Specyfikacja funkcjonalna programu tworzącego i badającego grafy

Wojciech Majchrzak, Dawid Stereńczak

March 2022

1 Cel projektu

Program służy do tworzenia grafów i zapisywania ich do plików oraz przetwarzania grafów odczytanych z plików w tym: badania spójności grafu oraz szukania najkrótszych ścieżek miedzy wskazanymi wcześniej wierzchołkami. Program działa w trybie nieinteraktywnym/wsadowym i jest obsługiwany za pomocą argumentów podczas wywoływania programu.

2 Funkcje programu

Program zawiera poniższe funkcje dostępne w odpowiadających trybach, które użytkownik wybiera za pomocą odpowiednich komend opisanych w sekcji "4. Sterowanie programem":

- Tworzenie grafu o podanych przez użytkownika rozmiarach, właściwościach i zapisywanie stworzonego wcześniej grafu do pliku tekstowego - tryb `generate`
- Odczyt grafu zapisanego w postaci pliku tekstowego i sprawdzanie spójności grafu - tryb `check`
- Odczyt grafu zapisanego w postaci pliku tekstowego i obliczanie długości ścieżki między podanymi węzłami w grafie - tryb `path`

3 Dane wejściowe

Plik wejściowy, potrzebny w trybach `check i path`, oraz wyjściowy w trybie `generate` mają taki sam format. Pliki zawierają następujące informacje o grafie:

- liczba wierszy(w) - podana w pierwszym wierszu pliku, w > 0;
- liczba kolumn(k) - oddzielona spacją od w, k > 0;
- numery wierzchołków, z którymi dany wierzchołek jest połączony - zawierają się w przedziale od 1 do w*k;
- wagi połączeń - waga > 0;

W trybie `generate`, parametry grafu są ustalane poprzez argumenty wywołania programu.

Przykładowe dane wejściowe/wyjściowe o poprawnym formacie:
2 2
2 :0.88 3 :0.21
1 :0.88 4 :0.64
1 :0.86 4 :0.42
2 :0.64 3 :0.42

W pierwszym wierszu powyższych danych znajdują się, oddzielone spacją, liczba wierszy oraz liczba kolumn. W kolejnych wierszach są zdefiniowane połączenia między wierzchołkami oraz wagi tych połączeń. W drugim wierszu są zdefiniowane połączenia dla wierzchołka numer 1, w trzecim wierszu dla wierzchołka 2, itd. Program wczytuje kolejne wiersze aż do wiersza o numerze `( liczba wierszy * liczba kolumn ) + 1`.

Schemat definiowania połączeń jest następujący:
`<w1> :<waga 1> <w 2> :<waga 2> ... <w n> :<waga n>`
gdzie:
w1, w2, ..., wn - numery kolejnych wierzchołków, z którymi jest połączony wierzchołek definiowany w danym wierszu.
waga1...n - wagi krawędzi łączących wierzchołek definiowany w danym wierszu z wierzchołkiem podanym przed znakiem ":".

Przedstawione dane odwzorowują więc graf o wymiarach 2x2 (2 wiersze x 2 kolumny). Wierzchołek 1 jest połączony z wierzchołkiem 2 - waga krawędzi 0.88 oraz z wierzchołkiem 3 - waga 0.21, wierzchołek 2 jest połączony z wierzchołkiem 1 - waga 0.88 oraz z wierzchołkiem 4 - waga 0.64, itd. Uwaga: waga przejścia z wierzchołka a do b powinna być taka sama jak z b do a, ponieważ graf jest nieskierowany, więc jest to to samo przejście.

4 Sterowanie programem

Sterowanie programem wykonuje się za pomocą następujących flag:

- `-m [generate | check | path]` określa operacją którą program ma wykonać: `generate` - generuje graf o wymiarach podanych w kolejnych flagach, `check` - sprawdza czy wczytany graf jest spójny, `path` - oblicza długość ścieżki między podanymi węzłami w grafie. W przypadku braku wyboru trybu domyślnym trybem jest `generate`.

- `-f [filename]` określa nazwę pliku na którym program będzie operował, domyślnie `graph.txt`.

- `-h` jest to flaga która wyświetla instrukcję użytkowania programu, a następnie kończy jego działanie. Nie wymaga użycia innych argumentów.

Dodatkowe argumenty są indywidualnie określone dla poszczególnych trybów programu w podsekcjach tego akapitu.

