**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. 6381 |  | Вероха В.Н. |
| Преподаватель |  | Филатов А.Ю. |

Санкт-Петербург

2018

**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 имён вершин и целого неотрицательного числа - пропускной способности (веса).

Входные данные:  
NN - количество ориентированных рёбер 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 рёбра 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 рёбер на найденном пути, а также для противоположных им рёбер, 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**

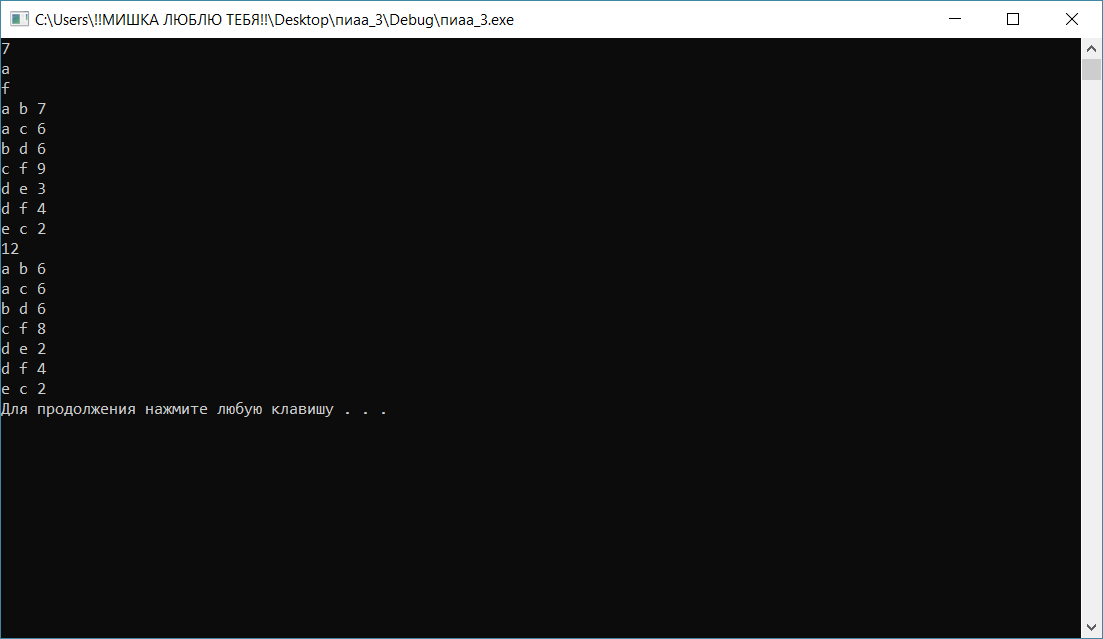
*int \_ff()* - вычисление максимального потока в графе.

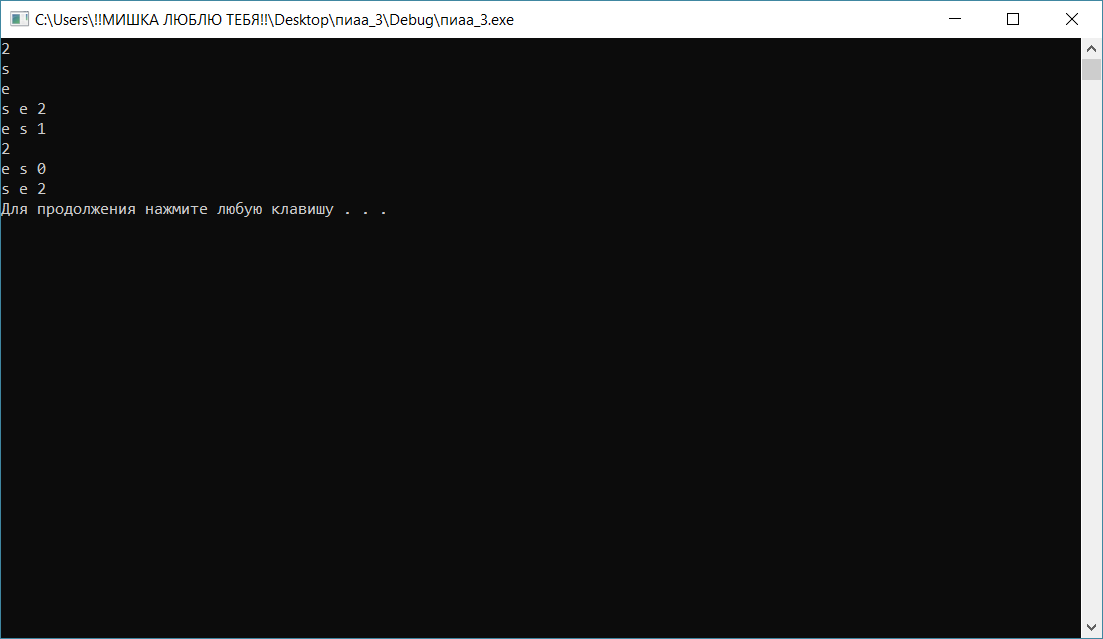
*void initPrevs()* – инициализация списка вершин.

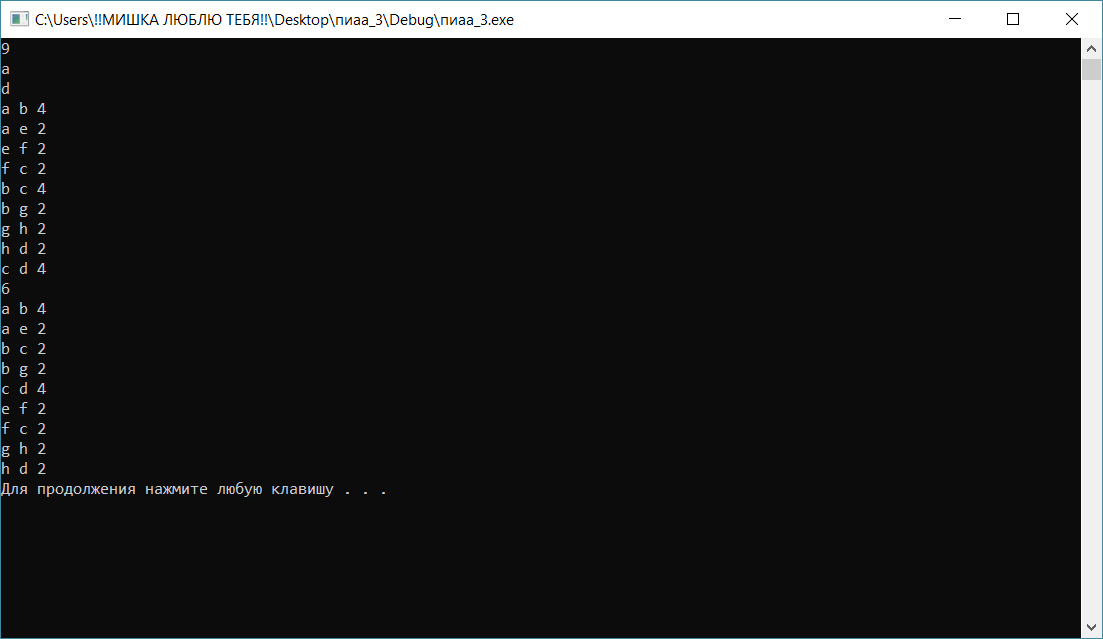
*void bf(int v)* – поиск в глубину.

*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.