

# Specyfikacja końcowa - Graf Java

13.06.2022 r.

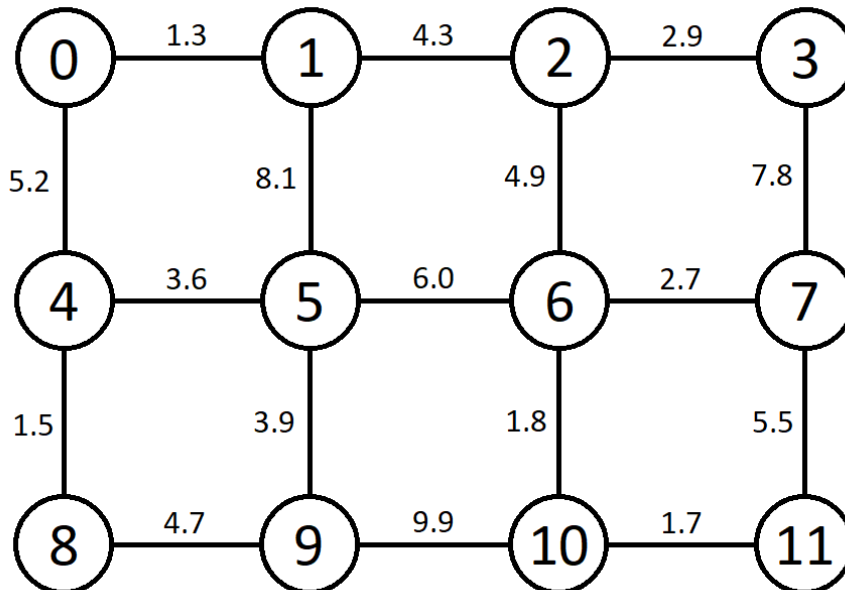
\* Wszystkie wprowadzone zmiany zostały oznaczone podkreśleniem.

## 1 Cel projektu

Celem projektu jest stworzenie aplikacji, która generuje, wyświetla i analizuje prostokątny, nieskierowany, ważony graf. Analiza polega na możliwości sprawdzenia, czy graf jest spójny za pomocą algorytmu BFS (breadth-first search) oraz znalezienia najkrótszej ścieżki pomiędzy dwoma wskazanymi przez użytkownika wierzchołkami grafu za pomocą algorytmu Dijkstry.

Wierzchołki grafu są ponumerowane od 0, od lewej do prawej strony grafu, a następnie przechodzą do wiersza poniżej.

Przykładowy graf:



## 2 Wstęp teoretyczny

### 2.1 Algorytm BFS

Algorytm BFS (breadth-first search), czy inaczej algorytm przeszukiwania wszerz, polega na przejściu grafu od wybranego wierzchołka aż do momentu, gdy wszystkie wierzchołki, do których można było dotrzeć zostały odwiedzone. Ze względu na swoją specyfikę może zostać wykorzystany do sprawdzania spójności grafu, ponieważ odwiedzone zostaną jedynie wierzchołki, które znajdują się w tym samym spójnym segmencie, co wierzchołek początkowy.

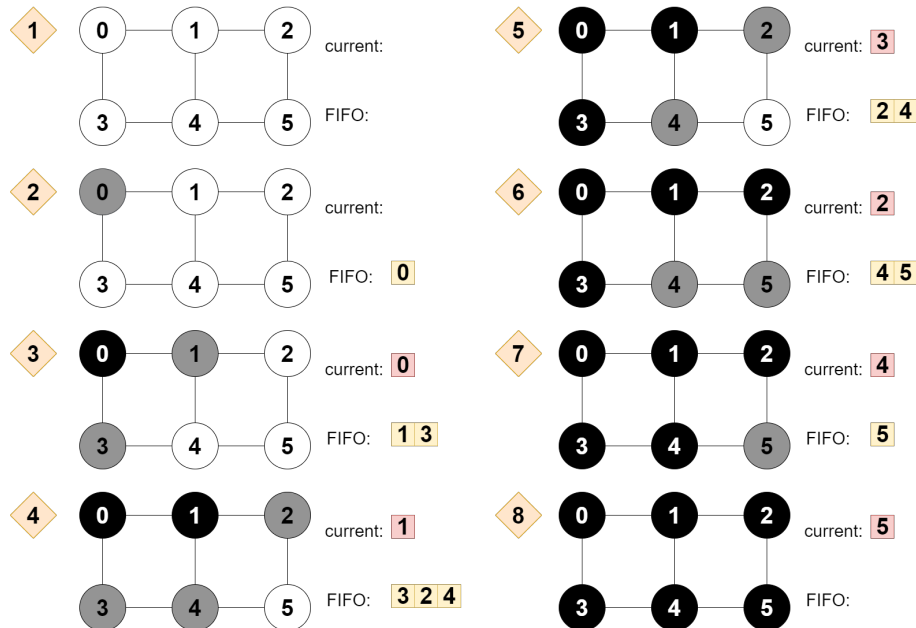
Każdy wierzchołek może się znaleźć w 1 z 3 stanów: nieodwiedzony (biały), odkryty (szary) i odwiedzony (czarny). Na początku wszystkie wierzchołki są białe. Gdy wierzchołek został dodany do kolejki FIFO, staje się szary. Gdy wszystkie jego sąsiednie wierzchołki zostały odkryte lub odwiedzone, staje się on czarny.

