

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: М. В. Леухин
Преподаватель: А. А. Кухтичев
Группа: М8О-306Б-20
Дата:
Оценка:
Подпись:

Москва, 2022

Лабораторная работа №7

Задача:

Задан взвешенный неориентированный граф, состоящий из n вершин и m ребер. Вершины пронумерованы целыми числами от 1 до n . Необходимо найти длину кратчайшего пути из вершины с номером *start* в вершину с номером *finish* при помощи алгоритма Дейкстры. Длина пути равна сумме весов ребер на этом пути. Граф не содержит петель и кратных ребер.

1 Описание

Алгоритм Дейкстры — алгоритм на графах, используемый для нахождения кратчайших путей от одной вершины графа до всех остальных. Работает только на графах без рёбер с отрицательными весом.

Идея алгоритма заключается в следующем. Заведём массив из n элементов — на месте с индексом i лежит значение кратчайшего расстояния от стартовой вершины до вершины с номером i . В начальный момент все элементы этого массива равны некому числу INF, заведомо большему, чем длина пути, который можем получить в ходе алгоритма. Далее начинается 1 итерация, на которой рассматриваются все возможные пути длины 1 из стартовой вершины — для этого используем алгоритм обхода в ширину. Попад в очередную вершину мы знаем длину пути, по которому мы в неё пришли — сравниваем с соответствующим значением в массиве кратчайших расстояний и обновляем в нём значение, если полученный путь оказался короче. Таким образом мы переберём все возможные пути для всех вершин.

```
1 |
2 | #include <bits/stdc++.h>
3 |
4 | using namespace std;
5 |
6 | const long long INF = 1e18;
7 |
8 | struct Edge {
9 |     int to;
10 |    long long cost;
11 | };
12 |
13 | using Matrix = vector<vector<Edge>>;
14 |
15 | int main() {
16 |
17 |     int n, m, start, finish;
18 |     cin >> n >> m >> start >> finish;
19 |     start--;
20 |     finish--;
21 |     Matrix g(n);
22 |     for (int i = 0; i < m; ++i) {
23 |         int a, b, c;
24 |         cin >> a >> b >> c;
25 |         a--;
26 |         b--;
27 |         g[a].push_back({b, c});
28 |         g[b].push_back({a, c});
29 |     }
30 |
31 |     vector<long long> d(n, INF);
```

```

32     d[start] = 0;
33
34     queue<int> q;
35     q.push(start);
36     while (!q.empty()) {
37         int v = q.front();
38         q.pop();
39         for (int j = 0; j < g[v].size(); j++) {
40             int to = g[v][j].to;
41             long long cost = g[v][j].cost;
42             if (d[v] + cost < d[to]) {
43                 d[to] = d[v] + cost;
44                 q.push(to);
45             }
46         }
47     }
48
49     if (d[finish] != INF) {
50         cout << d[finish];
51     } else {
52         cout << "No solution";
53     }
54
55 }

```

2 Консоль

```
C:/Users/leyhi/CLionProjects/contest/cmake-build-debug/lab9.exe
5 6 1 5
1 2 2
1 3 0
3 2 10
4 2 1
3 4 4
4 5 5
>8
```

3 Сложность алгоритма

В реализации алгоритма используется алгоритм обхода в ширину, который имеет сложность $O(n + t)$, где n — количество вершин в графе и t — количество рёбер, при этом больше никаких действий, влияющих на сложность, в нём нет, потому итоговая сложность алгоритма Дейкстры также $O(n + t)$.

4 Выводы

Выполнив девятую лабораторную работу по курсу «Дискретный анализ», я познакомился с различными алгоритмами на графах и научился реализовывать алгоритм Дейкстры для определения кратчайшего пути между любыми двумя вершинами в графе.

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] *Алгоритм Дейкстры* — *Википедия*.
URL: <https://ru.wikipedia.org/wiki/Алгоритм-Дейкстры> (дата обращения: 16.11.2022).