## 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: АЛГОРИТМ ФОРДА-ФАЛКЕРСОНА.**

|  |  |  |
| --- | --- | --- |
| Студент гр. 7304 |  | Сергеев И.Д. |
| Преподаватель |  | Филатов Ар.Ю. |

Санкт-Петербург 2019

# Цель работы:

Исследовать алгоритм Форда-Фалкерсона, найти максимальный поток в сети, а также фактическую величину потока, протекающего через каждое ребро

# Задача:

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

𝑣0v0 - исток

𝑣𝑛vn - сток

𝑣𝑖𝑣𝑗𝜔𝑖𝑗vivjωij - ребро графа

𝑣𝑖𝑣𝑗𝜔𝑖𝑗vivjωij - ребро графа

...

Выходные данные:

𝑃𝑚𝑎𝑥Pmax - величина максимального потока

𝑣𝑖𝑣𝑗𝜔𝑖𝑗vivjωij - ребро графа с фактической величиной протекающего потока

𝑣𝑖𝑣𝑗𝜔𝑖𝑗vivjωij - 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).

## Sample Input:

7

a f

a b 7

a c 6

b d 6

c f 9

d e 3

d f 4

e c 2

## Sample Output:

12

a b 6

a c 6

b d 6

c f 8

d e 2

d f 4

e c 2

# Основные теоретические положения:

**Алгоритм Форда — Фалкерсона** решает [задачу нахождения максимального](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BE_%D0%BC%D0%B0%D0%BA%D1%81%D0%B8%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%BC_%D0%BF%D0%BE%D1%82%D0%BE%D0%BA%D0%B5) [потока](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BE_%D0%BC%D0%B0%D0%BA%D1%81%D0%B8%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%BC_%D0%BF%D0%BE%D1%82%D0%BE%D0%BA%D0%B5) в [транспортной сети](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D0%BE%D1%80%D1%82%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C).

Идея алгоритма заключается в следующем. Изначально величине потока

присваивается значение 0: для всех . Затем величина

потока [итеративно](https://ru.wikipedia.org/wiki/%D0%98%D1%82%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D1%8F)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.

# Ход работы:

## Реализована функция поиска в ширину.

Данная функция необходима для нахождения пути из истока в сток. Наличие пути – главный критерий алгоритма Форда-Фалкерсона. На вход функция принимает граф, исток, сток и ассоциативный массив путей. Поиск продолжается пока очередь всех вершин не будет пуста.



bool bfs(map<char,map<char,int>>rgraph, char start, char end, map<char,char>& path){ map<char,bool>visited;

queue<char>q; q.push(start); visited[start] = true; while (!q.empty())

{

char u = q.front(); q.pop();

for (auto const &sub : rgraph[u]){

if (sub.second > 0 && visited[sub.first] == false) { q.push(sub.first);

visited[sub.first] = true; path[sub.first] = u;



}

}

}

return (visited[end] == true);

}

## Алгоритм Форда-Фалкерсона.

Функция реализует алгоритм нахождения максимального потока. Также результатом ее работы является граф с фактическими величинами потоков через дуги. На вход функция принимает граф, исток и сток.



int FordFulkerson(map<char,map<char,int>>&graph, map<char,map<char,int>>&rgraph, char start, char finish){



char u,v;



map<char,char>path;



for (auto const &main: graph)



for (auto const &sub: graph[main.first])



rgraph[main.first][sub.first] = sub.second;



int max\_flow = 0;

while (bfs(rgraph,start,finish,path))



{



int path\_flow = numeric\_limits<int>::max();



for (v = finish; v != start; v = path[v])



{



u = path[v];



path\_flow = min(path\_flow, rgraph[u][v]);



}



for (v = finish; v != start; v = path[v])



{



u = path[v];



rgraph[u][v] -= path\_flow;



rgraph[v][u] += path\_flow;



}



max\_flow += path\_flow;



}



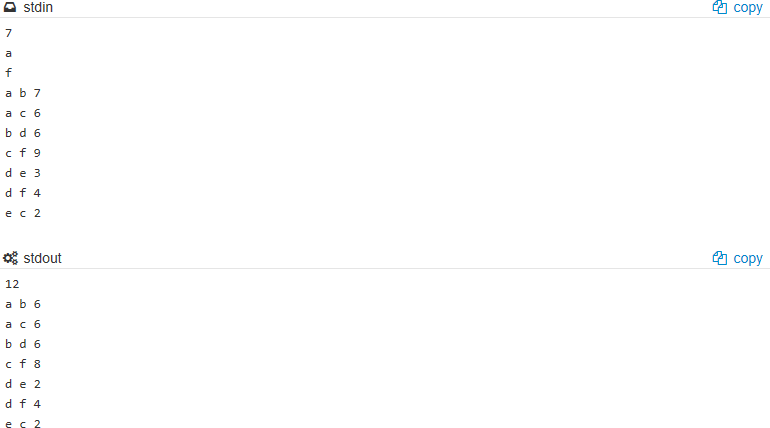
return max\_flow;



}

# Результат:

Из рисунка 1 видно, что разработанная программа выполняют поставленные задачи, а именно: программа находит максимальный поток через граф и граф с фактическим потоком ч/з каждую из дуг.



# Вывод:

Рисунок 1.

Таким образом, в ходе данной лабораторной работы было подробно изучено написание алгоритма Форда-Фалкерсона. Разработана функция поиска в глубину, которая на каждом из этапов ищет доступный путь и запоминает его. Пока существует путь хотя бы 1 путь алгоритм насыщает его минимально возможным потоком по нему, прибавляет поток по прямым дугам и вычитает из потока по обратным дугам. Результатом работы программы является граф с максимальными потоками по дугам и суммарный максимально возможный поток. Полученный результат удовлетворяет задание лабораторной работы.