Aby sprawdzić spójność grafu, należy przeanalizować stany wszystkich wierzchołków. Jeśli którykolwiek pozostał biały po przejściu algorytmu BFS, to graf jest niespójny.

Lista kroków - algorytm BFS:

1. Pomaluj wszystkie wierzchołki grafu na białe.
2. Pomaluj wierzchołek początkowy na szaro.
3. Dodaj wierzchołek początkowy do kolejki.
4. Dopóki kolejka nie jest pusta wykonuj kroki 5. - 10.
5. Wyjmij z kolejki wierzchołek  $u$ .
6. Dla każdego wierzchołka sąsiedniego do  $u$  wykonaj krok 7., następnie przejdź do kroku 10.
7. Jeśli wierzchołek jest biały wykonaj kroki 8. - 9., w przeciwnym wypadku wróć do kroku 6.
8. Pomaluj wierzchołek na szaro.
9. Dodaj wierzchołek do kolejki.
10. Pomaluj wierzchołek  $u$  na czarno.

Przykład:



## 2.2 Algorytm Dijkstry

Algorytm Dijkstry służy do znajdowania najkrótszej ścieżki w grafie. Dla każdego wierzchołka w grafie zapamiętywana jest długość aktualnie najkrótszej ścieżki od wierzchołka początkowego oraz poprzednik, czyli sąsiedni wierzchołek przez który prowadzi ta ścieżka. Do uporządkowania wierzchołków według długości ścieżki służy kolejka priorytetowa.

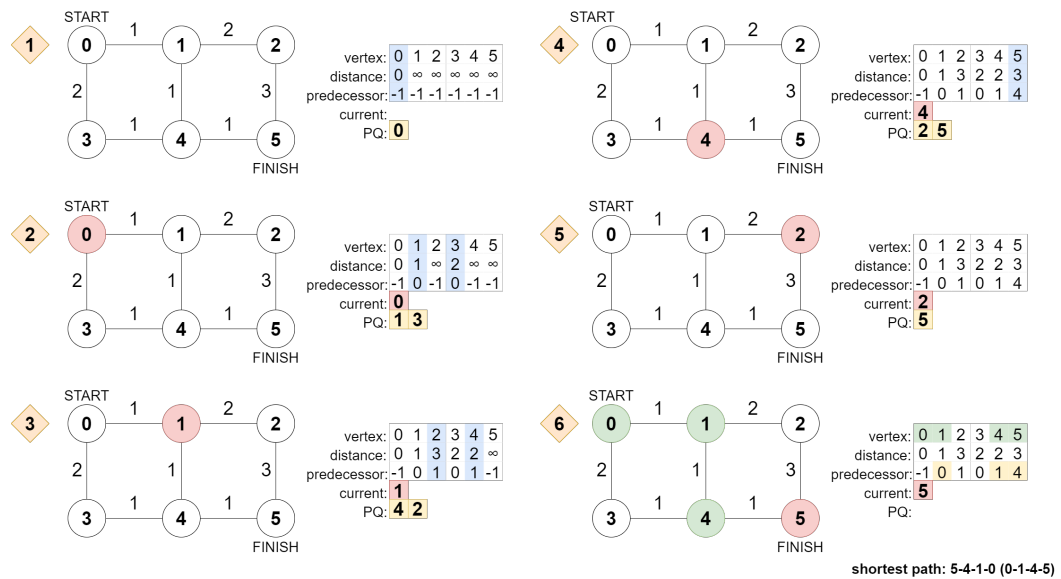
Najkrótszą ścieżkę od wybranego wierzchołka początkowego do wybranego wierzchołka końcowego można odczytać analizując poprzedników rozpoczynając od wierzchołka końcowego, następnie jego poprzednika, itd. aż do momentu dojścia do wierzchołka początkowego.

Lista kroków - algorytm Dijkstry:

1. Ustaw długość ścieżki dla każdego z wierzchołków grafu jako nieskończoność.
2. Ustaw poprzednika dla każdego z wierzchołków jako -1.
3. Ustaw długość ścieżki dla wierzchołka początkowego jako 0.
4. Dodaj wierzchołek początkowy do kolejki.
5. Dopóki kolejka nie jest pusta wykonuj kroki 6. - 11.
6. Wyjmij z kolejki wierzchołek u.

