

Specyfikacja funkcjonalna - Graf

Daria Danieluk, Weronika Zbierowska

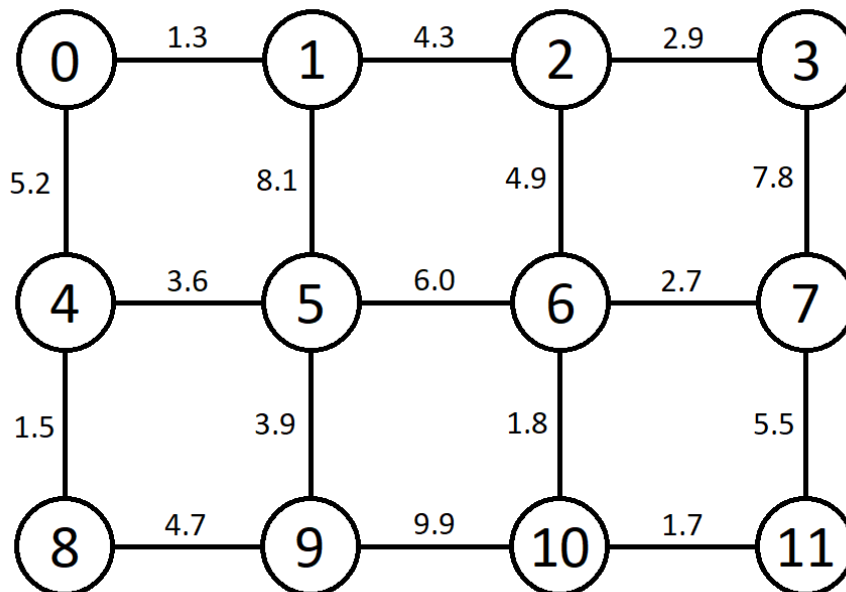
08.03.2022 r.

1 Cel projektu

Celem projektu jest stworzenie programu, który generuje 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 Sposób uruchomienia

Program należy uruchomić z linii poleceń w trybie wsadowym. Zmiana parametrów analizowanego grafu oraz wybór funkcji następuje poprzez użycie flag.

Program akceptuje następujące argumenty wywołania:

- **--size** *rows columns* - rozmiar generowanego grafu;
 $rows, columns \in N$; $rows \cdot columns \leq 10^6$;
domyślnie: $rows = 100$, $columns = 100$;
- **--weight** *w1 w2* - zakres losowanych wag dla krawędzi;
 $w1, w2 \in R$; $w1, w2 \geq 0.0$; $w1, w2 \leq 100.0$; $w1 < w2$;
domyślnie: $w1 = 0.0$, $w2 = 10.0$;
- **--segments** *n* - liczba spójnych segmentów, na które jest podzielony graf;
 $n \in N$, $n \leq 10$;
domyślnie: $n = 1$;
- **--in** *filename* - plik wejściowy o odpowiednim formacie, z którego ma być odczytany graf;
parametr wykluczający parametry **--size** oraz **--weight**;
domyślnie: wyłączony;
- **--out** *filename* - plik wyjściowy, do którego zostanie zapisany graf;
domyślnie: `graph.output`;
- **--connectivity** - sprawdzenie spójności grafu;
domyślnie: wyłączony;
- **--path** *v1 v2* - znalezienie najkrótszej ścieżki z wierzchołka *v1* do *v2*;
 $v1, v2 \in N$; $v1, v2$ należą do grafu;
domyślnie: wyłączony;
- **--help** - wyświetlenie skróconej instrukcji z opisem sposobu uruchamiania;
domyślnie: wyłączony.

Przykładowe wywołania programu:

- `./a.out --size 5 7 --weight 4.3 9.1 --segments 2 --out result --path 3 27`
Powyższe wywołanie generuje graf o rozmiarach 5x7 z wagami z przedziału $\langle 4.3, 9.1 \rangle$, podzielony na 2 spójne części. Graf zostanie zapisany w pliku `result`. Zostanie znaleziona najkrótsza ścieżka z wierzchołka o numerze 3 do wierzchołka o numerze 27. Wynik zostanie wypisany na `stdout`.
- `./a.out --in mygraph --connectivity`
Powyższe wywołanie odczyta graf z pliku `mygraph` oraz sprawdzi, czy jest on spójny. Wynik zostanie wypisany na `stdout`.

3 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
```

Plik reprezentujący ten graf jest przedstawiony na pierwszej stronie.

4 Dane wyjściowe

Program generuje plik wyjściowy o takim samym formacie, jak akceptowany format danych wejściowych. Umożliwia to ponowną analizę wygenerowanego grafu. Domyślnie generowany jest plik o nazwie **graph.output**.

5 Zachowania niestandardowe

W przypadku wystąpienia nieprawidłowości program zachowa się w następujący sposób:

- podanie nieistniejącego parametru (np. w postaci literówki) - program wypisuje odpowiedni komunikat o błędzie, kontynuuje działanie ignorując błędny parametr;

```
./a.out --sizee 4 5  
a.out: Error! The flag "sizee" does not exist. For further  
info please refer to the manual.
```
- podanie formatu parametru nieakceptowanego przez program - program wypisuje odpowiedni komunikat o błędzie, kontynuuje działanie przyjmując wartość domyślną danego parametru;

```
./a.out --size cztery 5.1  
a.out: Error! The flag "size" does not accept the given format  
of arguments. For further info please refer to the manual.
```
- podanie wartości argumentów poza dozwolonym zakresem - program wypisuje odpowiedni komunikat, kontynuuje działanie przyjmując wartość domyślną danego parametru;

```
./a.out --size -3 -4  
a.out: Error! The values of arguments for flag "size" are out  
of the allowed range. For further info please refer to the manual.
```
- użycie nieprawidłowego separatora dziesiętnego - program wypisuje odpowiedni komunikat, kontynuuje działanie przyjmując wartość domyślną danego parametru;

```
./a.out --weight 1,1 5,9  
a.out: Error! The only allowed decimal separator is ".". For  
further info please refer to the manual.
```
- podanie parametrów wzajemnie wykluczających się - program wypisuje odpowiedni komunikat, kontynuuje działanie na podstawie danych z pliku wejściowego;

```
./a.out --size 5 7 --weight 1.2 7.5 --in mygraph  
a.out: Error! The flags "size" + "weight" and "in" are mutually  
exclusive. For further info please refer to the manual.
```

Komunikaty o błędach zostaną wypisane na `stderr`.