Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа

по «Алгоритмам и структурам данных»

Выполнил:

Студент группы P3200

Ле Туан Зунг

Преподаватели:

Косяков М.С.

Тараканов Д.С.

Санкт-Петербург

2019

1080. Map Coloring

We consider a geographical map with N countries numbered from 1 to N ($0 < N < 99$). Write a program which determines whether it is possible to color the map only in two colors — red and blue in such a way that if two countries are connected their colors are different. The color of the first country is red.

Input

On the first line is written the number N.

On the following N lines, the i-th line contains the countries to which the i-th country is connected, the last one which is 0 and marks that no more countries are listed for country i.

If a line contains 0, that means that the i-th country is not connected to any other country, which number is larger than i.

Output

0 corresponds to red color, and 1 — to blue color. If a coloring is not possible, output the integer −1.

how to solve:

I used DFS to color those country.

We can set 1- for blue color and (-1)-for red color (instead of 0).

(Because may be not all the country are connected, we need to check each country whether it’s colored. If it’s not colored, paint it red. Starting with the first country.)

Method : PaintDFS(k,code):

Where:

- k is the number of current country
- Code is the color: 1- for blue color, (-1)-for red color.

how method works:

1. Mark this country is colored(code).
2. Determine the color of the next country.
3. For each country which is connected to k:
    a) If its color == nextColor continue;
    b) If its color != 0 then it’s painted but has the same color with curent country.
    There is no way to solve the problem. Exit
    c) Else paint it with nextColor.

1160. Network

There will be N hubs in the company. each hub must be accessible by cables from any other hub (with possibly some intermediate hubs).

a plan of hub connection, that the maximum length of a single cable is minimal.

Input

The first line contains two integer: N - the number of hubs in the network ($2 \le N \le 1000$) and M — the number of possible hub connections ($1 \le M \le 15000$).

Output

Output first the maximum length of a single cable in your hub connection plan (the value you should minimize). Then output your plan: first output P - the number of cables used, then output P pairs of integer numbers - numbers of hubs connected by the corresponding cable. Separate numbers by spaces and/or line breaks.

how to solve:

I just used "kruskal" algorithm for find minimun-spanning-tree.

First, sort all the cable according to their length from shorter to longer.

Then, following this order, add these edge until all the hubs are connected(when we added n-1 cables into the spanning tree)

1450. Russian Pipelines

Problem

N transfer station. For each of M pipelines the numbers of stations A[i] and B[i], which are connected by this pipeline, and its profitability C[i] are known.

Each two stations are connected by not more than one pipeline.

The pipelines are unidirectional. More over, if it is possible to transfer the gas from the station X to the station Y (perhaps, through some intermediate stations), then the reverse transfer from Y to X is impossible. gas arrives to the starting station number S and should be dispatched to F.

Find a route to transfer the gas from the starting to the final station. A profitability of this route should be maximal.

The gas transfer between the starting and the final stations may appear to be impossible...

Input

The first line contains the integer numbers N ($2 \le N \le 500$) and M ($0 \le M \le 124750$).

Each of the next M lines contains the integer numbers A[i], B[i] ($1 \le A[i], B[i] \le N$) and C[i] ($1 \le C[i] \le 10000$) for the corresponding pipeline.

The last line contains the integer numbers S and F ($1 \le S, F \le N; S \ne F$).

Output

If the desired route exists, you should output its profitability. Otherwise you should output "No solution".

how to solve:

I used Ford-Bellman algorithm to find the maximum profit of every station. Call the starting point is S.

The maximum number of edges in the path between X and Y is n-1.

- ⇨ So we just need to loop n-1 times, each time we go through all the egdes (a,b,w) and update best profit value of all station using this formula:

profit[b] = max(profit [b], profit [a] + w);