

Санкт-Петербургский национальный исследовательский университет ИТМО

Факультет систем управления и робототехники

Алгоритмы и структуры данных

Отчёт по лабораторной работе №17 (1450)

Преподаватель: Тропченко А. А.

Выполнил: Марухленко Д. С.

Группа: R3235

Санкт Петербург, 2021г.

1. Цель работы

Решить задачу №1450 на платформе Timus Online Judge

<https://acm.timus.ru/problem.aspx?space=1&num=1450>

2. Задача

Вступление

Большими неприятностями обернулся прошедший год для Государства Российского. То неурожай, то птичий грипп, то вечные споры хозяйствующих субъектов... А тут ещё и Президент задумал, наконец, собрать средства на покупку новой балалайки и ручного медведя для своего двоюродного племянника. Все эти факторы (в особенности, конечно, последний) сильно ударили по экономике государства. Посоветавшись со своими друзьями в валенках и ушанках, Президент решил воспользоваться традиционным методом укрепления национального бюджета - увеличением налога на транспортировку газа.

Условие

Сеть российских газопроводов представляет собой N перекачивающих станций, некоторые из которых соединены газопроводами. Для каждого из M газопроводов известны номера станций $A[i]$ и $B[i]$, которые он соединяет, и его прибыльность $C[i]$, т.е. то количество долларов, которое будет ежесуточно приносить в виде налогов перекачка газа по этому газопроводу. Каждая пара станций соединена не более чем одним газопроводом. Сеть была построена советскими инженерами, которые точно знали, что газ поставляется из месторождений Украины в Сибирь, а не наоборот. Поэтому все газопроводы являются односторонними, т.е. для каждого газопровода перекачка газа возможна только в направлении из станции с номером $A[i]$ на станцию с номером $B[i]$. Более того, для любых двух станций X и Y верно, что если возможна перекачка газа из X на Y (возможно, через промежуточные станции), то обратная перекачка из Y на X невозможна. Известно, что газ поступает на начальную станцию с номером S и отгружается потребителям на конечной станции с номером F . Президент потребовал от Правительства указать маршрут (т.е. линейную последовательность попарно соединённых газопроводами станций) перекачки газа из начальной станции на конечную, причём прибыльность этого маршрута должна быть максимальной. Под прибыльностью маршрута понимается суммарная прибыльность входящих в него газопроводов. К сожалению, Президент не учёл того факта, что многие газопроводы изначальной сети уже давно прекратили существование, в результате чего может оказаться, что перекачка газа из начальной станции на конечную вообще невозможна...

Ограничение времени: 0.5 секунды

Ограничение памяти: 64 МБ

Исходные данные

Первая строка содержит целые числа N ($2 \leq N \leq 500$) и M ($0 \leq M \leq 124750$). Каждая из следующих M строк содержит целые числа $A[i]$, $B[i]$ ($1 \leq A[i], B[i] \leq N$) и $C[i]$ ($1 \leq C[i] \leq 10000$) для соответствующего газопровода. Последняя строка содержит целые числа S и F ($1 \leq S, F \leq N$; $S \neq F$)

Результат

Если искомый маршрут существует, выведите его прибыльность. Иначе выведите "No solution".

Пример

Исходные данные	Результат
6 7 6 5 10 1 4 11 1 2 4 3 1 5 2 4 5 6 3 1 6 1 3 6 4	17

3. Материалы работы

3.1. Объяснение алгоритма

Задача сводится к использованию алгоритма Дейкстры на графах. Для его реализации создадим массив структур *pipe_line*, содержащих информацию обо всех путях. После – создадим массив размером $n+1$ («+1» чтобы не изменять индексы-номера станций) для записи результатов работы алгоритма. Сам алгоритм модифицируем для нахождения не «минимального» пути, как в оригинале, а «максимального»

3.2. Код программы.

```
1. #include <iostream>
2. using namespace std;
3.
4. struct pipe_line {
5.     int a, b;
6.     int c;
7. };
8.
9. int main() {
10.     int n, m, s, f;
11.     cin >> n >> m;
12.     pipe_line pipes[m];
13.     int res[n+1];
14.     fill_n(res, n+1, -1);
15.
16.     for (int i = 0; i < m; i++)
17.         cin >> pipes[i].a >> pipes[i].b >> pipes[i].c;
18.     cin >> s >> f;
19.
20.     res[s] = 0;
```

```

21.     for (int i = 1; i < n; i++)
22.         for (int j = 0; j < m; j++)
23.             if (res[pipes[j].a] != -1 && res[pipes[j].b] < res[pipes[j].a] +
                pipes[j].c)
24.                 res[pipes[j].b] = res[pipes[j].a] + pipes[j].c;
25.
26.     if (res[f] != -1) cout << res[f];
27.     else cout << "No solution";
28.     return 0;
29. }

```

4. Результат выполнения и ссылка на репозиторий GitHub

9269556	21:24:11 15 мар 2021	Daniil Marukhlenko	1450	G++ 9.2 x64	Accepted		0.437	1 852 КБ
-------------------------	-------------------------	------------------------------------	----------------------	----------------	----------	--	-------	----------

https://github.com/japersik/algorithms_and_data_structures/



5. Вывод

Работа выполнена, задача решена, полученный алгоритм, основанный на алгоритме Дейкстры, соответствует требованиям проверки программы.