Maintaining the Ancient Tree

Description

In the city where Kyaru lives, there's a huge ancient tree with a lot of branches and leaves. The city provides regular maintenance upon the tree which includes a task of cutting some of its leaves. Today, Kyaru is sent to do this task.

On this ancient tree, all of the branches and leaves are numbered with an integer. They grow on the trunk of the tree. There are sub-branches or leaves growing on a branch (or both of them), and no branch has nothing growing on it. There may also be some leaves directly growing on the trunk.

Kyaru has received a guidance showing which leaves are required to cut. As she is quite new to this job, the guidance is very detailed that it will show Kyaru how she could find all these leaves with complete paths. Meanwhile, the guidance also includes a whole list of the rest of leaves that shouldn't be cut in the same form above.

The tree is so huge that it takes Kyaru a lot of time to go through each path to cut the corresponding leaf. Luckily, she was informed that she can directly cut the branch that all leaves on it could be cut. With such two lists, Kyaru finds she could save time if she set a plan and choose some proper place to cut in advance.

Given the guidance, she would like you to help her find the minimum number of cutting she needs to perform.

Another thing you should noticed is that **Kyaru is not allowed to cut the trunk** even if she finds that all branches and leaves growing on it could be cut.

Input

The input contains several test cases. The first line contains a single positive integer T which is the number of test cases. For each test case, you are first given two non-negative numbers n and m . And then n non-empty lines of paths indicating leaves that should be cut, and m non-empty lines of paths for leaves that shouldn't be cut. All of these $n + m$ lines end at -1 , which doesn't belong to the path information.

Each path contains some non-negtive integers separated by spaces indicating the index numbers of branches or leaf along this path. For example, a line `2 1 3` in this list indicates that Kyaru should find and climb to the branch numbered `2` growing on the trunk, and then find and climb to the branch numbered `1` growing on the branch she's currently on, and then cut the leaf numbered 3. Please notice that there may be some leaves directly grow on the trunk. For example, a path of a single number `3` means Kyaru needs to cut a leaf numbered `3` growing on the trunk. There're some more constraints on these paths:

1. The index number of branches and leaves may be any integer within the range of $[0, 100]$, which means they are not necessarily starting from 1 or consecutive.
2. The path always indicates a leaf.
3. The paths are always unique (i.e. all the paths in a test case are different)

4. No branches and leaves growing on the same branch share the same index number. For example, there won't be two paths `1 2 1` and `1 2 1 2` in one test case. However, branches and leaves on different places **do not** subject to this constraint. For example, paths `1 2 1` and `1 2 2` are allowed to appear in the same test case.

Output

T lines of non-negative integers, the minimum number of cutting Kyaru needs to perform for each test case.

Sample Input

Input

```
2
3 0
2 4 -1
2 3 -1
3 -1
3 1
2 4 -1
2 3 -1
3 -1
2 2 -1
```

Output

```
2
3
```

Sample Explanation

In the first sample test case, Kyaru can perform only two cutting:

1. cut branch 2 growing on the trunk.
2. cut leaf 3 growing on the trunk.

In the second sample test case, Kyaru should perform three cutting:

1. cut leaf 3 growing on branch 3 growing on the trunk.
2. cut leaf 4 growing on branch 3 growing on the trunk.
3. cut leaf 3 growing on the trunk.

Constraint

$1 \leq n + m \leq 100$ holds and in the whole input there are no more than 10000 integers.