4.1 generate

Dla trybu `generate` dostępne są następujące flagi:

- `-s [width x length]` wymiar generowanego grafu, podany w formacie: wysokość(liczba naturalna) x szerokość(liczba naturalna) - bez spacji np. `21x15`. Domyślne wymiary grafu to `100x100`. Maksymalny wymiar grafu tj. wysokość * szerokość = 10 ^ 8.

- `-n [amount]` ilość podgrafów na które ma zostać podzielony graf w postaci liczby naturalnej, domyślnie 0.

- `-r [range_start-range_end]` zakres wag krawędzi występujących w generowanym grafie, podawany w formacie: początek_przedziału(liczba rzeczywista dodatnia z dokładnością do 2. miejsc po przecinku)-koniec_przedziału(liczba rzeczywista dodatnia) - bez spacji np. `1;5`. Domyślny przedział to <0-10>. Maksymalna górna granica przedziału wynosi 10 ^ 4. Należy pamiętać że górna granica przedziału powinna być większa lub równa dolnej granicy przedziału. W innym przypadku program zwróci błąd. W przypadku liczb niecałkowitych należy użyć kropki jako separatora dziesiętnego.

4.2 path

Dla trybu `path` dostępne są następujące flagi:

- `-a [x-coordinate,y-coordinate]` współrzędne pierwszego punktu do mierzenia odległości w formacie x(liczba naturalna),y(liczba naturalna) - bez spacji np. 2,3. Flaga nie posiada domyślnej wartości. Wartość maksymalna jest równa wymiarowi przeszukiwanego grafu.
- `-b [x-coordinate,y-coordinate]` współrzędne drugiego punktu do mierzenia odległości. Format oraz właściwości takie jak punkt wyżej.

4.3 check

Dla trybu `check` nie są przewidziane żadne dodatkowe flagi.

5 Obsługa programu

W tym rozdziale opisane zostaną przykładowe uruchomienia programu dla poszczególnych trybów

- tryb `generate` przykład użycia:
  `./nazwa_programu -m generate -f plik.txt -s 34x76 -r 5-21` - w tym przypadku program wygeneruje graf o wysokości 34 i szerokości 76 wierzchołków. Wagi krawędzi między wierzchołkami będą w przedziale <5-21>. Graf będzie niepodzielony na mniejsze podgrafy. Następnie wygenerowany graf zostanie zapisany w odpowiednim formacie do pliku plik.txt.
- tryb `check` przykład użycia:
  `./nazwa_programu -m check -f plik_testowy.txt` - w tym przypadku program sprawdzi spójność grafu zapisanego w pliku plik_testowy.txt, a następnie poinformuje użytkownika w postaci komunikatu w konsoli o tym czy graf jest spójny.
- tryb `path` przykład użycia:
  `./nazwa_programu -m path -f plik_testowy.txt -a 2,1 -b 7,6` - w tym przypadku program obliczy najkrótszą drogę między wierzchołkami (2,1) oraz (7,6) z grafu zapisanego w pliku plik_testowy.txt, a następnie wypisze najkrótszą drogę w konsoli. Jeżeli droga między podanymi wierzchołkami nie istnieje program poinformuje o tym użytkownika.

6 Obsługa błędów w programie

W przypadku podania błędnych flag program może zareagować na następujące sposoby:

- Gdy podane jest zbyt mało flag program użyje wartości domyślnych ustawionych dla brakujących argumentów.
- Gdy podane zostaną zbędne argumenty które nie kolidują z pracą programu , zostaną one zignorowane.
- Jeżeli program nie jest w stanie otworzyć do zapisu/odczytu podanego przez użytkownika pliku, program zwróci kod błędu 2 oraz wyświetli stosowny opis błędu.
- W przypadku gdy program nie będzie w stanie wykonać oczekiwanej operacji przerwie działanie i zwróci kod błędu 3 oraz wyświetli opis błędu który napotkał(np. badanie odległości węzłów w przypadku niespójnego grafu).
- Jeżeli dane w pliku podanym przez użytkownika są błędne to program zwróci kod błędu 4 i wyświetli odpowiedni opis błędu.
- W przypadku podania niewłaściwego przedziału wag krawędzi generowanego grafu w trybie `generate`, program zwróci kod błędu 5 i wyświetli opis błędu.
- Jeżeli poszczególne argumenty uruchomiania programu zostaną podane w innym formacie niż określone w sekcji "4 Sterowanie programem", program może zachować się w sposób niekontrolowany np. wyświetlić błąd `Segmentation fault (core dumped`.
- Gdy użytkownik będzie próbował obliczyć drogę między nieistniejącymi wierzchołkami program zwróci kod błędu 6 i wyświetli opis błędu.