

GridGraph - Specyfikacja

Skoczek Mateusz, Jędrzejewski Sebastian

29 marca 2022

Streszczenie

Dokument zawiera specyfikację funkcjonalną oraz implementacyjną dotyczącą projektu *Grid-Graph*

Spis treści

1	Specyfikacja funkcjonalna	2
1.1	Cel projektu	3
1.2	Opis funkcji	4
1.3	Opis wywołania	5
1.3.1	Tryb zapisu	5
1.3.2	Tryb czytania	6
1.4	Format danych wejściowych i wyjściowych	7
1.5	Opis błędów	8
2	Specyfikacja implementacyjna	10

Rozdział 1

Specyfikacja funkcjonalna

1.1 Cel projektu

Program `GridGraph` ma na celu wygenerowanie oraz zapis do pliku (lub na standardowe wyjście) grafu siatkowego o podanych paramentrach lub wczytanie grafu z pliku (lub ze standardowego wejścia) i sprawdzenie wybranych jego parametrów. Program działa w trybie wsadowym. Grafy są przedstawiane w plikach w postaci listy sąsiedztwa.

1.2 Opis funkcji

Program może działać w dwóch trybach: zapisu (`write`) i czytania (`read`).

W trybie zapisu program generuje graf o określonej przez użytkownika szerokości (ilości kolumn) (`width`), wysokości (ilości wierszy) (`height`), minimalnej (`edge_weight_min`) i maksymalnej (`edge_weight_max`) wagi krawędzi oraz minimalnej (`edge_count_min`) i maksymalnej (`edge_count_max`) ilości krawędzi wychodzących z jednego wierzchołka, a następnie zapisuje go w formie listy sąsiedztwa do pliku określonego przez użytkownika (lub wypisuje na standardowe wyjście).

Jeżeli graf zostanie pomyślnie zapisany do pliku (lub wypisany na standardowe wyjście), program zwróci 0. W przeciwnym wypadku zostanie wyświetlony komunikat błędu, a program zwróci 1.

W trybie czytania program wczytuje graf zapisany (w formie listy sąsiedztwa) w określonym przez użytkownika pliku (lub czyta ze standardowego wejścia), a następnie sprawdza określone przez użytkownika właściwości grafu:

- Spójność grafu (`connectivity`)
- Najkrótsza ścieżka z węzła A do innych węzłów (`shortest_path_a`) lub do określonego węzła B (`shortest_path_a` oraz `shortest_path_b`)

Jeżeli graf został wczytany oraz sprawdzony pomyślnie, zostanie wyświetlony wynik sprawdzania...

Przykład (graf spójny, ścieżka istnieje):

`Connectivity: connected`

`Shortest path from 0 to 10 (weight): 0-3-4-6-9-10 (0.778)`

Przykład (graf niespójny, ścieżka nie istnieje):

`Connectivity: disconnected`

`Shortest path from 0 to 10 (weight): path does not exist`

...a następnie program zwróci 0. W przeciwnym wypadku zostanie wyświetlony komunikat błędu, a program zwróci 1

1.3 Opis wywołania

1.3.1 Tryb zapisu

Wywołanie:

```
./gridgraph --write/-w [argumenty]
```

Argumenty:

- **--width/-xw** (Szerokość grafu - liczba kolumn)
Typ: Liczba naturalna
Zakres: > 0
Wymagany: TAK
- **--height/-xh** (Wysokość grafu - liczba wierszy)
Typ: Liczba naturalna
Zakres: > 0
Wymagany: TAK
- **--edge_weight_min/-Wmin** (Minimalna waga pojedynczej krawędzi)
Typ: Liczba rzeczywista
Zakres: $<0, \text{edge_weight_max}>$
Wymagany: NIE (domyślnie: 0)
- **--edge_weight_max/-Wmax** (Maksymalna waga pojedynczej krawędzi)
Typ: Liczba rzeczywista
Zakres: $<\text{edge_weight_min}, 1>$
Wymagany: NIE (domyślnie: 1)
- **--edge_count_min/-Cmin** (Minimalna liczba krawędzi wychodzących z jednego wierzchołka)¹
Typ: Liczba naturalna
Zakres: $<0, \text{edge_count_max}>$
Wymagany: NIE (domyślnie: 0)
- **--edge_count_max/-Cmax** (Maksymalna liczba krawędzi wychodzących z jednego wierzchołka)
Typ: Liczba naturalna
Zakres: $<\text{edge_count_min}, 4>$
Wymagany: NIE (domyślnie: 4)
- **--file/-f** (Plik w którym ma zostać zapisany graf)
Typ: Ścieżka do pliku
Zakres: -
Wymagany: NIE (domyślnie: standardowe wyjście)

¹Program będzie dążył do utworzenia co najmniej `edge_count_min` krawędzi, ale nie może tego zagwarantować. Nie jest możliwe wygenerowanie więcej niż 2 krawędzi dla wierzchołków w narożnikach oraz więcej niż 3 dla wierzchołków bocznych. Nie jest możliwe także utworzenie krawędzi, jeżeli wszystkie wierzchołki wokół osiągnęły już swoją nominalną (wylosowaną z podanego przedziału) liczbę krawędzi.

