

Sprawozdanie 3

Rafał Dzumaga

Karol Mruk

Program

Program jest aplikacją konsolową napisaną w języku C#. Jego zadaniem jest wyznaczanie cykli Eulera w grafach Eulera przy pomocy algorytmu Fleuri'ego. Ze względu na specyfikę zadania nie jest sprawdzane czy algorytm zwrócił poprawną ścieżkę w przypadku podania na wejściu grafu nie Eulerowskiego. Zgodnie z definicją każdy graf Eulerowski musi posiadać ścieżkę Eulera.

Źródłem danych dla programu jest plik tekstowy którego format został opisany w instrukcji obsługi. Wynikiem działania programu jest lista krawędzi wypisywanych w kolejnych wierszach na standardowe wyjście aplikacji które składają się na cykl Eulera. Wynik nie jest zapisywany do pliku co jednak nie jest problemem gdyż w łatwy sposób można przekierować standardowe wyjście do pliku korzystając z narzędzi systemowych.

Ze względu na swoją specyfikę i język który został użyty do implementacji program może być uruchamiany na komputerach z zainstalowanym systemem Windows.

Całość kodu wraz z dokumentacją znajduje się w repozytorium git pod adresem: [git@github.com:komatta91/GIS.git](https://github.com/komatta91/GIS.git).

Opis kodu

Dokumentacja kodu została przygotowana w aplikacji Doxygen w postaci strony internetowej i została dołączona do niniejszej dokumentacji w katalogu „dokumentacja”.

Instrukcja obsługi

Program jest aplikacją konsolową. Źródłem danych dla programu jest plik tekstowy którego nazwa powinna zostać przekazana jako parametr wywołania. Jeśli nazwa nie zostanie przekazana w formie parametru zostanie użyta domyślna nazwa „GraphData.txt”. Plik ten musi znajdować się w tym samym katalogu co główna aplikacja. W przypadku braku pliku aplikacja zgłosi błąd.

W przypadku gdy przekazujemy plik w postaci parametru, nie korzystając z nazwy domyślnej możemy przekazać dowolną ścieżkę jeśli tylko możliwe będzie odczytanie zawartości ze względu na uprawnienia systemowe.

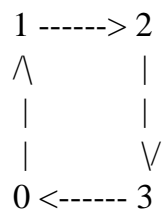
Plik wejściowy zawiera graf w postaci listy sąsiedztwa. W kolejnych wierszach znajdują się opisane wierzchołki ponumerowane kolejno poczynając od zera.

Format każdej linii jest następujący. Na pierwszej pozycji znajduje się numer wierzchołka którego dotyczy w następnych pozycjach oddzielonych spacją są wierzchołki połączone krawędzią skierowaną wychodzącą z wierzchołka podanego na pierwszym miejscu danej listy.

Przykładowy plik wejściowy dla prostego grafu o 4 wierzchołkach i 4 krawędziach:

0,1
1,2
2,3
3,0

Graf taki wygląda następująco:



Testy

Testy poprawności

W formie przeprowadzania testów poprawności posłużyliśmy się kilkoma przykładami które zostały stworzone oraz wygenerowane specjalnie przez nas stworzonym generatorem i ręcznie sprawdziliśmy czy program daje poprawny wynik.

Przykłady:

Plik wejściowy	Wynik
0,1 1,2 2,3 3,1	0-1 1-2 2-3 3-1 --- VISITED: 4 OF 4 EDGES ---
0,1,1,0,0,0,1 1,0,1,0,0	0-1 1-0 0-1 1-1 1-0 0-0 0-0 0-0 0-1

	1-0 --- VISITED: 10 OF 10 EDGES ---
0,2,2,1,0,1,1,1 1,0,1,0,0,0 2,0,2,0	0-2 2-0 0-2 2-2 2-0 0-1 1-0 0-0 0-1 1-1 1-0 0-1 1-0 0-1 1-0 --- VISITED: 15 OF 15 EDGES ---

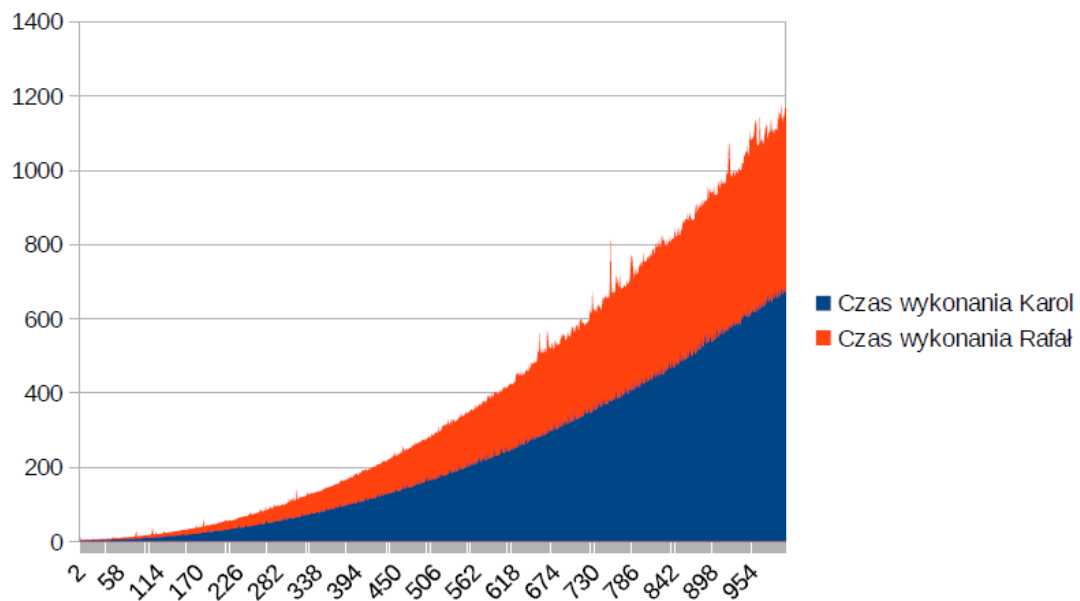
Jak widać dla powyższych przykładów program zwraca poprawne wyniki.

Testy wydajności

Na potrzeby badania wydajności aplikacji potrzebowaliśmy wielu grafów o różnej liczbie krawędzi. W celu ich pozyskania został stworzony specjalny generator. Został on napisany w języku C++.

Badanie wydajności przebiegało w ten sposób że generowaliśmy kolejno coraz większe grafy przy pomocy generatora i wynik jego działania był przekazywany do naszej aplikacji. Aplikacja zapisywała czasy działania w pliku log.txt. W ten sposób pozyskane wyniki zostały zagregowane w arkuszu kalkulacyjnym.

Wydajność była badana niezależnie na dwóch komputerach. Wyniki z obu komputerów prezentuje poniższy wykres.



Jak widać na powyższym wykresie szacowana przez nas teoretyczna złożoność $O(E^2)$ pokrywa się z wynikami rzeczywistej implementacji w obu tych przypadkach. Różnica w czasach na poszczególnych maszynach wynika z ogólnej różnicy wydajności maszyn.

Szczegółowe wyniki pomiarów wydajności znajdują się w pliku czasy.pdf.