

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

Моторин Е.В.

Преподаватель

Филатов А.Ю.

Санкт-Петербург

2019

Цель работы:

Ознакомиться с жадным алгоритмом и методом A^* , получить навыки их программирования и применения на языке программирования C++.

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

v_0v_0 - источник

v_nv_n - сток

$v_i v_j \omega_{ij} v_i v_j \omega_{ij}$ - ребро графа

$v_i v_j \omega_{ij} v_i v_j \omega_{ij}$ - ребро графа

...

Выходные данные:

P_{max} P_{max} - величина максимального потока

$v_i v_j \omega_{ij} v_i v_j \omega_{ij}$ - ребро графа с фактической величиной протекающего потока

$v_i v_j \omega_{ij} v_i v_j \omega_{ij}$ - 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 потока в транспортной сети.

Идея алгоритма заключается в следующем. Изначально величине потока

присваивается значение 0: для всех . Затем величина потока итеративно 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:

1. 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(Graph rGraph, WAY task, map<char, char>& path) {
    map<char, bool> visited;
    queue<char> q;
    q.push(task.from);
    visited[task.from] = 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[task.to] == true);
}

```

2. 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.

```

FF fordFulkerson(GRAPH& graph, WAY task) {
    char u, v;
    GRAPH rGraph;
    map<char, char> path;

    for (auto const &main: graph)
        for (auto const &sub: graph[main.first])
            rGraph[main.first][sub.first] = sub.second;

    int maxFlow = 0;
    while (bfs(rGraph, task, path)) {
        int pathFlow = INT_MAX;

        for (v = task.to; v != task.from; v = path[v]) {
            u = path[v];
            pathFlow = min(pathFlow, rGraph[u][v]);
        }

        for (v = task.to; v != task.from; v = path[v]) {
            u = path[v];
            rGraph[u][v] -= pathFlow;
            rGraph[v][u] += pathFlow;
        }

        maxFlow += pathFlow;
    }
    return FF(rGraph, maxFlow);
}

```

Результат:

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

```
7
a
f
a b 7
a c 6
b d 6
c f 9
d e 3
d f 4
e c 2
12
a b 6
a c 6
b d 6
c f 8
d e 2
d f 4
e c 2
Program ended with exit code: 0
```

Рисунок 1.

Вывод:

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 путем нахождения минимально возможного потока по нему и прибавления его к потоку по прямым дугам и вычитанию из потока по обратным дугам.

Результатом работы программы является граф с максимальными потоками по дугам и суммарный максимально возможный поток. Полученный результат удовлетворяет заданию лабораторной работы.