Social Network

ConnectIn company operates a social network. Each member of the network is idetified by a unique and permanent ID. He also has a status which expresses his trustfulness. The status can change over time, based on activities carried out by the member within the network. A member can invite other people to become new members. Each invitation contains an offered status, which is not higher than the current status of the member who sends the invitation. If a person accepts an ivitation, he becomes a member and receives the offered status and a random ID, different from all the already generated IDs.

The company analyzes behavioral patterns in the network. In particular, they study sequences of statuses $s_1,...,s_k$ for which there are members with IDs $M_1,...,M_{k+1}$ such that member with ID M_i sent an invitation with initial status s_i to a person who became member with ID M_{i+1} (for all $i=1,...,k$). Let us denote such sequences as *chained sequences of statuses*. The company aims to implement a search tool which takes a string describing a pattern and detects the longest chained sequence of statuses matching the pattern. The patterns are strings over $\Sigma \cup \{*\}$ where Σ is the set of all statuses and $*$ is a special character, not contained in Σ. It also holds that each occurrence of $*$ has to be preceded by a character from Σ. A chained sequence of statuses matches the pattern if and only if the sequence can be generated from the pattern as follows: each substring **x***, where **x** is from Σ, is replaced by an arbitrary number of occurrences of **x**, including zero number of occurrences.
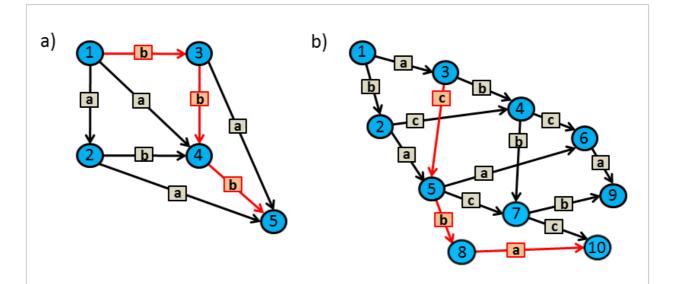

Image 1. Two examples of invitations sent by network members. Nodes represent network members, directed edges represent invitations (the source node is the invitation sender, the target node is the invitation addressee) and edge labels represent statuses offered in the invitations. a) There is a 3-element chained sequence of statuses matching the pattern **a*b***. A corresponding sequence of invitations is highlighted in red. Note that **abb** is the second optimal solution (see the path 1-2-4-5). b) There is a 3-element chained sequence of statuses matching the pattern **cb*a**. The only corresponding sequence of invitations is again highlighted in red.

The task

Given a set of invitations and a pattern, find the longest chained sequence of statuses which matches the pattern.

Input

The first input line contains a string over {'a',...,'z'} ∪ {'*'} which represents the pattern. The second input line contains integers M and I, separated by a space, where M is the number of members and I is the number of invitations. Next, I input lines follow. Each of these lines contains three values U_1, U_2 and S, separated by spaces. This triple represents an invitation with initial status S sent by member with ID U_1 to a person who became member with ID U_2. Member IDs are integers from 1 to M, statuses are characters from the set {'a',...,'z'}. The invitations are listed in a random order.

For each two members U_1 and U_2, there is at most one invitation sent either by U_1 to U_2 or vice versa. A person may receive several invitations, each of them sent by a different member (to become a member, he accepts one of the invitations). It is assumed there is at least one chained sequence of statuses which matches the input pattern.

It holds $M \leq 2 \times 10^5$ and $I \leq 4 \times 10^5$. The length of the pattern is at most 20.

Output

The output consists of one line containing integer L which is the maximal length of a chained sequence of statuses that matches the input pattern.

Example 1

Input

```
a*b*
5 8
1 2 a
1 3 b
1 4 a
2 4 b
2 5 a
3 4 b
3 5 a
4 5 b
```

Output

```
3
```

Example 1 is visualized in Image 1a).

Example 2

Input

```
cb*a
10 15
1 2 b
1 3 a
3 4 b
3 5 c
2 4 c
2 5 a
5 6 a
5 7 c
5 8 b
4 6 c
4 7 b
7 9 b
7 10 c
6 9 a
8 10 a
```

Output

```
3
```

Example 2 is visualized in Image 1b).

Public data

The public data set is intended for easier debugging and approximate program correctness checking. The public data set is stored also in the upload system and each time a student submits a solution it is run on the public dataset and the program output to stdout and stderr is available to him/her.

Link to public data set