Przykład:

```
./gridgraph -w -xw 6 -xh 6 --Wmin 0.65 --Wmax 0.2 -Cmax 3 -f "/home/user/graph"
```

Powyższy przykład ilustruje wywołanie programu, który generuje graf o 6 kolumnach i 6 wierszach, z wagami krawędzi mieszczącymi się w przedziale od 0.2 do 0.65, gdzie minimalna ilość krawędzi wychodzących z wierzchołka to 0, a maksymalna ilość krawędzi to 3. Program zapisuje graf w odpowiednim formacie do pliku o nazwie **graph** znajdującego się w **/home/user**.

1.3.2 Tryb czytania

Wywołanie:

```
./gridgraph --read/-r [argumenty]
```

Argumenty:

- **--connectivity/-c** (Sprawdza czy graf jest spójny, używając algorytmu BFS)
Typ: -
Zakres: -
Wymagany: NIE²
- **--shortest_path_a/-Sa** (Znajduje najkrótszą ścieżkę od wierzchołka A do pozostałych wierzchołków, używając algorytmu Dijkstry)
Typ: Liczba naturalna
Zakres: <0, ilość wierzchołków grafu>
Wymagany: NIE^{2 3}
- **--shortest_path_b/-Sb** (Znajduje najkrótszą ścieżkę od wierzchołka A do wierzchołka B, używając algorytmu Dijkstry)
Typ: Liczba naturalna
Zakres: <0, ilość wierzchołków grafu> / **shortest_path_a**
Wymagany: NIE²
- **--file/-f** (Plik z którego ma zostać wczytany graf)
Typ: Ścieżka do pliku
Zakres: -
Wymagany: NIE (domyślnie: standardowe wejście)

Przykład:

```
./gridgraph -r -c -Sa 0 -Sb 10 -f \home/user/graph"
```

Powyższy przykład ilustruje wywołanie programu, który czyta plik ze strukturą grafu o nazwie **graph** znajdujący się w **/home/user**, a następnie sprawdza czy ten graf jest spójny oraz wyznacza najkrótszą ścieżkę pomiędzy węzłami numer 0 i 10.

²Wymagany przynajmniej jeden

³Wymagane jeżeli **shortest_path_b** zostało zdefiniowane

1.4 Format danych wejściowych i wyjściowych

Dane wejściowe i wyjściowe przechowują graf w postaci listy sąsiedztwa. W pierwszej linii znajdują się dwie liczby, które oznaczają odpowiednio liczbę kolumn i wierszy danego grafu. Każda następna linijka reprezentuje jeden wierzchołek, przy czym wierzchołki numerujemy od 0 od lewej do prawej. Zatem druga linijka w pliku zawiera numery wierzchołków, z którymi połączony jest wierzchołek numer 0, kolejna dotyczy wierzchołka numer 1 itd. Przy każdym numerze wierzchołka po dwukropku podana jest waga krawędzi pomiędzy tymi dwoma wierzchołkami.

Przykład:

```
2 2
1 :0.54  2 :0.78
0 :0.54  3 :0.12
0 :0.78  3 :0.89
1 :0.12  2 :0.89
```

Powyżej przedstawiona jest przykładowa zawartość pliku przechowującego graf. W pierwszej linii można odczytać, że jest to graf o dwóch kolumnach i dwóch wierszach. W drugiej linii przedstawiona jest informacja o tym, że wierzchołek numer 0 połączony jest z wierzchołkiem numer 1, a krawędź ta ma wagę 0.54. Istnieje również krawędź pomiędzy wierzchołkiem 0 a 2 o wadze 0.78. W trzeciej linii znajdują się numery wierzchołków połączonych z wierzchołkiem numer 1 wraz z wagami itd.

