**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 |  | Токун М.С. |
| 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.

**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:*  
NN - 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 алгоритма.**

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 tmp.
   2. Для каждого ребра на найденном пути увеличиваем поток на tmp, а в противоположном ему — уменьшаем на tmp.
   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 deepSearch(int num) –** поиск путей в конечную точку.

**bool getPath(int istock, int stock) –** принимает на вход начальную и конечную вершину, определяет есть ли путь из начальной в конечную точку.

**int calcFlow(int stock, int istock) –** нахождение максимального потока.

**void printResult() –** вывод результатов на экран.

**Тестирование.**

|  |  |
| --- | --- |
| Данные на вход | Результат работы программы |
| 7  a f  a b 7  a c 6  b d 6  c f 9  d e 3  d f 4  e c 2 |  |
| 2  a b  a b 2  a c 3 |  |

**Вывод.**

В ходе лабораторной работы мы подробно ознакомились с алгоритмом Форда-Фалкерсона. В результате выполнения лабораторной работы была разработана программа, которая вычисляет максимальный поток в ориентированном графе.