

Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной
математики

Кафедра вычислительной математики и программирования

Лабораторная работа №9 по курсу «Дискретный анализ»

Студент: Г. С. Шубин
Преподаватель: Н. С. Капралов
Группа: М8О-308Б
Дата:
Оценка:
Подпись:

Москва, 2022

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 ФордаФалкерсона. 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 $V1$ в $V2$ поток p и уменьшили пропускную способность $V1-V2$, то мы увеличиваем пропускную способность $V2-V1$ на это же самое значение p . 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

```
1 #include <iostream>
2 #include <climits>
3 #include <cstring>
4 #include <queue>
5 #include <vector>
6 #include <time.h>
7
8 using namespace std;
9
10 bool BFS(vector<vector<int>> &graph, int s, int t, vector<int> &parent) {
11
12     queue<int> q;
13     q.push(s);
14     vector<bool> used(graph.size(), false);
15     used[s] = true;
16     parent[s] = -1;
17
18     while (!q.empty()) {
19         int k = q.front();
20         q.pop();
21
22         for (int i = 0; i < graph.size(); i++) {
23             if (!used[i] && graph[k][i] > 0) {
24                 q.push(i);
25                 parent[i] = k;
26                 used[i] = true;
27             }
28         }
29     }
30     return (used[t]);
31 }
32
33 long long FordFulkerson(vector<vector<int>> &graph, int s, int t) {
34
35     vector<int> parent(t + 1);
36
37     long long maxFlow = 0;
38
39     while (BFS(graph, s, t, parent)) {
40         int pathFlow = 1000000001;
41         for (int i = t; i != s; i = parent[i]) {
42             int k = parent[i];
43             pathFlow = min(pathFlow, graph[k][i]);
44         }
45
46         for (int i = t; i != s; i = parent[i]) {
47             int k = parent[i];
```

```

48         graph[k][i] -= pathFlow;
49         graph[i][k] += pathFlow;
50     }
51
52     maxFlow += pathFlow;
53 }
54 return maxFlow;
55 }
56
57 int main() {
58     int n, m;
59     cin >> n >> m;
60
61     vector<vector<int>> graph(n + 1, vector<int> (n + 1));
62
63     int from, to, val;
64     for (int j = 0; j < m; ++j) {
65         cin >> from >> to >> val;
66         graph[from][to] = val;
67     }
68
69     cout << FordFulkerson(graph, 1, n) << '\n';
70
71     return 0;
72 }

```

3 Консоль

```
gregory@DESKTOP-7L8SUG4:~/labs/da$ ./x09
5 6
1 2 4
1 3 3
1 4 1
2 5 3
3 5 3
4 5 10
7
gregory@DESKTOP-7L8SUG4:~/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 изучил 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.

Список литературы

- [1] Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. *Алгоритмы: построение и анализ, 2-е издание*. — Издательский дом «Вильямс», 2007. Перевод с английского: И. В. Красиков, Н. А. Орехова, В. Н. Романов. — 1296 с. (ISBN 5-8459-0857-4 (рус.))
- [2] *Сортировка подсчётом* — *Википедия*.
URL: http://ru.wikipedia.org/wiki/Сортировка_подсчётом (дата обращения: 16.12.2013).
- [3] Список использованных источников оформлять нужно по ГОСТ Р 7.05-2008