**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 \_\_\_\_\_\_\_\_\_\_\_\_\_ Федянин Н.И.

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_ Филатов А.Ю.

Санкт-Петербург

2018

**Цель работы.**

Изучение и реализация алгоритма Форда-Фалкерсона поиска максимального потока на графе.

**Условие задания.**

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

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 функций и структур данных.**

Структуры данных и методы:

struct Info { - информация о имени вершины в которую есть путь из данной вершины, её вес и вес, для вычисления оптимального потока;

Info(char name, float flow, float first\_flow): name(name), flow(flow), first\_flow(flow){};

Info(){};

bool operator <(const Info b) const {

return name < b.name;

}

char name;

float flow;

float first\_flow;};

class Vertex { - вершина с прилегающим к ней списком смежных вершин и их путь, а также посещаемость и информация метки;

public:

Vertex(char name) : name(name), back\_flow(-1), back\_name('&'){};

Vertex(){};

char get\_name() const {

return name;

}

bool isVisited() const{

return visit;

}

list<Info> list\_edge - список всевозможных путей, в другие вершины + вес;

bool visit = 0 – посещаемость;

float back\_flow - обратный поток

char back\_name - вершина из которой пришли

private:

char name;

};

class Adjacency\_list { - список смежности;

public:

Adjacency\_list() { }

Vertex& at(char a) – взятие по индексу;

void print\_list() – вывод выходных данных;

void optimal\_flow() - вывод выходных данных для степика;

void cleaning\_labels() – чистка меток и посещаемости;

void algorithm\_Ford\_Falkerson (const Vertex& i\_source, const Vertex& i\_effluent) – алгоритм Форда – Фалкерсона;

bool path\_effluent(const Vertex& i\_source, const Vertex& i\_effluent) - нахождение пути от истока к стоку;

void residual\_capacity(const Vertex& i\_source, const Vertex& i\_effluent) – вычисление остаточной пропускной способности;

float max\_flow = 0 – максимальный поток сети;

vector<Vertex> list\_vertex – вектор вершин;

Info curr – временная переменная для хранения min потока;

vector<Info> multiplicity – путь от истока к стоку;

**Тестирование.**

|  |  |
| --- | --- |
| Граф | Результат |
| http://graphonline.ru/tmp/saved/vb/vbSPlpPDHLDllHGn.png |  |
| http://graphonline.ru/tmp/saved/ID/IDeOzrShfVlSVsmB.png |  |
| http://graphonline.ru/tmp/saved/LW/LWzZZlfdGhMuwGyK.png |  |
| http://graphonline.ru/tmp/saved/IH/IHGCUkOALQkogcFa.png |  |

Графическое изображение графов сделано с помощью http://graphonline.ru.

**Вывод.**

В процессе выполнения данной лабораторной работы был изучен и реализован алгоритм Форда-Фалкерсона поиска максимального потока в графе.