

Specyfikacja funkcjonalna programu grapher

Szymon Półtorak i Sebastian Sikorski

10.03.2022r

Streszczenie

W dokumencie zawarte są informacje na temat projektu dotyczącego grafów, napisanego w języku *C*. Pokazane jest, jak uruchomić program oraz jakie są jego opcje. Ponadto dokument opisuje i tłumaczy kody błędów występujące w programie.

Spis treści

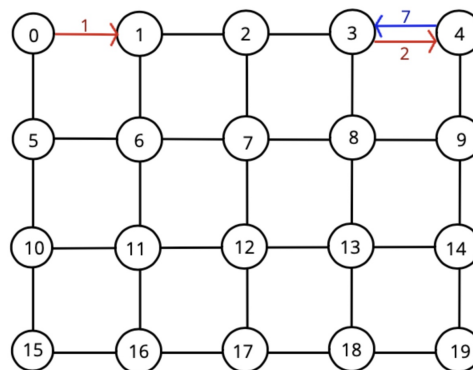
1	Cel Projektu	2
2	Struktura głównego folderu	3
3	Wywołanie Programu	3
	3.1 Składnia dla trybów Wage, Edge oraz Random	3
	3.2 Składnia dla trybu Read	3
	3.3 Format pliku dla trybu Read Mode	4
	3.4 Argumenty wywołania programu	4
	3.5 Przykładowe wywołanie programu	5
4	Wyjście programu	6
5	Obsługiwane błędy	6

1 Cel Projektu

Celem projektu jest stworzenie programu mającego za zadanie generowanie grafów, sprawdzanie ich spójności oraz wyszukiwanie w nich najkrótszej ścieżki między zadanymi punktami. Program oferuje trzy różne tryby posiadające różne funkcje.

- Wage Mode – program generuje graf o losowych wagach dróg między wierzchołkami w taki sposób, że jest on spójny,
- Edge Mode – program losuje istnienie krawędzi między wierzchołkami grafu oraz wagi do momentu powstania grafu spójnego. Do sprawdzania wykorzystuje algorytm BFS,
- Random Mode – program losuje wagi dróg oraz krawędzie między wierzchołkami. W tym trybie graf może być niespójny,
- Read Mode – program odczytuje odpowiednio sformatowany plik i szuka najkrótszej ścieżki między podanymi przez użytkownika punktami za pomocą algorytmu Dijkstry. Format pliku zostanie opisany w osobnej podsekcji, sekcji trzeciej.

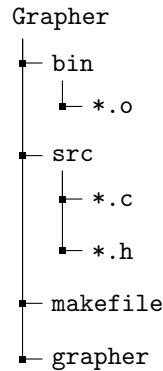
Struktura grafu oparta jest na koncepcji "kratka w kratkę" tzn. graf składa się z wierzchołków równo rozmieszczonych na liniach poziomych, i pionowych wyznaczanych przez liczbę wierszy i kolumn. Jedyne połączenia zachodzące między wierzchołkami dozwolone są pionowo i poziomo co pokazuje poniższy diagram, na którym zostały zaznaczone jedynie wagi wybranych krawędzi aby zachować czytelność całego diagramu, jednocześnie obrazując schemat połączeń.



Rysunek 1: Przykład grafu typu "kratka w kratkę"

2 Struktura głównego folderu

Projekt składa się z jednego folderu głównego *Grapher*, podfolderu *src* oraz pliku *makefile*. Plik *makefile* odpowiada za pół-automatyczną kompilację całego projektu, natomiast w folderze *src* są pliki **.c* z kodem źródłowym oraz pliki nagłówkowe **.h*. Skompilowanie programu polega na użyciu komendy *make*. Po jej użyciu program będzie posiadał następującą strukturę:



W folderze *bin* będą przechowywane pliki maszynowe **.o*, a plik *grapher* będzie odpowiedzialny za uruchomienie programu.

3 Wywołanie Programu

Program stworzony pod pracę z konsolą systemu Linux (przykładowa dystrybucja: Debian 11.0).

3.1 Składnia dla trybów Wage, Edge oraz Random

Program uruchamiamy za pomocą komendy:

```
./grapher [tryb] [Plik] [Wiersze] [Kolumny] [Początek] [Koniec]
```

Niezbędne do uruchomienia programu jest podanie wszystkich argumentów wywołania.

3.2 Składnia dla trybu Read

```
./grapher [tryb] [Plik] [Flaga] [punkt1] [punkt2] ... [punkt(n)] [punkt(n+1)]
```

Niezbędne do uruchomienia programu jest podanie przynajmniej pierwszych 5 argumentów wywołania nie licząc nazwy programu.

Poszczególne elementy składni wszystkich trybów wyjaśniamy w podrozdziale 3.4.

3.3 Format pliku dla trybu Read Mode

Program do działania w trybie Read Mode przyjmuje plik o określonych właściwościach:

- W pierwszym wierszu pliku znajduje się informacja o liczbie wierszy i kolumn jakie składają się na graf,
- W każdym następnym wierszu znajdują się informacje o tym z jakimi innymi wierzchołkami połączony jest dany wierzchołek oraz waga jaka odpowiada temu połączeniu.

Ze względu na numerowanie wierzchołków od zera, numer wiersza odpowiada numerowi wierzchołka zwiększonego o jeden.

