



Национальный исследовательский университет ИТМО
(Университет ИТМО)

Факультет систем управления и робототехники

Дисциплина: Алгоритмы и структуры данных
Отчет по практической работе (1450 задача).

Студент:
Евстигнеев Дмитрий
Группа: *R3242*
Преподаватель:
Тропченко Андрей Александрович

Санкт-Петербург
2021

Цель: написать программу для решения задачи №1450 на [сайте Timus Online](https://timusonline.ru/)

Задача:

1450. Российские газопроводы

Ограничение времени: 0.5 секунды

Ограничение памяти: 64 МБ

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.

Задача

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 ежедневно 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...

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

Принято системой (JUDGE_ID: 231802FR):

ID	Дата	Пользователь	Задача	Версия	Результат проверки	Теги
9376849	12:22:31 24 май 2021	Dmitry Evstigneev	1450. Российские газопроводы	G++ 9.2 x64	Accepted	

Решение на языке C++:

```
#include <stdio.h>

#include <vector>
#include <queue>

const int inf = -500*10000;

struct node
{
    std::vector<std::pair<int, int>> e;
    int d;
    int in;

    node() : in(0), d(inf) {}
};

node nodes[501];

int main()
{
    int N, M, S, F;
    scanf("%d %d", &N, &M);
    std::queue<node*> q;

    for(int i = 0; i < M; i++)
    {
        int x, y, c;
        scanf("%d %d %d", &x, &y, &c);
        nodes[x].e.push_back(std::make_pair(y, c));
        nodes[y].in++;
    }
}
```

```

    }
    scanf("%d %d", &S, &F);
    nodes[S].d = 0;

    for(int i = 1; i <= N; i++)
        if(nodes[i].in == 0)
            q.push(nodes + i);
    while(!q.empty())
    {
        auto nod = q.front();
        q.pop();
        for(auto it = nod->e.begin(); it < nod->e.end(); it++)
        {
            nodes[it->first].d = std::max(nodes[it->first].d, nod->d + it-
>second);
            nodes[it->first].in--;
            if(nodes[it->first].in == 0)
                q.push(nodes + it->first);
        }
    }
    if(nodes[F].d >= 0)
        printf("%d\n", nodes[F].d);
    else
        printf("No solution\n");
    return 0;
}

```

Суть алгоритма:

Топологическая сортировка с оценкой на каждом рассматриваемом ребре.