Sprawozdanie 2

Rafał Dżumaga
Karol Mruk

Szczegółowy opis algorytmu Fleury'ego

Pseudokod

```
U = dowolny wierzchołek z G;
E = u;
WHILE istnieje krawędź wychodząca z wierzchołka u
        idź do wierzchołka v preferując krawędzie które nie są mostami;
        usuń z G krawędź {u, v};
        u = v;
        E += u;
END WHILE;
```

Wykazanie poprawności

Algorytm Fleury'ego jest bardzo prostym algorytmem. Zgodnie z jego przebiegiem z grafu usuwane są kolejne przechodzone krawędzie grafu w którym wyszukujemy cyklu Eulera. Tak więc jeśli rozpoczynamy i kończymy algorytm w tym samym wierzchołku oraz w zadanym grafie liczba składowych spójnych jest równa liczbie wierzchołków to algorytm musiał poprawnie wyznaczyć cykl Eulera. Jest tak ponieważ wymaganie przechodzenia przez krawędzie które nie są mostami w pierwszej kolejności gwarantuje nam że graf nie rozpadnie się na kilka składowych spójnych.

Analiza złożoności algorytmu

Złożoność samego algorytmu zależy liniowo od liczby krawędzi i wynosi O(|E|). Algorytm bazuje jednak na informacji czy rozpatrywana krawędź jest mostem czy nie. W zależności od wybranego algorytmu końcowa złożoność może być różna.

Dla przykładu algorytm Tarjana pozwala na wyznaczenie mostów w grafie przy złożoności O(|E|). Jednak procedurę tą należy uruchomić po każdym usunięciu krawędzi w grafie czyli tyle razy ile wynosi liczba krawędzi. Końcowe oszacowanej wynosiłoby w tym przypadku O(|E|^2).

Przy użyciu innych algorytmów wyznaczania mostów można jednak złożoność zmniejszyć.

Możliwe zastosowania

Jako przykłady zastosowań można wymienić na przykład w bioinformatyce wykorzystywanie ścieżek Eulera do rekonstruowania sekwencji DNA z jej fragmentów. Kolejnym przykładem może być wyszukiwanie optymalnych ustawień bramek logicznych w układach CMOS.

Wykorzystanie jednak algorytmu Fleury'ego jest mocno ograniczone przez jego bardzo słabą wydajność wynikającą z konieczności ciągłego wyznaczania mostów w grafie. W praktyce więc nie korzysta się z tego algorytmu a w zamian wykorzystuje dużo wydajniejsze algorytmy.

Dowód twierdzenia

Twierdzenie

Spójny graf skierowany $G = (V, E)$ ma cykl Eulera wtedy i tylko wtedy, gdy stopień wejściowy każdego wierzchołka $v \in V$ jest równy jego stopniowi wyjściowemu

Dowód

W grafie skierowanym aby przejść ścieżkę Eulera musimy przejść każdą krawędź zgodnie z jej kierunkiem. Tak więc gdy przechodzimy jakąż krawędź i usuwamy ją z grafu zgodnie z algorytmem Fleuriego obniżamy o jeden stopień wychodzący wierzchołka z którego wychodzi krawędź i o jeden stopień wchodzący wierzchołka do którego biegnie krawędź. Zgodnie z tym rozumowaniem aby powstał cykl Eulera musimy z każdego wierzchołka wyjść dokładnie taką samą liczbę razy jak do danego wierzchołka wchodzimy. Z tego wynika że stopień wejściowy i stopień wyjściowy każdego wierzchołka muszą być sobie równe.

W przeciwnym wypadku musiałby ponieważ w całym gracie suma stopni wychodzących i wchodzących wszystkich wierzchołków musi być sobie równa jeśli istniałaby para z nierówną liczbą krawędzi wchodzących i wychodzących jeden z nich musiałby mieć więcej krawędzi wchodzących niż wychodzących. W takim przypadku musi dojść do momentu w którym wejdziemy do wierzchołka ale nie będziemy w stanie z niego wyjść co uniemożliwi przebycie całego cyklu Eulera.

Implementacja

Podstawowe założenia implementacyjne

1. Program jest aplikacją konsolową napisaną w języku C#.
2. Graf skierowany wczytywany jest z pliku .txt w postaci listy sąsiedztwa, jedna linijka w pliku odpowiada liście sąsiedztwa jednego wierzchołka, gdzie pierwszym elementem w danej linijce jest dany wierzchołek, a dalej, rozdzielone spacją lub przecinkami wierzchołki sąsiadujące z nim. Wszystkie wierzchołki są przekazane jako numery.
3. Program zaczyna działanie od pierwszego wprowadzonego wierzchołka.
4. Dla danego wierzchołka zostaje wybrana krawędź łącząca go z pierwszym z wierzchołków na jego liście sąsiedztwa, która nie została jeszcze usunięta.
5. Następuje sprawdzenie, czy dana krawędź jest mostem. Jeśli nie, to ta krawędź zostaje dopisana do rozwiązania i usunięta, czyli wpis w liście sąsiedztwa dla danego wierzchołka zostaje ustawiony na wartość '-1', a następnie program ustawia jako kolejny analizowany wierzchołek jako drugi z wierzchołków tworzących daną krawędź i wykonywany jest operacja jak w kroku nr 4. Jeśli natomiast analizowana krawędź byłaby mostem, to zostaje wybrany kolejny wierzchołek z listy sąsiedztwa analizowanego wierzchołka tworzący inną krawędź i ponownie wykonywany jest krok nr 5.
6. Wyznaczanie mostów odbywa się metodą Depth-First-Search czyli przeszukiwaniem w głąb. Dla danej analizowanej krawędzi, zostają policzone wszystkie możliwe do odwiedzenia wierzchołki przed i po usunięciu danej krawędzi. Jeśli liczba możliwych do odwiedzenia wierzchołków po usunięciu krawędzi jest mniejsza niż przed usunięciem to dana krawędź jest mostem.

7. W grafie skierowanym, przy wyznaczaniu cyklu Eulera zawsze dochodzi do sytuacji, że każda z pozostałych krawędzi jest mostem. Więc w takiej sytuacji, gdy po analizie wszystkich krawędzi przyległych do danego wierzchołka nie można było wybrać żadnej, która nie jest mostem, należy wybrać taką, która pozwala na odwiedzenie jak największej liczby wierzchołków po jej usunięciu (aby móc wyznaczyć cykl Eulera). Dlatego przy wyznaczaniu mostów podczas analizy krawędzi z listy sąsiedztwa danego wierzchołka, tworzona jest hierarchia mostów na podstawie możliwej do odwiedzenia liczby wierzchołków co pozwala wybrać “najlepszy” most w sytuacji gdy zostały już tylko mosty dla danego wierzchołka.
8. Końcowym wynikiem działania programu jest lista kolejno odwiedzanych krawędzi w postaci ‘Numer wierzchołka - Numer wierzchołka’, np “3-5”.

Projekt testów

1. Program zostanie przetestowany dla kilku grafów skierowanych o różnej liczbie wierzchołków i krawędzi.
2. Oczywiście sprawdzana będzie poprawność wyznaczania cyklu Eulera w grafie.
3. Zostanie wyznaczony czas działania programu dla każdego grafu, przy czym będą także wyznaczane czasy pracy w konkretnych częściach kodu (np. przy wyznaczaniu mostów).