**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. 6381 |  | Герасимова Д.В. |
| 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.

**Описание алгоритма**

В процессе работы алгоритма происходит обработка всех возможных результатов поиска в глубину пути из истока в сток с пересчетом максимального возможного потока через этот путь на каждой итерации. При пересчете потока через данный путь происходит поиск допустимого увеличения потока на каждом из рёбер и выбор наименьшего из них с последующим увеличением или уменьшением потока на эту величину на каждом из рёбер в зависимости от его направления.

**Описание методов и полей классов**

class Node – вершина графа

Поля класса:

* char name – имя вершины
* bool visited – была ли посещена вершина
* vector<Edge\*> neighbours – список смежных ребер

Методы класса:

* Node(char name) : name(name), visited(false) – конструктор
* char getName() const – имя вершины
* bool isVisited() const – была ли вершина посещена
* void SetVisit(bool visited) – отметить вершину как посещенную или непосещенную

class Graph - граф

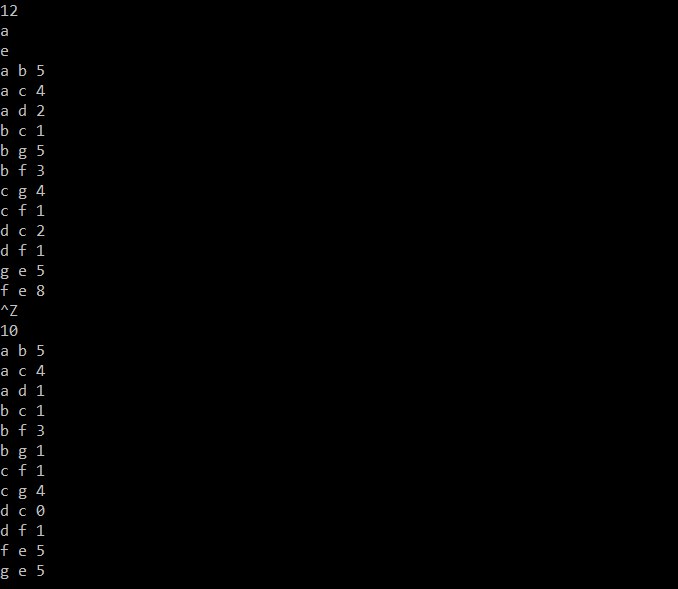
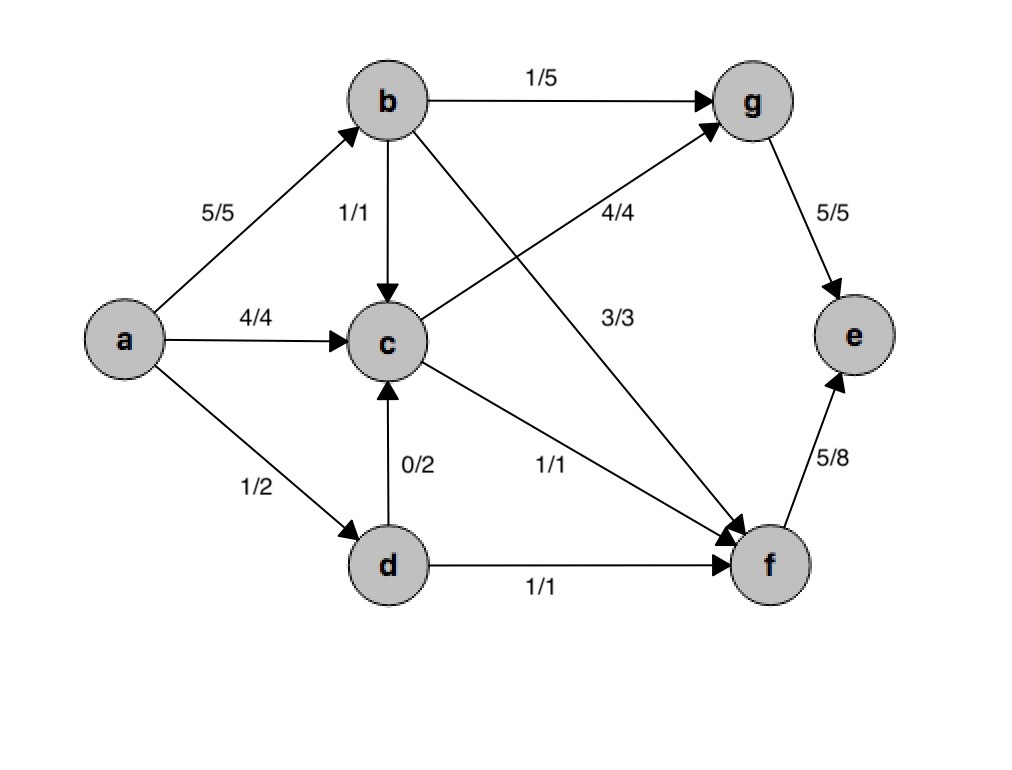
Поля класса:

