# МИНОБРНАУКИ РОССИИ

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра МО ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №3**

**по дисциплине «Построение и анализ алгоритмов» ТЕМА: «ПОТОКИ»**

|  |  |
| --- | --- |
| Студент гр. 7304 | Соколов И.Д. |
| Преподаватель | Филатов А.Ю. |

Санкт-Петербург 2019

# Цель работы

Найти максимальный поток в сети, а также фактическую величину потока, протекающего через каждое ребро, используя алгоритм Форда- Фалкерсона.

# Задание

Найти максимальный поток в сети, а также фактическую величину потока, протекающего через каждое ребро, используя алгоритм Форда- Фалкерсона.

Сеть (ориентированный взвешенный граф) представляется в виде триплета из имѐн вершин и целого неотрицательного числа - пропускной способности (веса).

Входные данные:

NN - количество ориентированных рѐбер графа v0 - исток

vn - сток

vi vj ωij - ребро графа vi vj ωij - ребро графа

...

Выходные данные:

Pmax - величина максимального потока

vi vj ωij - ребро графа с фактической величиной протекающего потока vi vj ωij - ребро графа с фактической величиной протекающего потока

...

В ответе выходные рѐбра отсортируйте в лексикографическом порядке по первой вершине, потом по второй

# Описание алгоритма Форда-Фалкерсона

1. Обнуляем все потоки. Остаточная сеть изначально совпадает с исходной сетью.
2. В остаточной сети находим путь из источника в сток. Если такого пути нет, останавливаемся.
3. Пускаем через найденный путь максимально возможный поток:
   1. На найденном пути в остаточной сети ищем ребро с минимальной пропускной способностью tmp.
   2. Для каждого ребра на найденном пути увеличиваем поток на tmp, а в противоположном ему - уменьшаем на tmp.
   3. Модифицируем остаточную сеть. Для всех рѐбер на найденном пути, а также для противоположных им рѐбер, вычисляем новую пропускную способность. Если она стала ненулевой, добавляем ребро к остаточной сети, а если обнулилась, стираем его.
4. Возвращаемся на шаг 2.

# Описание функций

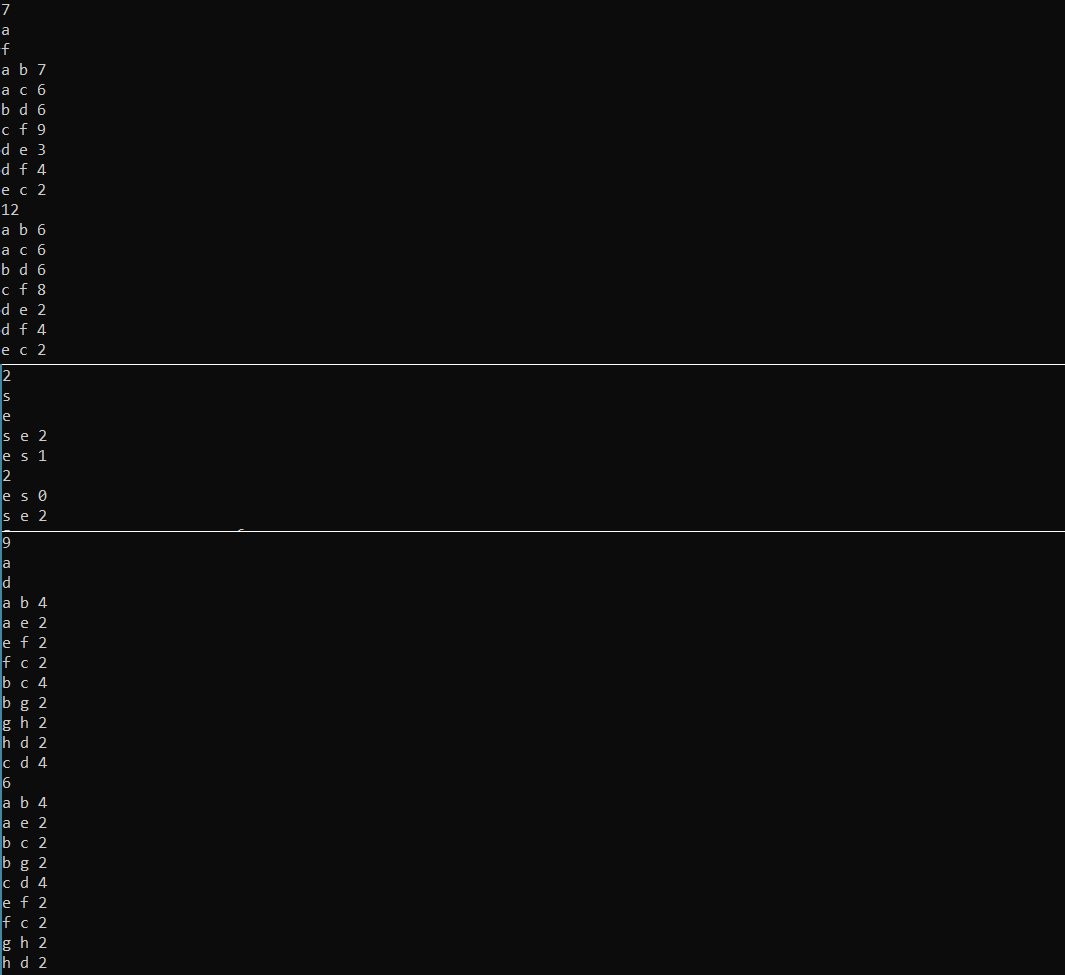
*int FF()* - вычисление максимального потока в графе, алгоритм Форда- Фалкерсона.

*void initPrevs()* – инициализация списка вершин.

*void DFS(int v)* – поиск в глубину.

*bool getPath(int v)* – поиск пути.

# Тестирование



**Вывод**

В ходе лабораторной работы мы подробно ознакомились с алгоритмом Форда-Фалкерсона. В результате выполнения лабораторной работы была

разработана программа, которая вычисляет максимальный поток в ориентированном графе. Решена проблема реализации обратного хода по ориентированному графу.