

**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 в сети**

Студент гр. 7383

\_\_\_\_\_

Васильев А.И.

Преподаватель

\_\_\_\_\_

Жангиров Т.Р.

Санкт-Петербург

2019

## **Цель работы.**

Исследовать и реализовывать задачу нахождения максимального потока в сети, применяя алгоритм Форда – Фалкерсона.

Формулировка задачи: 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 – пропускная способность (веса).

Уточнение задачи: граф представлен в виде списка смежности, поиск пути задаётся через поиск в ширину.

Входные данные: в первой строке указывается количество ориентированных рёбер графа, затем идут значения начальной и конечной вершин. Далее вводят данные о рёбрах графа и их весе, пропускной способности.

Выходные данные: максимальный поток в сети, а также фактическая величина потока, протекающего через каждую дугу, все рёбра отсортированы в лексикографическом порядке.

## **Реализация задачи.**

В работе был использован класс `Graph`.

Параметры, хранящиеся в классе:

`way` – одномерный массив, в котором хранятся значения родителей вершин;

`Given_graph` – упорядоченный ассоциативный массив типа `map`, благодаря которому заданный граф представлен как список смежности;

`RealFlow` – упорядоченный ассоциативный массив типа `map`, в котором хранятся значения фактического потока, проходящего через все рёбра графа;

`check` – массив типа `bool`, в котором хранятся метки, показывающие пройдена данная вершина или нет.

Методы класса:

**ResidualCapacity** – функция нахождения разницы между величинами потока в заданном и фактическом графах, остаточная пропускная способность;

**Bfs** – функция, реализующая поиск в ширину, возвращает **true**, если путь до стока существует, в противном случае – **false**;

**AddEdge** – функция заполнения списка смежности, заданными значениями;

**Ford\_Fulkerson** – алгоритм Форда – Фалкерсона, возвращает максимальное значение потока, а также по ходу работы данной функции заполняются фактические значения потока;

**Print** – функция вывода результата работы программы на экран.

Так же в классе определены деструктор и конструктор по умолчанию.

В функции **main()** считывается количество дуг графа, затем задаются исток и сток, между которыми проходит поток. Далее в цикле начинается считывание рёбер графа и их пропускная способность. Запускается алгоритм Форда – Фалкерсона, который работает до тех пор, пока можно найти путь из истока в сток, используя функцию поиска в ширину **Bfs**. На каждом шаге итерации увеличиваются потоки в рёбрах данного маршрута на минимальную остаточную пропускную способность.

### **Тестирование.**

Тестовые случаи не выявили не правильного поведения программы, следовательно можно судить о том, что поставленная задача была выполнена.

Тестовые случаи приведены в табл. 1.

Таблица 1 – Тестовые случаи

Входные данные	Вывод	Верно?
7 a f	12	Да
a b 7	a b 6	
a c 6	a c 6	

b d 6 c f 9 d e 3 d f 4 e c 2	b d 6 c f 8 d e 2 d f 4 e c 2	
13 a h a b 6 a c 6 b d 4 b e 2 c b 2 c e 9 d f 4 d g 2 e d 8 e g 7 f h 7 g f 11 g h 4	11 a b 6 a c 5 b d 4 b e 2 c b 0 c e 5 d f 4 d g 0 e d 0 e g 7 f h 7 g f 3 g h 4	Да

### Исследование.

Поскольку на каждой итерации поток как минимум увеличивается на один, а поиск пути в графе происходит за  $O(|E|)$  операций, то сложность алгоритма составляет  $O(F|E|)$ , где  $F$  – максимальный поток в сети. Данная оценка требует знать величину максимального потока, но так как он не может превышать сумму пропускных способностей истока и сумму пропускных способностей стока, то можно заменить  $F$  на максимальную из этих двух сумм. Тогда  $O(M|E|)$ .

### **Выводы.**

В ходе лабораторной работы был изучен алгоритм поиска максимального потока в сети, используя алгоритм Форда – Фалкерсона. Был написан код на языке программирования C++, который применял этот метод для поставленной задачи. Сложность реализованного алгоритма составляет  $O(|M|E)$ .