Jakub Pędziński, 155662, bioinformatyka semestr 3, grupa 1

Zastosowany format grafu: lista następników

W moim formacie każda linia zawiera wszystkie następniki konkretnego wierzchołka oddzielone spacją. Poza tym na początku każdej linii jest numer wierzchołka którego następniki są w niej zapisane, w postaci ‘1 : ‘. Wierzchołki muszą być zapisane po kolei.

Opis algorytmu

Grafy w algorytmie są zapisywane w formie dwóch, globalnych tablic dwuwymiarowych, jeden zapisuje graf wejściowy, a drugi wyjściowy oraz tymczasowej, lokalnej listy łuków, która jest następnie przekształcana w graf wyjściowy. Algorytm wywołuje w mainie po kolei 6 funkcji: wczytująca graf z pliku, sprawdzająca czy graf jest 1-grafem, sprawdzająca czy jest sprzężony, przekształcająca graf w jego graf oryginalny, sprawdzająca czy jest liniowy, wypisująca graf oryginalny do pliku. Opis funkcji:

1. Wczytywanie grafu z pliku – funkcja sczytuje linie z pliku i wczytują je do strumienia, następnie ze strumienia są odczytywanie następniki wierzchołka, którego dotyczy linia i zapisywane do macierzy. Jest to funkcja typu void.
2. Sprawdzanie czy graf jest 1-grafem – ta funkcja wywołuje zagnieżdżone pętle, które przechodzą po wszystkich następnikach konkretnego wierzchołka i jeśli znajdzie się taki sam następnik w jednym wierzchołku to funkcja zwraca informację, że graf nie jest 1-grafem. Natomiast warunek w mainie kończy działanie programu z odpowiednią informacją, gdy graf nie jest 1-grafem. Funkcja jest typu bool.
3. Sprawdzanie czy graf jest sprzężony – w tej funkcji porównywane są listy następników każdego wierzchołka z każdym innym. Jeśli w liście następników drugiego wierzchołka występuje taki sam następnik to jest to zapisywane w tablicy pomocniczej w postaci 1 pod odpowiednim indeksem. Następnie dane z tablicy są analizowane i jeśli występują inne dane niż same 1 lub same 0 to funkcja zwraca informację, że graf nie jest sprzężony. Tak samo jak w poprzedniej funkcji warunek w mainie kończy działanie programu z odpowiednią informacją, gdy graf nie jest sprzężony. Funkcja jest typu bool.
4. Przekształcanie grafu w jego graf oryginalny – przekształcanie przeprowadziłem w sposób polecany w treści zadania. Najpierw funkcja tworzy listę łuków uzupełniając ją kolejnymi liczbami rozpoczynając od 1, łuki są tworzone w pętli kończącej się, gdy zostanie utworzone tyle łuków, ile wierzchołków w grafie. Następnie, jeśli w grafie istnieje przejście z jednego wierzchołka do drugiego, łączy łuki oznaczone indeksem tych wierzchołków przeindeksowując wszystkie wystąpienia poprzednika łuku w grafie. Na koniec funkcja zmienia listę łuków na graf wyjściowy. Nie usuwałem wierzchołków izolowanych. Odbywa się to w pętli wykonywanej tyle razy, ile jest łuków w liście łuków, dla wierzchołka wskazanego na liście jest zapisywana następnik również wskazany na liście. Jest to funkcja typu void.

5. Sprawdzanie czy graf jest liniowy – ta funkcja działa analogicznie do funkcji sprawdzającej czy graf jest 1-grafem, ale obsługuje na grafie oryginalnym. Jest to funkcja typu bool. Zwraca ona wartość do maina na podstawie, której na konsoli jest wyświetlana odpowiednia informacja.
6. Wypisywanie grafu oryginalnego do pliku – funkcja działa w dwóch pętlach, przechodzących po wierzchołkach i ich następnikach oraz wypisuje je do pliku wyjściowego, razem z numerem wierzchołka.

Złożoność

Złożoność funkcji:

1. Wczytywanie grafu – $O(n^2)$
2. Sprawdzanie czy graf jest 1-grafem – $O(n^3)$
3. Sprawdzanie czy graf jest sprzężony – $O(n^5)$
4. Przekształcanie grafu w jego graf oryginalny – $O(n^3)$
5. Sprawdzanie czy graf jest liniowy – $O(n^3)$
6. Wypisywanie grafu oryginalnego – $O(n^2)$

Złożoność całego algorytmu – $O(n^5)$

Testowanie

- Pierwsze 4 testowanych grafów to grafy sprzężone, liniowe, każdy z nich ma 15 wierzchołków, w tych grafach występują wierzchołki bez następników, pętle własne i cykle
- Następne 4 grafy są podobne, ale nie są liniowe i zawierają każdą ze struktur wymienionych w opisie zadania
- Kolejne 4 grafy również są sprzężone, ale zawierają 10 wierzchołków, do tych grafów dołączam wizualizacje rozwiązań, pierwsze 2 są liniowe, a kolejne 2 nie
- Następne 5 grafów nie są grafami sprzężonymi
- Ostanie 3 grafy nie są 1-grafami (więc również nie są sprzężone)

Wnioski

Z przeprowadzonych analiz testów wnioskuję, że algorytm działa prawidłowo. Największy problem sprawiła mi funkcja sprawdzająca czy graf jest sprzężony. Miałem problem ze zrozumienie zależności określających czy graf jest sprzężony, po zrozumieniu ich zastosowanie wiedzy w kodzie również nie było łatwe. Przy realizacji projektu poznałem strumienie, które okazały się bardzo przydatne przy wczytywaniu grafu z pliku.

Źródła: Marta Kasprzak „Wybrane algorytmy i modele grafowe w bioinformatyce”