Stealing Railways
Problem ID: 6stealingrailways

This problem formulation is almost identical to the one written by Måns Magnusson for the course in 2017[1].

Introduction

The war is near. Soon we are expecting an attack from the right (calling it east or R*s*i* might be controversal). Soldiers will then be shipped from the base camp to the front line – the left most point – using railways. You want to steal some railways (in order to sell the metal on the black market). Luckily for you the railway system is quite oversized and can ship far more soldiers than needed. So possibly you will be able to steal some of the railways, and you want to steal as many as possible without crushing the generals plans to ship C soldiers to the left (this number C have you found somehow). How many railways can you steal without the general noticing (which he only will if he cannot ship C soldiers)?

Aims

The goals of the lab are:

- Implementing a Network-Flow-algorithm.
- Debugging your code.
- Structuring your code in a logical fashion.

Problem formulation

You are given the structure of the railway system, in the form of nodes (cities) and edges (railways connecting the cities). For each edge you will be given the capacity (the number of soldiers it can carry). Then you will be given a plan for which railways to steal in a specific order – in which you need to steal the railways if possible. If you cannot (which you cannot if the total maximal flow from the base camp to the front line is less than C) steal a railway you stop your plan and instead start selling the metal. Your task is to answer how many of the railways you can steal and what the maximal flow will be after stealing these railways.

Input

The first line consists of four integers N, M, C, P, the number of nodes, the number of edges, the number of soldiers to transfer from the basecamp (node 0) to the left (node $N-1$) and the number of railways in your plan. It is guaranteed that $2 \leq N \leq 1000, 1 \leq M \leq \frac{N(N-1)}{2}, 0 \leq P \leq M$ and that the capacity of the network before any railways are removed is at least C. Then follows M lines where the i-th line (0-indexed) consists of three integers u_i, v_i, c_i, where $0 \leq u_i, v_i < N$ are the nodes and c_i is the capacity of the railway, $1 \leq c_i \leq 100$. Note that soldiers can travel both from u_i to v_i and from v_i to u_i (the graph is undirected) but in total at most c_i soldiers. It is guaranteed that the same pair of nodes only occurs once in the input and that $u_i \neq v_i$. Then follows P lines with one integer each, where the i-th line contains the index of the i-th railway you want to steal. Note that you have to steal them in the order they are given.

Output

The output should contain one line with two space-separated integers x, f, where x is the number of railways you can steal and f is the maximal flow from node 0 to node $N-1$ after stealing these railways.

Examination and Points of Discussion

To pass the lab make sure you have:

- Have successfully implemented the algorithm with the correct time complexity.

[1] https://lu.kattis.com/problems/lu.railroads

- Have neat and understandable code.
- Have descriptive variable names.
- Have filled in the blanks in the report.
- Have run the check_solution script to validate your solution.

During the oral presentation you will discuss the follwoing questions with the lab assistant:

- What is the time complexity, and more importantly why?
- Which other (well-known) algorithmic problems can be solved using Network-Flow?
- If the capacities of the edges are very large, how can one get a different (better) time complexity?

Sample Input 1

```
3 3 10 3
0 1 10
0 2 10
1 2 10
0
2
1
```

Sample Output 1

```
2 10
```