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: Ибрагимов Р. Р.
Группа: М8О-303Б-22
Вариант: 5
Преподаватель: Макаров Н.К.
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2024

Постановка задачи
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 Беллмана-Форда. Длина пути равна сумме весов ребер на этом пути. Обратите внимание, что в данном варианте веса ребер могут быть отрицательными, поскольку алгоритм умеет с ними работать. Граф не содержит петель, кратных ребер и циклов отрицательного веса

Формат ввода
В первой строке заданы 1<= n <= 3 * 10^5 и 1 <= m <= 3 * 10^5, 1 <= start < n и 1 <= finish < n. В следующих m строках записаны ребра. Каждая строка содержит три числа – номера вершин, соединенных ребром, и вес данного ребра. Вес ребра – целое число от -10^9 до 10^9.

Формат вывода
Необходимо вывести одно число – длину кратчайшего пути между указанными вершинами. Если пути между указанными вершинами не существует, следует вывести строку "No solution" (без кавычек).

Метод решения
По условию будем использовать алгоритм Беллмана-Форда для поиска кратчайшего пути в графе между парой вершин.
Сначала инициализируется массив расстояний, где начальной вершине присваивается значение 0, а всем остальным — бесконечность. Затем выполняется n-1 итерация, где n — количество вершин. На каждой итерации производится релаксация ребёр: если через текущую вершину расстояние до конечной можно улучшить, это расстояние обновляется. Если на какой-то итерации изменения не происходят, алгоритм завершается досрочно.
После выполнения проверяется, достижима ли конечная вершина: если расстояние до нее равно бесконечности, путь отсутствует, иначе в массиве расстояний хранится длина кратчайшего пути.
Алгоритм работает за O(n * m), где m — количество ребер. Также он работает правильно, если в графе есть отрицательные рёбра (но при этом нет отрицательных циклов).

Исходный код
```
#include <iostream>
#include <vector>
#include <limits>

struct Edge {
    size_t u, v;
    int weight;
```

```
};

long long bellmanFord(size_t n, size_t m, size_t start, size_t
finish, std::vector<Edge>& edges) {
    std::vector<long long> distance(n + 1,
std::numeric_limits<long long>::max());
    distance[start] = 0;
    std::vector<int> parent(n + 1, -1);

    bool updated;

    for (size_t i = 1; i < n; ++i) {
        updated = false;

        for (size_t j = 0; j < m; ++j) {
            size_t u = edges[j].u;
            size_t v = edges[j].v;
            int weight = edges[j].weight;

            if (distance[u] != std::numeric_limits<long
long>::max() && distance[u] + weight < distance[v]) {
                distance[v] = distance[u] + weight;
                parent[v] = u;
                updated = true;
            }
        }

        if (!updated) {
            break;
        }
    }

    if (distance[finish] == std::numeric_limits<long long>::max())
{
        return -1;
    }

    return distance[finish];
}

int main() {
    std::ios::sync_with_stdio(false); std::cin.tie(0);
std::cout.tie(0);

    size_t n, m, start, finish;
    std::cin >> n >> m >> start >> finish;

    std::vector<Edge> edges(m);
    for (size_t i = 0; i < m; ++i) {
        std::cin >> edges[i].u >> edges[i].v >> edges[i].weight;
    }
```

```
    long long result = bellmanFord(n, m, start, finish, edges);

     if (result == -1) {
     std::cout << "No solution\n";
     }
     else {
          std::cout << result << '\n';
     }
    return 0;
}
```

Тесты

Номер теста	Ввод	Вывод
1	5 6 1 5 1 2 2 1 3 0 3 2 -1 2 4 1 3 4 4 4 5 5	5
2	5 4 1 5 1 2 1 2 3 1 3 4 1 4 5 1	4
3	4 4 1 4 1 2 1 2 3 1 3 4 1 1 4 1	1

Выводы

В ходе выполнения данной лабораторной работы был реализован алгоритм Беллмана-Форда. Я вспомнил основные моменты работы с графами, способы их представления. Эта лабораторная в какой-то степени показалась проще предыдущих.