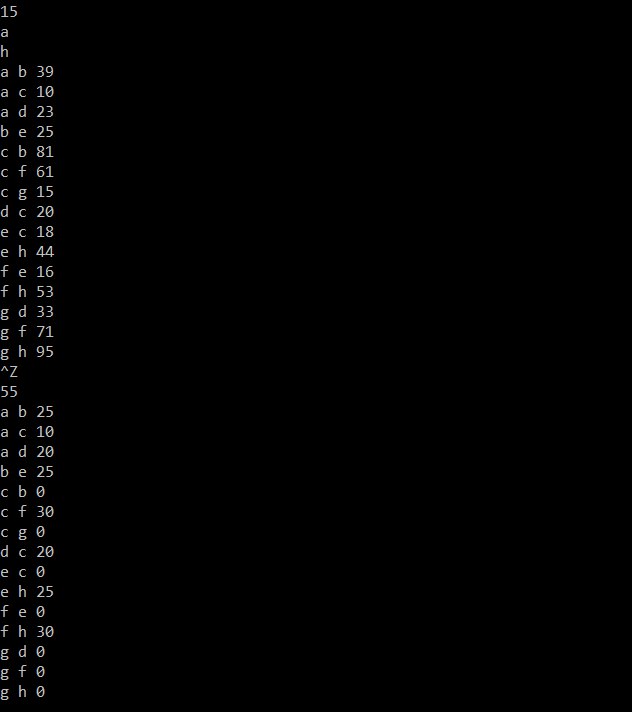
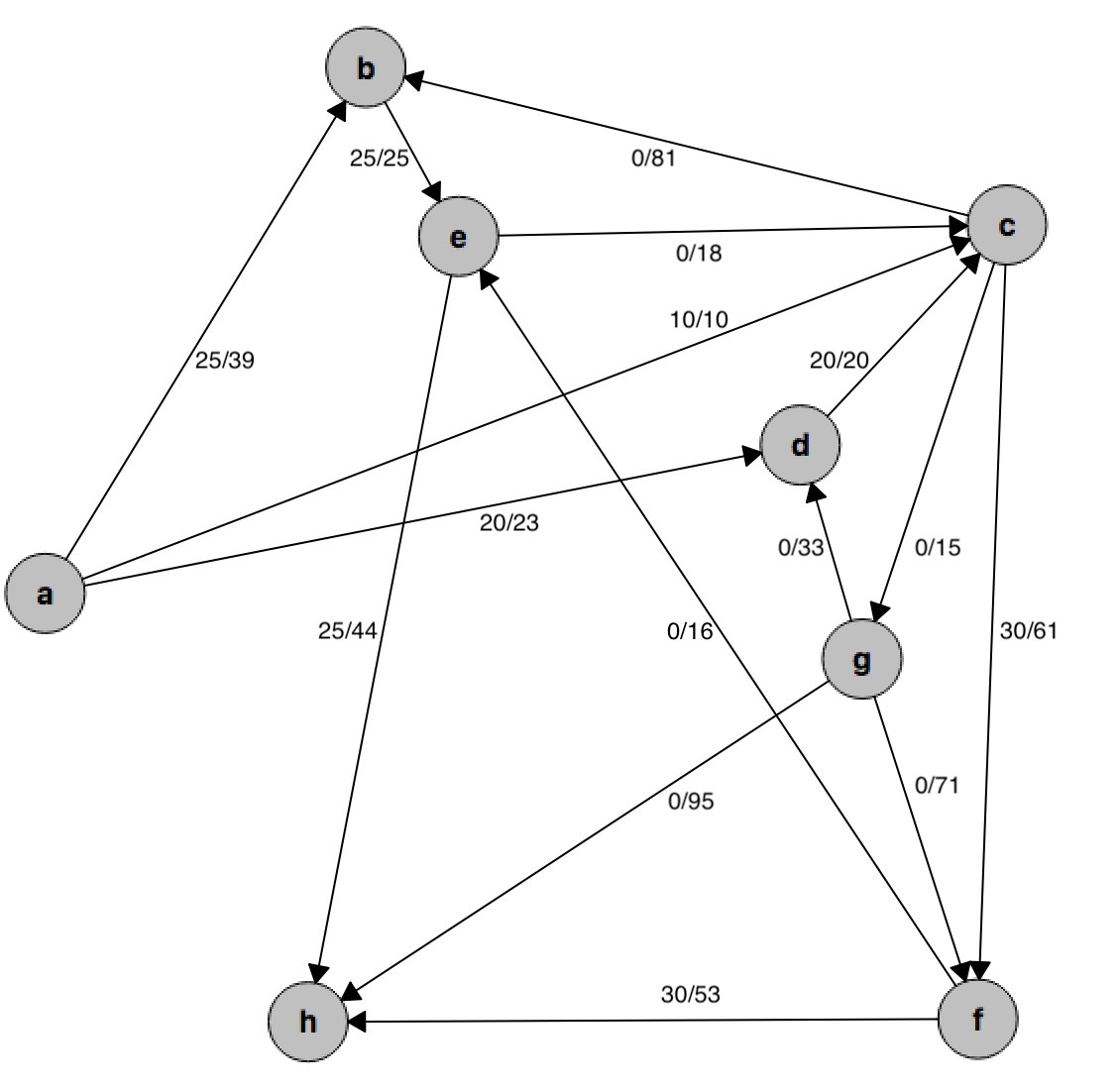
* vector<Node\*> nodes – список вершин графа
* map<Node\*, Ede\*> came\_from – ассоциативный массив переходов из вершину в вершину
* stack<pair<Edge\*, bool>> next\_nodes – стек вершин, которые необходимо посетить

Методы класса:

* Node\* getNode(char name) – возврат указателя на объект вершины по ее имени name
* void visitNode(Node\* node, Node\* point\_to) – посещение вершины node
* void findMaxFlow(Node\* point\_from, Node\* point\_to) – поиск максимального поток в графе из point\_from в point\_to
* int findMinimum(Node\* point\_from, Node\* point\_to) – поиск минимальной величины увеличения потока в пути из point\_from в point\_to
* void setMinimum(Node\* point\_to, int minimum) – установить значения потока в пути
* void printAnswer(Node\* point\_from) – вывод максимального потока в графе и потока через каждое ребро

**Тестирование**

****

****

**Вывод**

В процессе выполнения лабораторной работы был реализован алгоритм Форда-Фалкерсона для поиска максимального потока в сети, а также фактической величины потока, протекающего через каждое ребро.