza/równa 0: Dolna granica przedziału losowanych wartości wag musi być większa 0. Ustawiam wartość 0.01.
7. Górna granica zakresu losowanych wag mniejsza/równa 0: Górna granica przedziału losowanych wartości wag musi być większa 0. Ustawiam wartość  $x + 1$ .
8. Flaga spójności nie równa się 0, 1 lub 2: Nieprawidłowa flaga spójności. Ustawiam wartość 2 (losowa spójność)