Politechnika Śląska
Wydział Informatyki, Elektroniki i Informatyki

Programowanie Komputerów 3

Agorytm Dijkstry

autor	Michał Ślusarczyk
prowadzący	mgr. Grzegorz Kwiatkowski
rok akademicki	2019/2020
kierunek	Informatyka
rodzaj studiów	SSI
semestr	3
sekcja	2
termin oddania sprawozdania	2020-11-08

1 Treść zadania

Opis ogólny:
Napisać program generujący dla wskazanych przez użytkownika wierzchołków grafu tablice ze zbiorem decyzji określające przez które z sąsiednich wierzchołków grafu prowadzą najlepsze trasy do wybranych wybranych przez użytkownika wierzchołków oraz podając koszty (metryki) tych tras. W celu realizacji programu skorzystać z algorytmu Dijkstry.

Szczegółowe informacje na temat realizacji:
Opis grafu składa się z numerów wierzchołków, numerów sąsiadujących wierzchołków oraz metryk krawędzi grafu.
Opis podawany jest przez użytkownika za pomocą interfejsu który przeprowadza użytkownika przez proces konstrukcji grafu oraz określenia parametrów. Możliwe jest również wczytanie tych informacji z dwóch różnych plików. Możliwe są również warianty pośrednie (część informacji wczytanych z plików, a część z konsoli).

Postać informacji na temat grafu:
<nr. wierzchołka>: <nr. wierzchołka-sąsiada> <metryka>; …

Postać informacji na temat tego jakie przejścia chcemy wyznaczyć:
<nr. wierzchołka>: <nr. wierzchołka-sąsiada>; …

Postać informacji o rozwiązaniu powinna wyglądać w następujący sposób:
<dla wierzchołka>:
<wierzchołek docelowy>: → <przez wierzchołek-sąsiada> : <koszt trasy>

Metryki między wierzchołkami mogą się różnić w przeciwnych kierunkach.
Założenie o braku metryki z wierzchołka A do A (wartość zostanie zignorowana),
Program powinien dopytywać się użytkownika o brakujące informacje w opisie grafu.

Wynik programu:
Opis grafu oraz wynik programu powinien zostać wyświetlony w interfejsie oraz zapisany w pliku o rozszerzeniu .txt wskazanym przez użytkownika (jeśli nie istnieje powinien zostać utworzony). Program powinien być uruchamiany z wykorzystaniem konsoli.

Interfejs:
Interfejsem programu wykorzystywanym do komunikacji z użytkownikiem jest konsola. W konsoli powinny wyświetlać się odpowiednie polecenia oraz pytania na które odpowiadać będzie użytkownik. Na tej podstawie program wyświetla opis grafu oraz wynik swojego działania.

Analiza zadania

Zagadnienie przedstawia zagadnienie wyznaczania optymalnej drogi z punktu A do punktu B w określonym układzie.

5.1 Struktury danych

W programie wykorzystano kilka różnych struktur danych które są ze sobą ściśle zwiazane . Pozwalają one na zapis danych w postaci list.

5.2 Algorytmy

Program opiera się na algorytmie Dijkstry. Oprócz tego zastosowane zostało proste sortowanie przy wstawianiu co pozwoliło na automatyczne uszeregowanie danych.

6 Specyfikacja zewnętrzna

Program jest uruchamiany z linii poleceń. Należy przekazać do programu nazwy plików: dwóch wejściowych oraz wyjściowego po odpowiednich przełącznikach -G, -D oraz -R, np:

program.exe –G GRAPH.txt, -D DATA.txt, -R SOLUTION.txt

Pliki posiadają rozszerzenie tekstowe.

Podanie nieprawidłowej nazwy pliku powoduje wyświetlenie zwrócenie błędu i przerwanie programu .

Podanie danych w nieodpowiedniej formule powoduje zwrócenie błędu i przerwanie programu .

4 Specyfikacja wewnętrzna

Program został zrealizowany zgodnie z paradygmatem strukturalnym. W programie rozdzielono interfejs (komunikację z użytkownikiem) od logiki aplikacji (realizacja układu).

4.1 Ogólna struktura programu

W funkcji głównej main po spełnieniu warunków związanych z poprawnym uruchomieniem programu w konsoli, wywoływany jest konstruktor tej klasy w którym uruchamiana jest metoda w której tworzone są obiekty potrzebne do obsługi określonych części programu.

Program podzielony został na etapy:
- User – Komunikacja z użytkownikiem i pozyskanie dokładnych informacji co ma wykonać program,

- Create – Wczytanie lub stworzenie danych wejściowych i grafu na których operuje program,

- Algorithm – realizacja algorytmy Dijkstry czyli stworzenie rozwiązania,

- Save – zapis danych wyjściowych do pliku wynikowego,

- Print – wyświetlenie danych na ekranie,

4.2 Szczegółowy opis typów i funkcji

Szczegółowy opis typów i funkcji zawarty jest w załączniku.

5 Testowanie

Program został przetestowany na różnego rodzaju plikach.Przy braku pliku wejściowego o podanej nazwie zwracany jest odpowiedni komunikat. Niepoprawny format danych powoduje przerwanie programu oraz wyświetlenie odpowiedniego komunikatu. Program został sprawdzony pod kątem wycieków pamięci.

6 Wnioski

Program realizujący algorytm Dijkstry wymagał wiedzy z zakresu zrządzania pamięcią w sposób dynamiczny, umiejętności zapisu i odczytu z pliku, radzenia sobie z zagnieżdżoną strukturą danych oraz sprawdzał świadomość piszącego na temat jego programu. Najtrudniejszym elementem w mojej opinii było zinterpretowanie polecania w sposób umożliwiający jego realizacje za pomocą kodu.

Osobiście uważam że sam projekt był wielce pouczający i pozwolił w sposób kreatywny wykorzystać dotychczas zdobytą wiedzę oraz uzupełnić ewentualne braki. Szczególnie kładąc nacisk na zarządzanie architekturą projektu w ten sposób aby był on przejrzysty i jego elementy ze sobą nie kolidowały szczególnie biorąc pod uwagę mechanizm dziedziczenia oraz obiektów składowych. Wiedza którą pozyskałem w zakresie synchronizacji elementów składowych programu przełoży się z pewnością na następne projekty.

Literatura

Jerzy Grębosz "Symfonia C++ standard", Tom 1 i 2

[1]http://sjp.pwn.pl
[2]http://sjp.pwn.pl/zasady
[3]https://pl.wikipedia.org/wiki/Pomoc:Powszechne_błędy_językowe

Dodatek
Szczegółowy opis typów i funkcji