Competitive Programming

Saarland University — Summer Semester 2020

Julian Baldus, Markus Bläser, Karl Bringmann, Marian Dietz, Simon Schwarz,
Christoph Weidenbach, Dominic Zimmer

Assignments Week 5

Deadline: **June 9, 2020 at 16:00 sharp**

Please submit solutions to the problems in our judge system, available at
`https://compro.mpi-inf.mpg.de/`.
You can find your credentials on your personal status page in our CMS.

Problem	`coronatracking`	`inequalities`	`fares`	`lufthansa`
Points	3	3	1 + 2	3
Difficulty	🌶️	🌶️🌶️	🌶️🌶️🌶️	🌶️🌶️🌶️🌶️
Time Limit	3s	1s	1s / 6s	2s
Memory Limit	2 GB	2 GB	2 GB	2 GB

Please note:

- Later, we will reopen the judge for the problems of the last weeks. However, you won't get any points for submissions of these problems.
- In the judge you can switch between the exercises of different weeks in the top-right corner.
- Your solution will be judged immediately after submitting. This may take some time, depending on the current server load.
- You can submit as many times as you want. However, don't abuse the server or try to extract the secret test cases.
- If your solution is **accepted**, you will receive the points specified in the table above.
- If you get **another verdict**, you will receive 0 points.

Corona Tracking
Problem ID: coronatracking

An important aspect in limiting the spread of the coronavirus pandemic is testing and quarantine of infected persons.

Whenever someone tests positive for SARS-CoV-2, all contact persons are also tested. To ensure containment, it has been suggested to also start quarantining and testing contacts of contacts, or even contacts of contacts of contacts and so on. However, this might be infeasible due to the limited availability of tests. Your task is to determine if testing indirect contacts is feasible by calculating the number of persons that have to be tested.

More formally, we can model our social interactions as an undirected graph. Each node represents a person, and edges between persons mean that those two persons had direct contact. If a person A had direct contact to person B, we say that they have a distance of 1. If two persons had contact with some common person but did not have contact directly, their distance will be 2. In general, two persons have distance d if the shortest path between them consists of d edges.

Please implement a program that reads the social interaction graph and outputs the number of people having distance at most d to a newly infected person s.

Input

The input consists of:

- one line with four integers
  - n ($2 \leq n \leq 10^5$), the number of nodes in the social graph
  - m ($1 \leq m \leq 10^6$), the number of edges in the social graph
  - s ($1 \leq s \leq n$), the newly infected person
  - d ($1 \leq d \leq n$), the distance we are interested in
- m lines describing the edges in the graph. Each line i contains two integers a_i ($1 \leq a_i \leq n$) and b_i ($1 \leq b_i \leq n$), meaning there is an edge between a_i and b_i.

Note that indices in the graph start at 1.

Output

Output a single integer, the number of persons that have distance at most d to the person s.

Sample Input 1	Sample Output 1
6 5 2 3 2 4 4 3 3 1 1 5 4 6	5

Sample Input 2	Sample Output 2
9 10 4 2 1 2 2 3 1 7 2 4 3 4 4 7 7 8 9 7 7 6 5 6	8

Sample Input 3	Sample Output 3
5 10 5 1 1 2 1 3 1 4 1 5 2 3 2 4 2 5 3 4 3 5 4 5	5

Inequalities

Problem ID: inequalities

Though Competitive Programming is a very enjoyable lecture, some of your friends decide to give the course Introduction to Advanced Arithmetics a try. On their last exercise sheet there was a problem that they struggled with particularly:

Given n strict inequalities between variables, find an ordering on the variables that fulfils all the inequalities. Knowing their professor, you suspect that he might have designed the exercises somewhat sloppily. Therefore, it may happen that the problem is actually unsolvable (e.g. there is no ordering on the variables that fulfils all inequalities).

Can you help them find an ordering of the variables or indicate that this is indeed impossible?