Przykładowa zawartość pliku:

```
3 3
1 :0.33  3 :2.32
2 :3.21
1 :5.11  5 :2.46
0 :0.89  4 :3.23  6 :2.21
1 :1.23  3 :3.27  5 :2.25  7:5.12
4 :2.33
3 :3.63  7 :1.22
6 :6.21  4 :1.34
5 :4.26  7 :8.1
```

3.4 Argumenty wywołania programu

Poniżej wyjaśniamy wszystkie potrzebne argumenty wywołania programu:

1. [Tryb]:
Działanie poszczególnych trybów opisane są w celu projektu, flagi do poszczególnych trybów:
 - *-WM* – Wage Mode
 - *-EM* – Edge Mode
 - *-RM* – Random Mode
 - *-ReM* – Read Mode
2. [Plik]:
Plik, do którego zostanie zapisany wygenerowany przez program graf, plik będzie zawsze nadpisywany nową zawartością. W trybie *Read Mode* będzie to plik do czytania.
3. [Wiersze]:
Liczba wierszy jakie zostaną wygenerowane w grafie. Liczba ta musi być większa od zera.
4. [Kolumny]:
Liczba kolumn jakie zostaną wygenerowane w grafie. Liczba ta musi być większa od zera.
5. [Początek]:
Początek przedziału z jakiego będą generowane wagi dla krawędzi między wierzchołkami. Musi to być wartość większa od 0.
6. [Koniec]:
Koniec przedziału z jakiego będą generowane wagi dla krawędzi między wierzchołkami. Musi to być wartość większa od 0.

7. [Flaga]:

Flagi mogą być złożone zarówno z dużych jak i małych liter. Wymaganą flagę wybieramy spośród niżej wymienionych:

- **-standard** – flaga pozwalająca na wyświetlenie skróconej wersji najkrótszej ścieżki między dwoma zadanymi punktami.

Format wyświetlania: (od,do); od → następny punkt → ... → do

np.

(7,8); 7 → 6 → 5 → 9 → 8,

- **-extended** – flaga pozwalająca na wyświetlenie rozszerzonej wersji najkrótszej ścieżki między dwoma zadanymi punktami.

Format wyświetlania: (od,do); (od,do); od(waga przejścia) → następny punkt(waga przejścia) → ... → do

np.

(7,8); 7(0,4) → 6(0,2) → 5(0,3) → 9(0,1) → 8.

8. [Punkt(n)]:

Punkt **od**, którego szukamy drogę w grafie.

9. [Punkt(n+1)]:

Punkt **do**, którego szukamy drogę w grafie.

3.5 Przykładowe wywołanie programu

1. Wywołanie dla składni trybów Wage, Edge, Random

Poniżej prezentujemy przykładową komendę wywołania dla trybu Wage mode z poprawnymi danymi. Wybraliśmy tryb Wage Mode, podaliśmy plik, liczbę wierszy i kolumn oraz zakres losowania wag.

np.

```
./grapher -WM mygraph 3 5 1 6
```

Wywołanie dla złych danych, tutaj nie podamy pliku:

np.

```
./grapher -WM
```

Program udzieli informację o braku podania pliku oraz wyświetli poprawną składnię.

2. Wywołanie dla składni trybu Read

Poniżej podaliśmy poprawną komendę wywołania dla trybu Read. Wybraliśmy tryb, podaliśmy plik, flagę oraz podaliśmy punkty od i do, którego szukamy najkrótszej ścieżki.

```
np.  
./grapher -ReM mygraph -standard 7 8
```

Wynikiem działania programu będzie wyświetlenie najkrótszej ścieżki.
Tutaj przedstawiamy przykład wywołanie programu bez flagi:

```
np.  
./grapher -ReM mygraph
```

Program wyświetli komunikat o braku flagi i składnię programu.

4 Wyjście programu

Program zapisuje strukturę wygenerowanego grafu do pliku oraz wypisuje na konsolę informacje o wyniku swojego działania.

W zależności od trybu na konsoli zostanie wyświetlona informacja:

- Wage Mode – o pomyślnym wygenerowaniu grafu,
- Edge Mode – o wygenerowaniu grafu spójnego jeśli uda się to wykonać poniżej 500 prób, w przeciwnym wypadku program zapyta użytkownika czy chce kontynuować losowanie krawędzi i wag,
- Random Mode – o tym czy graf jest spójny, a następnie poda plik do którego go zapisał,
- Read Mode – o spójności grafu i w przypadku odczytania grafu spójnego o najkrótszej ścieżce między zadanymi punktami.

Wszelkie komunikaty o błędach zostają wypisane na konsolę, wraz z instrukcją dotyczącą składni programu.

5 Obsługiwane błędy

W poniższej tabeli wypisaliśmy błędy jakie mogą wystąpić w czasie działania programu oraz wyjaśniliśmy co znaczą.

Nazwa Błędu	Kod	Wyjaśnienie błędu
NO_MODE_FOUND	501	Niepoprawny tryb lub jego brak
NO_FILE_FOUND	502	Nie podano pliku lub plik nie istnieje
WRONG_NUM_OF_ROWS	503	Podano niepoprawną liczbę wierszy
WRONG_NUM_OF_COL	504	Podano niepoprawną liczbę kolumn
WRONG_RANGE_OF_WAGES	505	Zły zakres losowania wartości wag
NO_FLAG_FOUND	611	Nie podany flagi w trybie Read Mode
WRONG_POINTS	612	Podano nieistniejący punkt lub ich złą liczbę
NO_COHERENT	613	Graf jest niespójny