7. Dla każdego wierzchołka  $v$  sąsiedniego do  $u$  wykonaj krok 8.
8. Jeśli suma drogi do wierzchołka  $u$  i wagi krawędzi między wierzchołkami  $u$  i  $v$  jest mniejsza od obecnie zapamiętanej drogi do wierzchołka  $v$ , to wykonaj kroki 9. - 11.
9. Zapisz nową drogę do wierzchołka  $v$  jako sumę drogi do wierzchołka  $u$  i wagi krawędzi między wierzchołkami  $u$  i  $v$ .
10. Zaktualizuj pozycję wierzchołka  $v$  w kolejce.
11. Zapisz wierzchołek  $u$  jako poprzednika wierzchołka  $v$ .

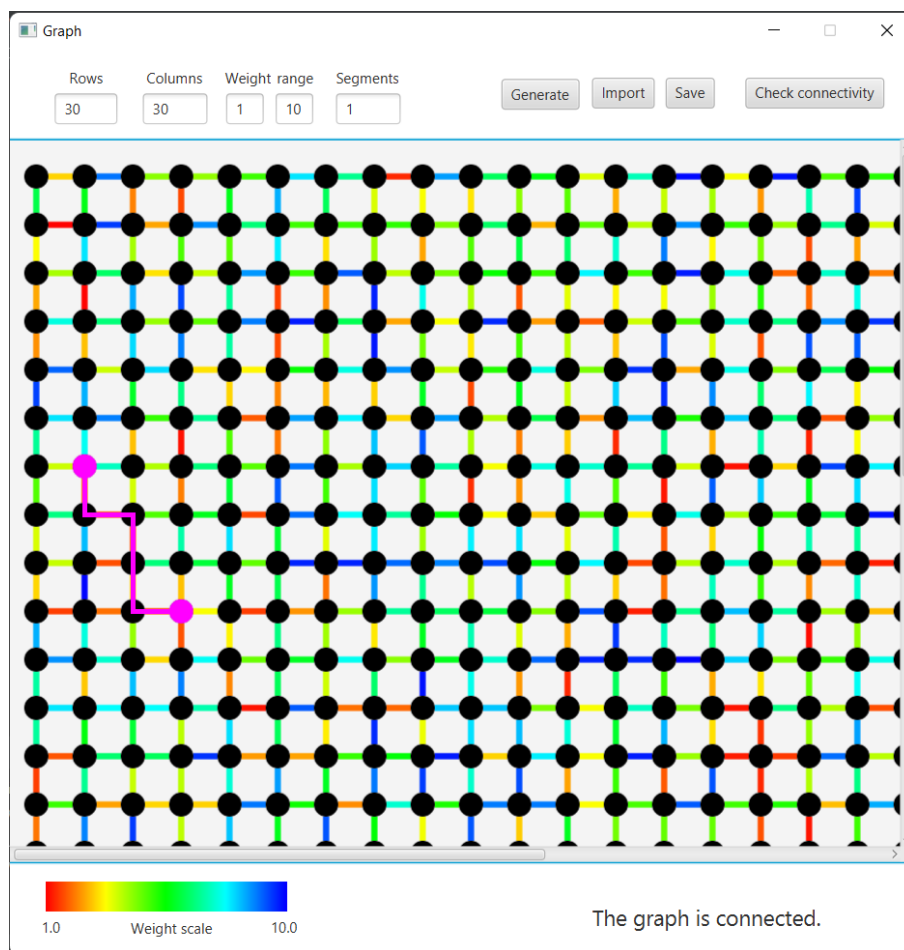
Przykład:



### 3 Sposób uruchomienia

Program będzie uruchamiany i obsługiwany za pomocą interfejsu graficznego.

Projekt GUI:



W przypadku wygenerowania grafu o wymiarach większych niż okno aplikacji, graf można przewijać paskami umieszczonymi na dole i z prawej strony planszy.

Kolory krawędzi między wierzchołkami grafu odpowiadają ich wagom według skali umieszczonej na dolnym pasku.

Przy uruchomieniu aplikacji okno jest puste, a przyciski `Save` i `Check connectivity` wyłączone.

