

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №3**  
**по дисциплине «Построение и анализ алгоритмов»**  
**Тема: Потоки в сети**

Студентка гр. 7304

\_\_\_\_\_

Нгуен Т.Т. Зуен

Преподаватель

\_\_\_\_\_

Филатов А.Ю.

Санкт-Петербург

2019

## Цель работы:

Исследование алгоритма Форда-Фалкерсона и его реализация на языке C++.

## 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:

$N$  - количество ориентированных рёбер графа

$v_0$  - исток

$v_n$  - сток

$v_i \quad v_j \quad w_{ij}$  - ребро графа

$v_i \quad v_j \quad w_{ij}$  - ребро графа

...

Выходные данные:

$P_{max}$  - величина максимального потока

$v_i \quad v_j \quad w_{ij}$  - ребро графа с фактической величиной протекающего потока

$v_i \quad v_j \quad w_{ij}$  - ребро графа с фактической величиной протекающего потока

...

В ответе выходные рёбра отсортируйте в лексикографическом порядке по первой вершине, потом по второй (в ответе должны присутствовать все указанные входные рёбра, даже если поток в них равен 0).

## Описание алгоритма

Алгоритм Форда — Фалкерсона решает задачу нахождения максимального потока в транспортной сети. Идея алгоритма заключается в следующем. Изначально величине потока присваивается значение 0:  $f(u,v)=0$  для всех  $u, v \in V$ . Затем величина потока итеративно увеличивается посредством поиска увеличивающего пути (путь от источника  $s$  к стоку  $t$ , вдоль которого можно послать больший поток). Процесс повторяется, пока можно найти увеличивающий путь.

### Неформальное описание

1. Обнуляем все потоки. Остаточная сеть изначально совпадает с исходной сетью.
2. В остаточной сети находим любой путь из источника в сток. Если такого пути нет, останавливаемся.
3. Пускаем через найденный путь (он называется *увеличивающим путём* или *увеличивающей цепью*) максимально возможный поток:
  1. На найденном пути в остаточной сети ищем ребро с минимальной пропускной способностью `path_flow`.
  2. Для каждого ребра на найденном пути увеличиваем поток на `path_flow`, а в противоположном ему — уменьшаем на `path_flow`.
  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:

`void DFS(int cur)` — поиска в глубину для поиска пути с истока до стока.

`bool buildPath(char start, char end)` — поиска пути.

`int FordFulkerson(char start, char end)` — алгоритм Форда — Фалкерсона найти нахождения максимального потока в сети.

## Результаты:

```
C:\Users\duyenNH\source\repos\PiAA_sem2\Debug\lab3.exe
7
a
f
a b 7
a c 6
b d 6
c f 9
d e 3
d f 4
e c 2
12
a b 6
a c 6
b d 6
c f 8
d e 2
d f 4
e c 2
Press any key to continue . . .
```

## Выводы:

В результате работы программы был реализован алгоритм Форда — Фалкерсона для поиска нахождения максимального потока в транспортной сети. Также реализована DFS поиска в глубину для поиска пути с истока до стока и получены результаты.