Competitive Programming

Saarland University — Summer Semester 2022

Markus Bläser, Karl Bringmann, Martin Bromberger, Christoph Weidenbach
Julian Baldus, Yasmine Briefs, Simon Schwarz, Dominic Zimmer

Assignments Week 5

Deadline: **May 19, 2022 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	`bars`	`inequalities`	`fares`		`lufthansa`
Points	3	3	2	2	2
Difficulty	🌶️	🌶️🌶️	🌶️	🌶️🌶️🌶️	🌶️🌶️🌶️🌶️
Time Limit	3s	1s	1s	6s	2s
Memory Limit	2 GB	2 GB	2 GB	2 GB	2 GB

Please note:

- 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.

Bars

Problem ID: bars

It's Friday evening and you and Dieter Schlau are sitting in your favorite pub *The Drunken Knurd*. You are drinking, playing cards and meeting new people. Dieter met some of his classmates and they start talking about bioinformatics. This conversation keeps going on for hours. So long in fact that you – a computer science major – are starting to get bored and decide to leave the pub.

You and Dieter need to return home eventually to interview a new possible flatmate. Until then, you have d hours to spare in which you will be drinking in the local pubs. Both Dieter and you will stay at least one hour in any pub you visit before you go on to another pub.

You haven't seen Dieter for several bars and suddenly you hear the bartender yell "Closing time, drink up, guys!". Time sure flies when you are drunk. Also, you have no idea where Dieter is. You are afraid something happened to him, or worse: he could miss the interview. You start looking for him in all the – by now closed – pubs in the city.

You last met him in the initial bar *The Drunken Knurd* and you are certain that he is still in one of the pubs in town. How many different bars could he have reached within the d hours given?

Input

The input describes a graph which forms the map of all pubs in town. In the graph, all pubs have a unique number associated with them.

The first line contains four integers n, m, s and d ($2 \leq n \leq 10^5, 1 \leq m \leq 10^6, 1 \leq s, d \leq n$), the number of pubs in town, the number of paths between pubs, the *Drunken Knurd*'s unique number and the amout of time d you have.

Then, m lines follow, each describing the paths between two pubs. Each of those m lines contains two integers a_i and b_i ($1 \leq a_i, b_i \leq n$), meaning that you pubs a_i and b_i are reachable from each other.

Output

Output the number of pubs Dieter can have possibly reached in the d hours of drinking.

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

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

Sample Input 3

```
9 10 4 2
1 2
2 3
1 7
2 4
3 4
4 7
7 8
9 7
7 6
5 6
```

Sample Output 3

```
8
```

Sample Input 4

```
5 10 5 1
1 2
1 3
1 4
1 5
2 3
2 4
2 5
3 4
3 5
4 5
```

Sample Output 4

```
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
```

Fares

Problem ID: fares
Time limit: 1 second

This year, the government decided to offer 9€-tickets for everyone.[1] While they provide a great way to travel cheaply, holders of this ticket are only allowed to take regional trains. In contrast, long-distance trains still need an extra paid ticket for each ride.

You and your friends, of course, want to use this unique opportunity to travel to Sylt for a great party. In order to arrive on time,[2] you are willing to pay for at most t long-distance tickets to speed up your journey.

What is the minimal time you have to travel to get to Sylt?

Sylt, before the party has started.

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** (2 points) You are on a budget. Thus, you do not take any long-distance train, i.e. $t = 0$.
- **Subtask 2** (2 points) You may take $0 \leq t \leq 10$ long distance trains.

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 regional train 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 regional train connection that takes l hours between u and v. The next m lines contain one bidirectional long-distance train connection each, in the same format as above. Taking a long-distance train connection requires one ticket, 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 regional trains.

Also note that indices in the graph start at 1.

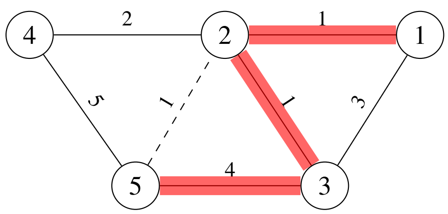
Output

Sylt is located at location n. You are currently at location 1. Print the length of a shortest path from 1 to n using at most t long-distance train connections.

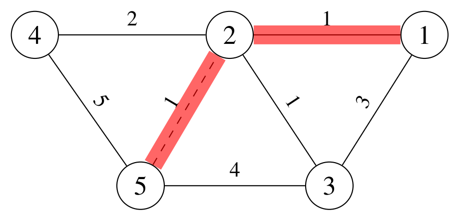
[1] Notably, also the *Pöbel* can buy such a cheap ticket.
[2] The party starts on June 1st.

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 regional train connection, while a dashed line stands for a long-distance train connection. The connections marked red form the optimal path.

Sample Input 1	Sample Output 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	6

Sample Input 2	Sample Output 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	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

```
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
```

Sample Output 4

```
2
3
impossible
```