1.5 Opis błędów

W przypadku błędu program wypisuje błąd na standardowy strumień błędów i zwraca 1. Komunikat błędów jest poprzedzony słowem **ERROR** oraz nazwą trybu w nawiasie (np. (**Write mode**)), jeżeli błąd dotyczy konkretnego trybu.

Poniżej przedstawione są komunikaty generowane przez program, gdy ten wykryje błąd, wraz z ich wyjaśnieniem:

WIDTH_NOT_POSITIVE_NUMBER Został wybrany argument **width**, ale nie została podana wartość lub wartość nie jest liczbą.

HEIGHT_NOT_POSITIVE_NUMBER Został wybrany argument **height**, ale nie została podana wartość lub wartość nie jest liczbą.

EDGE_WEIGHT_MIN_NOT_POSITIVE_NUMBER Został wybrany argument **edge_weight_min**, ale nie została podana wartość lub wartość nie jest liczbą (nieujemną).

EDGE_WEIGHT_MAX_NOT_POSITIVE_NUMBER Został wybrany argument **edge_weight_max**, ale nie została podana wartość lub wartość nie jest liczbą (nieujemną).

EDGE_COUNT_MIN_NOT_POSITIVE_NUMBER Został wybrany argument **edge_count_min**, ale nie została podana wartość lub wartość nie jest liczbą (nieujemną).

EDGE_COUNT_MAX_NOT_POSITIVE_NUMBER Został wybrany argument **edge_count_max**, ale nie została podana wartość lub wartość nie jest liczbą (nieujemną).

WIDTH_LOWER_OR_EQUAL_TO_ZERO Wartość argumentu **width** jest mniejsza lub równa 0 (musi być większa od 0).

HEIGHT_LOWER_OR_EQUAL_TO_ZERO Wartość argumentu **height** jest mniejsza lub równa 0 (musi być większa od 0).

EDGE_WEIGHT_MIN_LOWER_THAN_ZERO Wartość argumentu **edge_weight_min** jest mniejsza od 0 (musi być większa lub równa 0 i mniejsza lub równa **edge_weight_max**).

EDGE_WEIGHT_MAX_GREATER_THAN_ONE Wartość argumentu **edge_weight_max** jest większa od 1 (musi być mniejsza lub równa 1 i większa lub równa **edge_weight_min**).

EDGE_WEIGHT_MIN_GREATER_THAN_EDGE_WEIGHT_MAX Wartość argumentu **edge_weight_min** jest większa od **edge_weight_max** (musi być większa lub równa 0 i mniejsza lub równa **edge_weight_max**).

EDGE_COUNT_MIN_LOWER_THAN_ZERO Wartość argumentu **edge_count_min** jest mniejsza od 0 (musi być większa lub równa 0 i mniejsza lub równa **edge_count_max**).

EDGE_COUNT_MAX_GREATER_THAN_FOUR Wartość argumentu **edge_count_max** jest większa od 4 (musi być mniejsza lub równa 4 i większa lub równa **edge_count_min**).

EDGE_COUNT_MIN_GREATER_THAN_EDGE_COUNT_MAX Wartość argumentu **edge_count_min** jest większa od **edge_count_max** (musi być większa lub równa 0 i mniejsza lub równa **edge_count_max**).

SHORTEST_PATH_A_NOT_POSITIVE_NUMBER Został wybrany argument **shortest_path_a**, ale nie została podana wartość lub wartość nie jest liczbą (nieujemną).

SHORTEST_PATH_B_NOT_POSITIVE_NUMBER Został wybrany argument **shortest_path_b**, ale nie została podana wartość lub wartość nie jest liczbą (nieujemną).

SHORTEST_PATH_B_WITHOUT_SHORTEST_PATH_A_SPECIFIED Został wybrany argument `shortest_path_a`, ale nie został wybrany argument `shortest_path_b`.

SHORTEST_PATH_B_EQUAL_TO_SHORTEST_PATH_A Argument `shortest_path_b` jest równy `shortest_path_a` (wartości argumentów muszą być różne od siebie).

CHECKING_OPTIONS_NOT_SPECIFIED Nie została wybrana przynajmniej jedna opcja sprawdzająca (przynajmniej jedna wymagana).

SHORTEST_PATH_A_GREATER_THAN_TOTAL_NUMBER_OF_VERICES Wartość argumentu `shortest_path_a` jest większa niż całkowita liczba wierzchołków grafu.

SHORTEST_PATH_B_GREATER_THAN_TOTAL_NUMBER_OF_VERICES Wartość argumentu `shortest_path_b` jest większa niż całkowita liczba wierzchołków grafu.

Rozdział 2

Specyfikacja implementacyjna