

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский Авиационный Институт»
Национальный Исследовательский Университет

Институт №8 «Информационные технологии и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

Лабораторная работа №9
по курсу «Дискретный анализ»

Студент:	Хренникова А. С.
Группа:	М8О-208Б-19
Преподаватель:	Симахин И. Н.
Подпись:	
Оценка:	
Дата:	

Москва, 2021

Лабораторная работа №9

Вариант №7:

Разработать программу на языке C или C++, реализующую указанный алгоритм согласно заданию:

Задан взвешенный ориентированный граф, состоящий из n вершин и m ребер. Вершины пронумерованы целыми числами от 1 до n . Необходимо найти величину максимального потока в графе при помощи алгоритма Форда-Фалкерсона. Для достижения приемлемой производительности в алгоритме рекомендуется использовать поиск в ширину, а не в глубину. Истоком является вершина с номером 1, стоком – вершина с номером n . Вес ребра равен его пропускной способности. Граф не содержит петель и кратных ребер.

Формат входных данных:

В первой строке заданы $1 \leq n \leq 2000$ и $1 \leq m \leq 10000$. В следующих m строках записаны ребра. Каждая строка содержит три числа – номера вершин, соединенных ребром, и вес данного ребра. Вес ребра – целое число от 0 до 10^9 .

Формат результата:

Необходимо вывести одно число – искомую величину максимального потока. Если пути из истока в сток не существует, данная величина равна нулю.

1 Описание

Граф представляется в виде матрицы смежности в двумерном векторе.

Алгоритм Форда-Фалкерсона – алгоритм, который решает задачу нахождения максимального потока в транспортной сети. Данный алгоритм состоит из поиска в ширину и изменения весов графа в соответствии с минимальным весом ребра в найденном пути.

Мы ищем любой путь из истока в сток и пускаем там максимальный поток, который можем пустить (минимальное значение пропускной способности из всех задействованных ребер). При этом мы модифицируем граф, то есть если мы пустили из V_1 в V_2 поток n и уменьшили пропускную способность V_1 - V_2 , то мы увеличиваем пропускную способность V_2 - V_1 на это же самое значение n . Как только мы не можем найти новый путь, мы построили максимальный поток, и, сложив все те значения потоков на каждой из итераций мы получаем искомое значение максимального потока.

2 Исходный код:

```
#include <iostream>
#include <climits>
#include <cstring>
#include <queue>
#include <vector>

using namespace std;

bool BFS(vector<vector<int>> &graph, int s, int t, vector<int> &parent) {

    queue<int> q;
    q.push(s);
    vector<bool> used(graph.size(), false);
    used[s] = true;
    parent[s] = -1;

    while (!q.empty()) {
        int u = q.front();
        q.pop();

        for (int i = 0; i < graph.size(); i++) {
            if (!used[i] && graph[u][i] > 0) {
                q.push(i);
                parent[i] = u;
                used[i] = true;
            }
        }
    }
    return (used[t]);
}

long long FordFulkerson(vector<vector<int>> &graph, int s, int t) {

    vector<int> parent(t + 1);

    long long maxFlow = 0;

    while (BFS(graph, s, t, parent)) {
        int pathFlow = 1000000001;
        for (int i = t; i != s; i = parent[i]) {
            int j = parent[i];
            pathFlow = min(pathFlow, graph[j][i]);
        }

        for (int i = t; i != s; i = parent[i]) {
            int j = parent[i];
            graph[j][i] -= pathFlow;
            graph[i][j] += pathFlow;
        }

        maxFlow += pathFlow;
    }
    return maxFlow;
}
```

```
int main() {  
    int n, m;  
    cin >> n >> m;  
  
    vector<vector<int>> graph(n + 1, vector<int> (n + 1));  
  
    int from, to, val;  
    for (int j = 0; j < m; ++j) {  
        cin >> from >> to >> val;  
        graph[from][to] = val;  
    }  
  
    cout << FordFulkerson(graph, 1, n) << endl;  
  
    return 0;  
}
```

3 Консоль:

```
lina_tucha@LAPTOP-44CRFC1U:~/labs/da$ ./x09
```

```
5 6
```

```
1 2 4
```

```
1 3 3
```

```
1 4 1
```

```
2 5 3
```

```
3 5 3
```

```
4 5 10
```

```
7
```

```
lina_tucha@LAPTOP-44CRFC1U:~/labs/da$ ./x09
```

```
8 15
```

```
1 2 39
```

```
5 8 44
```

```
7 4 33
```

```
6 8 53
```

```
1 3 10
```

```
5 3 18
```

```
7 8 95
```

```
1 4 23
```

```
3 6 61
```

```
3 2 81
```

```
7 6 71
```

```
2 5 25
```

```
3 7 15
```

```
4 3 20
```

```
6 5 16
```

```
55
```

4 Тест производительности:

	BFS(sec.)	DFS(sec.)
10	0.000172	0.002214
100	0.002685	0.326340
1000	0.033154	4.872891

5 Выводы:

Выполняя данную лабораторную работу, я изучила способы представления графа на C++ и некоторые алгоритмы по работе с ними: обход в ширину и алгоритм Форда-Фалкерсона для нахождения максимального потока в транспортной сети.