Wojciech Sitek

Projekt Analiza Algorytmów

Dokumentacja

Tytuł i treść problemu

Przepustowość

> *Dany jest graf opisujący pewną sieć wodociągową, w którym wierzchołki są oznaczone jako zawory, źródła lub punkty odbioru. Każdy zawór może być ustawiony na przepustowość w zakresie [0,1], każde źródło ma określoną wydajność, a odcinek stałą przepustowość w zakresie [0,1].*
>
> *Opracować algorytm, który dla danych nastaw zaworów określi ilość wody docierającej do wszystkich punktów odbioru.*

Interpretacja problemu

W opisie problemu występują trzy rodzaje obiektów:

- Źródło – obiekt, który dostarcza wodę, o parametrze *wydajność*
- Zawór – obiekt, który znajduje się na końcach odcinków, o parametrze *przepustowość* w zakresie [0,1]
- Odcinek – obiekt o parametrze *przepustowość* w zakresie [0,1].
- Punkt odbioru – obiekt o parametrze *przepływ*, który jest do znalezienia przez program.

Wszystkie te obiekty są częścią grafu przepływowego skierowanego od wierzchołka s do wierzchołka t. Od wierzchołka s krawędzie skierowane są do źródeł i mają przepustowości równe wydajności źródeł, do których prowadzą. Następnie odcinki odwzorowywane są w krawędzie o przepustowości równej przepustowości odcinków. Zawory odwzorowywane są w wierzchołki, które także posiadają przepustowość równą przepustowości zaworów. Punkty odbioru są odwzorowywane w wierzchołki o nieskończonej przepustowości. Następnie wszystkie punkty odbioru są łączone krawędziami z wierzchołkiem t. Przepływy krawędzi to szukane wartości programu.

Algorytmy

Do rozwiązania problemu zostanie użyty algorytm największego przepływu metodą Forda-Fulkersona, a dokładnie algorytm Edmondsa-Karpa. Znajduje on ścieżkę powiększającą w grafie, a następnie aktualizuje przepływ w grafie wzdłuż tej ścieżki. Algorytm kończy działanie, gdy nie istnieje już żadna ścieżka powiększająca w grafie.

Zostanie on zmodyfikowany faktem, iż w niniejszym projekcie wierzchołki także posiadają przepustowość, co zostanie uwzględnione w implementacji algorytmu.

Ścieżka powiększająca będzie poszukiwana algorytmem BFS (przeszukiwania wszerz).

```
załaduj_dane_wejściowe_do _struktur()
while(istnieje_ścieżka_powiększająca) {
        znajdź_ścieżkę_powiększającą()
        aktualizuj_przepływ_i_przepustowość_wzdłuż_ścieżki()
}
```

wypisz_przepływ_punktów_odbioru()

Wartość zwracana

W przypadku gdyby nie było możliwe przepłynięcie wody do żadnego punktu odbioru, program wypisze, że dostarczono 0j wody do każdego punktu odbioru.

W przypadku gdy istnieje rozwiązanie, algorytm zwraca listę punktów odbioru wraz z ilością wody, która do niego dopłynęła.

Złożoność algorytmu

- Załadowanie danych wejściowych do struktur będzie miało złożoność P(E) = O(|E|)
- Ścieżka powiększająca będzie znajdowana za pomocą algorytmu BFS o złożoności O(|E| + |V|).
- Aktualizacja przepływu i przepustowości w najgorszym przypadku będzie miała złożoność O(|E|), gdy ścieżka będzie zajmowała cały graf.
- Maksymalna ilość powtórzeń pętli algorytmu ma złożoność O(|V| * |E|)[1].

W związku z tym cały algorytm będzie miał złożoność O((E+E+V)*V*E) = **O($E^2V + EV^2$)**. W związku z tym zależnie od tego czy graf będzie rzadki czy gęsty, dominował będzie albo składnik wierzchołków, albo krawędzi.

Interfejs użytkownika

Część algorytmiczna projektu zostanie napisana w języku C++. Dane wejściowe będą wczytywane ze standardowego wejścia lub z pliku. Format danych wejściowych jest przedstawiony poniżej. Przedstawione w nawiasie informacje są pomocnicze.

(liczba źródeł) 2
(pierwszy punkt odbioru) 5 (liczba punktów odbioru) 2
(w1) 3 c13 4 c14
(w2) 4 c24
(w3) c3 5 c35
(w4) c4 5 c45 6 c46

W pierwszym wierszu znajduje się liczba źródeł, są to wierzchołki o numerach rozpoczynających się od 1. W drugim wierszu znajduje się numer wierzchołka pierwszego punktu odbioru i liczba punktów odbioru.

Następnie w kolejnych liniach wejścia programu znajdują się charakterystyki kolejnych wierzchołków, rozpoczynając od wierzchołka 1. Znajduje się tam kolejno: przepustowość wierzchołka (tylko dla zaworów), lista par {numer wierzchołka z którym dany wierzchołek tworzy krawędź; przepustowość tej krawędzi}. Punkty odbioru nie są charakteryzowane przez linie wejścia programu.

Struktury danych

Po wczytaniu danych, są one umieszczane w strukturach danych. Podstawową strukturą danych jest wektor wierzchołków i ich charakterystyk. Typem tego wektora jest

[1] Dowód na stronie: https://brilliant.org/wiki/edmonds-karp-algorithm/

para {przepustowość wierzchołka, tablica mieszająca krawędzi wychodzących z tego wierzchołka}. Krawędź wychodzącą z danego wierzchołka reprezentuje struktura *Edge*, opisana poniżej.

```
vector<pair<double, unordered_map<int, Edge>>>
```

```
struct Edge{
    double capacity;
    double flow;
    struct Edge *reverse_edge; //potrzebny do aktualizacji c
    int v2_number;
    bool is_visited; //needed to BFS
};
```

Wizualizacja danych

Po wykonaniu obliczeń i otrzymaniu wyników przez algorytm, parametry czasowe i inne statystyki wykonania programu zostaną zapisane w pliku tekstowym. Następnie zostanie wykonany skrypt w języku **Python**, w którym zostanie zaimplementowana wizualizacja danych za pomocą m.in. biblioteki **numpy** w postaci wykresów i tabel ilustrujących złożoność algorytmu.

Generator danych testowych

W języku **C++** zostanie przygotowany także generator losowych danych testowych z parametrami liczby źródeł, zaworów, odcinków, punktów odbioru i opcjonalnymi parametrami średniej długości ścieżki powiększającej oraz odchylenia standardowego długości ścieżki powiększającej.

Algorytm

1. Graf przepływowy zawierał będzie na początku
   - niepołączone węzły s i t
   - połączone z s wierzchołki źródeł
   - połączone z t wierzchołki punktów odbioru
2. Do grafu zostaną dodane wierzchołki zaworów w ilości podanej w parametrach
3. Zostanie dodany licznik wierzchołków spójnych z podstawową wersją grafu tak, aby na koniec uspójnić graf
4. W pętli będą dodawane losowo ścieżki powiększające do grafu i zliczać dodane do spójnego grafu wierzchołki i krawędzie. Początkowo ścieżki powiększające będą dodawane w taki sposób, aby dodać jak najwięcej krawędzi, czyli będą tworzone z wierzchołków niepołączonych.