Лабораторная работа № 9 по курсу дискретного анализа: Поиск кратчайшего пути алгоритмом Беллмана-Форда

Выполнил студент группы 08-308 Иванов Андрей

Условие

1. Общая постановка задачи. Задан взвешенный неориентированный граф, содержа-щий n вершин и m рёбер. Требуется найти длину кратчайшего пути из вершины с номером start в вершину с номером finish при помощи алгоритма Беллмана-Форда.

2. Вариант задания. Граф содержит отрицательные веса рёбер, не имеет петель, кратных рёбер и циклов отрицательного веса. Если пути между указанными вер-шинами не существует, необходимо вывести строку «No solution».

Метод решения

Используется алгоритм Беллмана-Форда. Алгоритм проходит через все рёбра n — 1 раз, обновляя значения кратчайшего пути. Если в процессе дальнейших итераций происхо-дят изменения, это указывает на наличие отрицательного цикла. Все рёбра считыва-ются и хранятся в структуре Edge.

Описание программы

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

using namespace std;

const long long INF = numeric limits<long long>::max();

struct Edge
 {int u, v;
    long long weight
:};


int main() {
    int n, m, start ,cin >> n >> m >>


finish;
start >> finish;
```

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


                                    vector<long long>distance[start] =
distance(n + 1, INF);0;


    for (int i = 1; i <= n —
        1; i++) {bool updated =
        false;
        for (const Edge& edge : edges) {
            if (distance[edge.u] != INF && distance[edge.u] + edge.
                weight < distance[edge.v] = distance[edge.u] + edge.
                weight;
                updated
            = true;}
        }
        if (!updated
    ) break;}

    for (const Edge& edge : edges) {
        if (distance[edge.u] != INF && distance[edge.u] + edge.
            weight < discout << "No␣solution" << endl;
            r
        eturn
        0;}
    }

    if (distance[finish] ==
        INF) { cout << "No␣␣
        solution" << endl;
    } else {
        cout << distance[finish]
    << endl;}

    r
eturn
0;}
```

Дневник отладки

- Первый тест: неверный ответ (ошибка в инициализации массива distance).
- Второй тест: ошибка времени выполнения (цикл отрицательного веса не обраба-тывался корректно).

- Третий тест: программа успешно завершила работу.

Тест производительности

- 10 вершин, 15 рёбер: время выполнения 257 мс.
- 100 вершин, 200 рёбер: время выполнения 3.5 секунды.
- 1000 вершин, 5000 рёбер: время выполнения 9.8 секунд.

Недочёты

При обработке больших графов возможны задержки, связанные с высокой сложностью алгоритма $O(n \cdot m)$.

Выводы

Алгоритм Беллмана-Форда позволяет находить кратчайшие пути даже при наличии от-рицательных рёбер, но требует осторожности при обработке графов с отрицательными циклами. Задание продемонстрировало важность оптимизации и аккуратной работы с граничными случаями. Работа над этой лабораторной помогла закрепить знание рабо-ты с графами и анализа сложности алгоритмов.