Input

The input consists of:

- one line with an integer n ($1 \leq n \leq 10^5$), where n is the number of inequalities;
- n lines describing one inequality each. Each inequality is described by:
  - one line with “$s_1 < s_2$” or “$s_1 > s_2$”, telling whether the variable s_1 is less or greater than variable s_2. s_1 and s_2 are two different variables.

A variable name consists of at most 10 letters from “A” to “Z” and “a” to “z”. A variable name does not contain spaces.

Output

Output “`impossible`” if the statements are not consistent, otherwise output “`possible`”. Additionally, if the statements are consistent, print the names of the variables in decreasing order. If multiple orderings are consistent with the input, print any of them.

Sample Input 1

```
3
a > b
b > c
a < c
```

Sample Output 1

```
impossible
```

Sample Input 2

```
3
theta > omega
omega > alpha
theta > alpha
```

Sample Output 2

```
possible
theta omega alpha
```

Dodgy Fares

Problem ID: fares

Dieter's last heist was a huge success. He and his friend Marc made a fortune from a wealthy neighbourhood. However, they need to leave this town very quickly. The once wealthy neighbours will likely not let them escape easily. Their plan is to go to one of Marc's friends, who is living in a nearby town and is willing to hide the two crooks.

Carrying a phone poses a huge risk to their anonymity. Therefore, they need to reach the hideout without the help of the internet or GPS. Luckily, Dieter bought a public transport map at the kiosk around the corner. Both Dieter and Marc have t metro tickets left. They are willing to fare-dodge the bus, but it's not possible to enter the metro without a valid ticket.

What is the minimal distance they have to travel to get to their hideout?

Subtasks

This problem contains subtasks. The solution to one subtask is usually an improvement over or adaption of the solution to a previous subtask. You should therefore think of subtasks as hints that guide you in finding a correct solution.

Due to constraints of the judge system, each subtask appears as a separate problem on the scoreboard and you have to submit your program to each of them.

- **Subtask 1** (1 point) Dieter and Marc have no metro tickets left: $t = 0$.
- **Subtask 2** (2 points) Dieter and Marc each have $0 \leq t \leq 10$ metro tickets left.

Input

The first line contains four numbers n, b, m and t ($2 \leq n \leq 100\,000$, $1 \leq b, m \leq 100\,000$, $0 \leq t \leq 10$).

The next b lines contain one bus connection each: every line consists of three integers u, v, l ($1 \leq u, v \leq n$, $1 \leq l \leq 10^9$). This means that there is a bidirectional bus connection of length l between u and v. The next m lines contain one bidirectional metro connection each, in the same format as the bus connections. Taking a metro connection requires one metro ticket (only one ticket for both Dieter and Marc is needed), regardless of its length.

Note that it may be possible that there are multiple connections between two locations. However, it is guaranteed that there are no self loops (a single connection going from a location to itself) and every location is reachable from every other location using only bus rides.

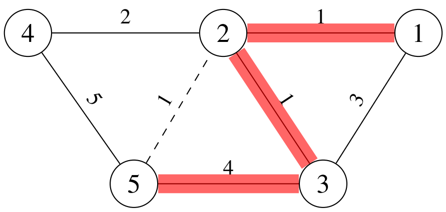
Also note that indices in the graph start at 1.

Output

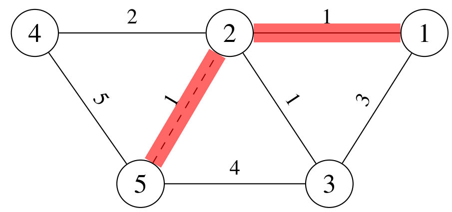
The hideout is located at location n. Dieter and Marc are currently at location 1. Print the length of a shortest path from 1 to n using at most t metro tickets.

Sample Inputs

Note that the first input can occur in both subtasks. However, the second input is not valid in subtask 1.



Visualisation of Sample Input 1



Visualisation of Sample Input 2

A solid line stands for a bus connection, while a dashed line stands for a metro connection. The connections marked red form the optimal path.

