



Национальный исследовательский университет ИТМО
(Университет ИТМО)

Факультет систем управления и робототехники

Дисциплина: Алгоритмы и структуры данных
Отчет по практической работе (1162 задача).

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

Санкт-Петербург
2021

Цель: написать программу для решения задачи №1162 на сайте Timus Online

Задача:

1162. Currency Exchange

Ограничение времени: 1.0 секунды

Ограничение памяти: 64 МБ

Several currency exchange points are working in our city. Let us suppose that each point specializes in two particular currencies and performs exchange operations only with these currencies. There can be several points specializing in the same pair of currencies. Each point has its own exchange rates, exchange rate of A to B is the quantity of B you get for 1A. Also each exchange point has some commission, the sum you have to pay for your exchange operation. Commission is always collected in source currency.

For example, if you want to exchange 100 US Dollars into Russian Rubles at the exchange point, where the exchange rate is 29.75, and the commission is 0.39 you will get $(100 - 0.39) * 29.75 = 2963.3975$ RUR.

You surely know that there are N different currencies you can deal with in our city. Let us assign unique integer number from 1 to N to each currency. Then each exchange point can be described with 6 numbers: integer A and B - numbers of currencies it exchanges, and real RAB, CAB, RBA and CBA - exchange rates and commissions when exchanging A to B and B to A respectively.

Nick has some money in currency S and wonders if he can somehow, after some exchange operations, increase his capital. Of course, he wants to have his money in currency S in the end. Help him to answer this difficult question. Nick must always have non-negative sum of money while making his operations.

Исходные данные

The first line contains four numbers: N - the number of currencies, M - the number of exchange points, S - the number of currency Nick has and V - the quantity of currency units he has. The following M lines contain 6 numbers each - the description of the corresponding exchange point - in specified above order. Numbers are separated by one or more spaces. $1 \leq S \leq N \leq 100$, $1 \leq M \leq 100$, V is real number, $0 \leq V \leq 10^3$.

For each point exchange rates and commissions are real, given with at most two digits after the decimal point, $10^{-2} \leq \text{rate} \leq 10^2$, $0 \leq \text{commission} \leq 10^2$.

Let us call some sequence of the exchange operations simple if no exchange point is used more than once in this sequence. You may assume that ratio of the numeric values of the sums at the end and at the beginning of any simple sequence of the exchange operations will be less than 10^4 .

Результат

If Nick can increase his wealth, output YES, in other case output NO.

Примеры

исходные данные	результат
3 2 1 10.0 1 2 1.0 1.0 1.0 1.0 2 3 1.1 1.0 1.1 1.0	NO
3 2 1 20.0 1 2 1.0 1.0 1.0 1.0 2 3 1.1 1.0 1.1 1.0	YES

Принято системой (JUDGE_ID: 231802FR):

ID	Дата	Автор	Задача	Язык	Результат проверки
9370153	03:21:22 18 май 2021	Dmitry Evstigneev	1162. Currency Exchange	G++ 9.2 x64	Accepted

Решение на языке C++:

```
#include <iostream>
#include <vector>
#include <algorithm>

struct edge
{
    int a, b;
    double r, c;
};

std::vector<edge> v;
double nodes[101];

int result(bool b)
{
    std::cout << (b ? "YES" : "NO") << "\n";
    return 0;
}

int main()
{
    int N, M, S;
    double V;
    std::cin >> N >> M >> S >> V;
    nodes[S] = V;
    for (int i = 0; i < M; i++)
    {
        int a, b;
        double rab, cab, rba, cba;
        std::cin >> a >> b >> rab >> cab >> rba >> cba;
        v.push_back({ a, b, rab, cab });
        v.push_back({ b, a, rba, cba });
    }
    for (int i = 0; i < N - 1; i++)
        for (int j = 0; j < v.size(); j++)
            nodes[v[j].b] = std::max((nodes[v[j].a] - v[j].c)*v[j].r,
nodes[v[j].b]);
    for (int i = 0; i < v.size(); i++)
        if ((nodes[v[i].a] - v[i].c)*v[i].r > nodes[v[i].b] + 0.000001)
            return result(true);
    return result(false);
}
```

Суть алгоритма:

Алгоритм Беллмана — Форда — алгоритм поиска кратчайшего пути во взвешенном графе