Dostępne funkcjonalności:

- generowanie grafu o zadanych parametrach:
  1. Wprowadzenie liczby rzędów i kolumn grafu w pola tekstowe **Rows** i **Columns**:
    - $rows, columns \in N$
    - $rows \cdot columns \leq 10^4$
    - typ: int
  2. Wprowadzenie zakresu wag w pola tekstowe **Weight range**:
    - $weight1, weight2 \in R$
    - $weight1, weight2 \geq 0.0$
    - $weight1, weight2 \leq 100.0$
    - $weight1 < weight2$
    - typ: double
  3. Wprowadzenie liczby spójnych segmentów, na które jest podzielony graf w pole tekstowe **Segments**:
    - $segments \in N$
    - $segments \leq 10$
    - typ: int
  4. Zatwierdzenie wartości za pomocą przycisku **Generate**.
- wczytanie i wyświetlenie grafu z pliku tekstowego o określonym formacie:
  1. Kliknięcie przycisku **Import**.
  2. Wybór pliku w wyświetlonym oknie dialogowym.
- zapis wygenerowanego grafu do pliku tekstowego o określonym formacie:
  1. Kliknięcie przycisku **Save**.
  2. Wybór lokalizacji zapisu w wyświetlonym oknie dialogowym.
  3. Zapis grafu do pliku jest dostępny tylko dla grafów wygenerowanych za pomocą przycisku **Generate**.
- sprawdzenie spójności grafu:
  1. Kliknięcie przycisku **Check connectivity**.
  2. Wyświetlenie komunikatu o spójności grafu w prawym dolnym rogu.
- znalezienie najkrótszej ścieżki pomiędzy 2 wybranymi wierzchołkami:
  1. Wybór pierwszego wierzchołka prawym przyciskiem myszki.
  2. Wybór drugiego wierzchołka prawym przyciskiem myszki.
  3. Wyświetlenie najkrótszej ścieżki pomiędzy wierzchołkami na różowo.

4. Ponowny wybór któregoś z wybranych wierzchołków powoduje reset planszy (odzaczenie poprzedniej ścieżki).
5. Wybór trzeciego wierzchołka powoduje odzaczenie poprzedniej ścieżki oraz zaznaczenie wybranego wierzchołka na różowo.
6. W przypadku braku ścieżki pomiędzy wybranymi wierzchołkami, wyświetlany jest komunikat:  
Path doesn't exist  
The path between vertices ... and ... doesn't exist.

## 4 Dane wejściowe

Program akceptuje dane wejściowe w formie pliku tekstowego o określonym formacie. Separatorem dziesiętnym jest kropka.

Opis grafu powinien zamierać następujące elementy:

- wymiary grafu (liczba wierszy i liczba kolumn);
- opis krawędzi wychodzących z poszczególnych wierzchołków (numery wierzchołków docelowych wraz z wagami krawędzi).

Ogólny format pliku:

```
#-rows #-columns
description-of-the-0th-vertex
description-of-the-1st-vertex
...
```

Format opisu każdego wierzchołka:

```
vertex-number :edge-weight vertex-number :edge-weight ...
```

Fragment przykładowego pliku:

```
3 4
1 :1.3 4 :5.2
0 :1.3 2 :4.3 5 :8.1
1 :4.3 3 :2.9 6 :4.9
2 :2.9 7 :7.8
0 :5.2 5 :3.6 8 :1.5
1 :8.1 4 :3.6 6 :6.0 9 :3.9
2 :4.9 5 :6.0 7 :2.7 10 :1.8
3 :7.8 6 :2.7 11 :5.5
4 :1.5 9 :4.7
5 :3.9 8 :4.7 10 :9.9
6 :1.8 9 :9.9 11 :1.7
7 :5.5 10 :1.7
```

Graf reprezentowany przez ten plik jest przedstawiony na pierwszej stronie.

## 5 Dane wyjściowe

Program ma możliwość wygenerowania pliku wyjściowego o takim samym formacie, jak akceptowany format danych wejściowych. Umożliwia to ponowną analizę wygenerowanego grafu. Zapis do pliku jest możliwy tylko dla grafów wygenerowanych za pomocą przycisku **Generate**.

## 6 Struktura programu

Aktualna struktura programu została przedstawiona w javadoc'u dołączonym tej do specyfikacji końcowej.

Projekt został wykonany w środowisku IntelliJ z użyciem narzędzia Maven oraz technologii JavaFX.

## 7 Zachowania niestandardowe

W przypadku wystąpienia nieprawidłowości program zachowa się w następujący sposób:

- brak parametru (w polu: **Rows**, **Columns**, **Weight range**, **Segments**) - program pokazuje okno z informacją o błędzie, wyświetla graf poprzednio wygenerowany;  
Incorrect argument format  
Please provide arguments in the correct format.
- podanie wartości parametru poza dozwolonym zakresem - program pokazuje okno z informacją o błędzie, wyświetla graf poprzednio wygenerowany;  
Arguments out of range  
Please provide arguments only in the allowed range.
- podanie parametru nieakceptowanego przez program - program pokazuje okno z informacją o błędzie, wyświetla graf poprzednio wygenerowany;  
Incorrect argument format  
Please provide arguments in the correct format.
- użycie pliku o nieprawidłowym formacie - program pokazuje okno z informacją o błędzie, wyświetla graf poprzednio wygenerowany;  
Incorrect file format  
Please provide an input file formatted correctly.