Sample Input 1

```
5 6 1 0
1 2 1
1 3 3
2 3 1
2 4 2
3 5 4
4 5 5
2 5 1
```

Sample Output 1

```
6
```

Sample Input 2

```
5 6 1 1
1 2 1
1 3 3
2 3 1
2 4 2
3 5 4
4 5 5
2 5 1
```

Sample Output 2

```
2
```

Lufthansa
Problem ID: lufthansa

You are glad that your CEO finally convinced the german politicians to grant Lufthansa sufficient funds to keep the company alive. Not only are you an enthusiastic and also particularly successful participant of programming contests, you have also been serving as the Head of Transportation of the “Time Limit Exceeded” – an international, multitier, team-based programming competition. One of your responsibilities is to book the flights for all teams around the world who would like to attend their respective regional finals. Back in the days, this was an easy job. You simply entered the departure and destination airport in one of the online platforms like *scanthesky.com*, *podomo.com*, or *dooswoo.com*, and picked the cheapest connection.

The problem is that students these days tend to be really picky. They do not want to have stopovers at airports that do not offer a certain level of luxury. According to them a must-have for all airports is a free WiFi-connection. Besides that they would like to have enough power plugs to charge all their electronic devices. If the airport also provides free pizza, snacks, and soft drinks, then this is also highly appreciated.

To meet the demands of the students, you have already classified the airports according to their level of luxury by assigning each of them an integer value a_i that represents the airport’s *luxury class*. The lower the value of a_i, the higher the luxury of the airport. Your job is now to find for each team the cheapest flight connection between cities x_i and y_i such that the particular team only travels through airports whose luxury class is at most ℓ_i.

Input

Each test case starts with a line containing three space-separated integers n, m, and k, where n is the number of cities, m is the number of flight connections, and k is the number of team requests. One line follows containing n space-separated integers $a_1, a_2, \ldots, a_n$ where a_i is the luxury class of the airport in city i. m lines follow, each containing three space-separated integers x_i, y_i, and c_i, denoting that there is a direct flight from city x_i to city y_i that costs c_i Euro. k lines follow each containing three space-separated integers x_i, y_i, and ℓ_i denoting a team’s request for a flight connection of minimal costs from the airport in city x_i to the airport in y_i such that all intermediate airports have luxury class at most ℓ_i.

Note that indices in the graph start at 1.

Output

Print k lines. In line i, print the minimal price to be paid when traveling between airports defined in the i-th request while also taking the luxury class of the i-th request into account. If no flight connection exists that satisfies the request, output `impossible` in the corresponding line.

Constraints

- $2 \leq n \leq 500$
- $1 \leq m \leq 10^5$
- $1 \leq k \leq 10^5$
- $1 \leq a_i \leq 1000$ for all $1 \leq i \leq n$.
- $1 \leq c_i \leq 1000$ for all $1 \leq i \leq m$.
- $\max\{a_{x_i}, a_{y_i}\} \leq \ell_i \leq 1000$ for all $1 \leq i \leq k$.

Sample Input 1

```
3 4 5
3 1 1
3 2 4
3 1 1
1 2 2
1 2 3
3 2 1
3 2 2
3 2 3
3 2 4
2 3 2
```

Sample Output 1

```
4
4
3
3
impossible
```

Sample Input 2

```
5 5 6
2 2 4 5 5
1 3 4
1 4 9
3 2 5
3 4 3
4 2 1
1 2 2
1 2 3
1 2 4
1 2 5
1 2 6
1 5 6
```

Sample Output 2

```
impossible
impossible
9
8
8
impossible
```

Sample Input 3

```
4 7 6
1 4 4 5
1 2 3
4 2 1
4 3 2
1 2 1
4 1 4
4 3 2
4 3 5
4 2 5
2 3 5
2 3 5
4 2 5
3 4 5
2 4 5
```

Sample Output 3

```
1
impossible
impossible
1
impossible
impossible
```

Sample Input 4	Sample Output 4
5 6 3 1 1 3 4 4 2 4 2 1 4 3 1 2 1 5 3 4 3 1 2 3 4 2 3 4 5 1 4 5 4 3 4	2 3 impossible