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 7

Deadline: **June 23, 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	`labyrinth`	`marriage`	`reelection`	`snow`
Points	3	3	3	3
Difficulty	🌶🌶🌶🌶	🌶🌶	🌶🌶	🌶🌶🌶
Memory Limit	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.

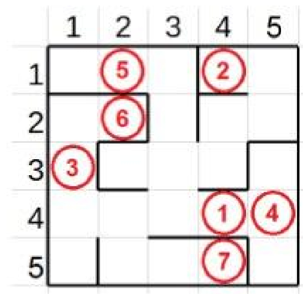
Labyrinth

Problem ID: labyrinth
Time limit: 10 seconds

Luna loves board games. Her favorite game is called labyrinth. In this game, the player moves a game piece through a labyrinth that is $h \times w$ cells in size. Each cell is exactly 1×1 units big and connected to at least one neighboring cell. Also, there is exactly one path between any two cells. Altogether, the game field forms a rectangular labyrinth without any loops.

Luna has already thought of a strategy. She has marked m cells on the game field, which she wants to visit with her game piece in exactly this order. In one turn, Luna can move her game piece from her current cell to any adjacent cell, as long as there is no wall in between.

Now she wonders how many turns she will have to make to visit all the marked cells in the correct order (since there are no loops, she will always take the direct path from one marked cell to the next). The first marked cell indicates where the game piece is at the beginning, and the last marked cell indicates where it must be at the end.



Visualization of Sample Input 2

Input

The first line contains two integers h and w ($2 \leq h, w \leq 1000$), the height and width of the labyrinth.

$h + 1$ lines follow, describing the game field. The description always follows these rules:

- Each of these lines contains $2 \cdot w + 1$ characters.
- Columns with odd index (starting at 1) contain either walls ’|’ or spaces (indicating that there is no vertical wall). Columns with even index contain either walls ’_’ or spaces (indicating that there is no horizontal wall).
- The first row describes the upper wall of the labyrinth which always consists only of horizontal walls. Every subsequent row describes a row of cells.
- A cell is located at every even column index. Its left and right walls are located at the directly neighboring odd column indices respectively. Its upper wall is located at the same column index but one row above and its lower wall can be found at its own position. If a wall is missing, the corresponding position contains a space instead.

After the description of the labyrinth, an integer m ($2 \leq m \leq 10^4$) describing the number of marked cells is given. m lines describing the marked cells follow. The i-th of them contains two integer coordinates x and y ($1 \leq x \leq h$ and $1 \leq y \leq w$) indicating the i-th marked cell on the game field. The first pair of coordinates is the start point of the game piece, the last pair the end point. Cells may appear multiple times but never twice or more in a row. $(1, 1)$ is the top left cell and (h, w) is the bottom right cell.

It is guaranteed that the labyrinth itself is enclosed. Furthermore it is guaranteed that exactly one path exists between any two cells.

Output

Output one integer, the number of turns needed to visit all the marked cells in the correct order.

Technical Note

Note that there is a space before and after the first line of the description of the labyrinth (the upper wall). If you however copy the sample inputs from this PDF, these spaces will be missing. We suggest inserting them manually or dowloading the sample input files from the judge.

Sample Input 1

```
2 6
 _ _ _ _ _ _ 
|  _ _ _ _ _|
|_ _ _ _ _ _|
5
1 5
1 1
1 6
1 1
1 5
```

Sample Output 1

```
18
```

Sample Input 2

```
5 5
 _ _ _ _ _ 
|_ _  |_  |
|  _| |  _|
| |_   _| |
|    _ _  |
|_|_ _ _|_|
7
4 4
1 4
3 1
4 5
1 2
2 2
5 4
```

Sample Output 2

```
43
```

Marriage

Problem ID: marriage
Time limit: 10 seconds

Finally, Dieter made it back to Saarbrücken. During his bar tour in Zweibrücken, Dieter unfortunately drank away so much money that he is now broke. Therefore, Dieter needs a way to get some easy money. He decides that marrying a wealthy woman should be the way to go. However, since he has returned to the Saarland, he has the uneasy feeling that he should better watch out who to marry - otherwise, he might marry one of his cousins.

Dieter has already compiled a list of the women that want to get married. Furthermore, he has obtained the family tree of those women. Conveniently, it is actually a tree, and in particular all people in his list descend from the first person on his list and everyone has only one direct ancestor. We say that a person A descends from person B if $B = A$ or if B is either mother, grandmother, grand-grandmother, ... of A.

To avoid getting in trouble with the law again, Dieter wants to find the degree of the relationship between two arbitrary persons q_1 and q_2. Dieter is interested in two properties:

- The *lowest common ancestor*, e.g. the person p lowest in the tree such that q_1 descends from p and q_2 descends from p. Furthermore, there should be no person $p' \neq p$ which descends from p with q_1 and q_2 descending from p'.
- The *distance* of the persons q_1 and q_2. The distance is the length of the shortest link between the two persons. For example, a person A and her mother B have distance 1. A person has distance 2 to her grandmother, distance 2 to her siblings and, furthermore, distance 4 to her cousins. A person has distance 0 to themself.

Input

The first line of the input contains an integer n and q ($1 \leq n, q \leq 2 \cdot 10^5$).

The second line contains n space-separated strings l_i representing the names of the persons. All of them consist of lower- and uppercase letters and are less than 20 characters long. It is guaranteed that all names are distinct.

The third line contains $n - 1$ names $s_2, s_3, \ldots, s_n$, meaning that s_i is the mother of l_i. It is guaranteed that all names are contained in the first line and that there are no inconsistencies in the input (e.g. everyone is a descendant of the first person and there are no cycles).

The following q lines each contain two names q_i^1, q_i^2, representing a pair of persons Dieter is interested in.

Output

Print q lines. In the i-th line, print the name of the lowest common ancestor and the distance between q_i^1 and q_i^2.

Sample Input 1	Sample Output 1

`6 4`
`Gertrude Annegret Elisabeth Lea Jule Dieter`
`Gertrude Gertrude Elisabeth Annegret Annegret`
`Dieter Jule`
`Dieter Annegret`
`Dieter Lea`
`Jule Gertrude`
	`Annegret 2`
	`Annegret 1`
	`Gertrude 4`
	`Gertrude 2`

Sample Input 2	Sample Output 2
6 3 Sissi Liesbeth Gertrude Annegret Dieter Cecilia Cecilia Liesbeth Gertrude Annegret Sissi Liesbeth Liesbeth Dieter Sissi Annegret Cecilia	Liesbeth 0 Sissi 5 Cecilia 3

Reelection

Problem ID: reelection
Time limit: 2 seconds

You are the president of a great country. Getting into the office the first time was not that difficult – you just invested a whole lot of money into promoting yourself. However, contrary to your expectations you have nothing to do the whole day, giving you more time to fly to your golf courses every day. That's why you want to candidate for the next election this fall as well. Unfortunately, the candidate of another political party wants to replace you in the office.

You have one advantage: Currently, you are the president and he is not. Therefore you could make the citizens happy by doing what is written in your job description for the first time in your political career. Hopefully this will convince the people that you should stay president.

Your country consists of n cities connected by m bidirectional streets. The citizens like some of the streets (e.g. those which have bicycle lanes and have many trees), while they hate others (e.g. streets passing directly through their home). Therefore every street has a corresponding quality between -10^9 and 10^9, indicating whether it is being liked by the citizens (positive value), or not (negative value).

You have a lot of money remaining (because you don't have to pay for the presidential aircraft yourself) and can decide which streets to keep and which streets to remove. The goal is to maximize the total quality (sum of qualities of all streets which are being kept), with the constraint that every city stays reachable from every other city after the removal.

Input

The first line contains two integers n and m ($1 \leq n \leq 10^5$, $1 \leq m \leq 2 \cdot 10^5$), the number of cities and the number of streets.

The following m lines each contain three integers s_i, d_i, v_i ($1 \leq s_i, d_i \leq n$, $-10^9 \leq v_i \leq 10^9$), indicating that there is a bidirectional street between s_i and d_i of quality v_i.

It is guaranteed that it is possible to reach each city from each other city using only streets given in the input.

Output

Output a single integer: The maximum total quality.

Sample Input 1

```
4 6
1 2 15
1 3 3
2 3 1
2 4 -1
3 4 -2
1 1 5
```

Sample Output 1

```
23
```

Sample Input 2

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

Sample Output 2

```
-7
```

Snow

Problem ID: snow
Time limit: 2 seconds

Dieter needs a break. The weekend on Sylt was nice, but the 20-hour journey in overcrowded regional trains and the all-night parties were quite exhausting. So he decided to go on a winter vacation.

He is located in a remote valley where there are n villages connected by m roads. Unfortunately, it has started to snow heavily. As time goes by, more and more roads are covered in snow to the point that they are no longer passable. Unfortunately, since he can't get anywhere this way, there is little to do and Dieter gets bored.

To pass the time, he asks himself the following question: He wants to know, after each additional snow-covered road, into how many components the villages have been divided.

Input

The first line contains three integers n, m and k ($1 \leq n \leq 10^5$, $1 \leq m \leq 2 \cdot 10^5$, $1 \leq k \leq m$), the number of villages, the number of roads and the number of snow-covered roads. The villages are numbered $1, \ldots, n$.

The following m lines each contain two integers a and b ($1 \leq a, b \leq n$), indicating that there is a bidirectional road between villages a and b. Each road is between two different villages and there is at most one road between two villages.

Finally, there are k lines describing the snow-covered roads. Each of these k lines contain two integers a and b: The road between villages a and b is covered by snow and cannot be used anymore. You are guaranteed that there is a road between a and b in this case and that this road wasn't covered by snow before.

Output

Output k integers: the i-th of them should be the number of components after the i-th snow-covered road.

Sample Inputs

In the first sample, initially all villages are connected. After the road between villages 3 and 4 is covered by snow, the villages are divided into the two components $\{1, 2, 3\}$ and $\{4, 5\}$, so the first output is 2. The second snow-covered road (the one between 2 and 3) doesn't change the number of components. The third snow-covered road (the one between 4 and 5) divides the component $\{4, 5\}$ into two components, so the resulting components are $\{1, 2, 3\}$, $\{4\}$, $\{5\}$ and the third output is 3.

Sample Input 1

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

Sample Output 1

```
2 2 3
```

Sample Input 2

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

Sample Output 2

```
1 2 2 3 4
```