

Problem K. We apologize for any inconvenience

Input file: *standard input*
Output file: *standard output*
Time limit: 10 seconds
Memory limit: 512 mebibytes

Studying at the Jagiellonian University in Krakow has its pros and cons. Pros: Jagiellonian University. Cons: Krakow... Or, more precisely, a constant necessity to deal with tram track reconstructions.

Initially, the public transport network consists of some number of tram lines. Then one of them gets suspended, then another one, and then another... As you may know yourself, the inviolable rule in Krakow is to always suspend a line before any of the previously suspended lines resumes its operation. Right now not all of the lines have been suspended (yet), and you sit in a tram, annoyed that your direct connection to the university has just disappeared. You look out of the window and you ask yourself: "Am I actually the most unlucky passenger in this city? Or is there someone out there somewhere who needs to change lines even more times in order to get where they want?"

More precisely, we say that stop B is reachable from stop A with c changes if there exist lines l_0, \dots, l_c such that l_0 serves stop A , l_c serves stop B , and for each $0 \leq i < c$ there exists some stop served by l_i and l_{i+1} . At each point in time, you want to know the largest value of c such that there exists a pair of stops (A, B) where B is reachable from A with c changes and B is not reachable from A with c' changes for any $c' < c$.

Note that sometimes it might not be possible to travel between a pair of stops at all. As follows from the definition above, you decide not to take such pairs into consideration in your analysis – you conclude that if someone wishes to travel between those stops, they will take an Uber anyway.

Input

The first line of input contains the number of test cases z ($1 \leq z \leq 35$). The descriptions of the test cases follow.

The first line of each test case contains the number of stops n and the number of tram lines k ($2 \leq n, k \leq 750$). The stops are numbered from 1 to n and the lines are numbered from 1 to k .

Then, k lines follow. The i -th of those lines describes the route of the tram line number i . Each line starts with an integer r_i ($2 \leq r_i \leq n$) followed by r_i distinct integers $a_{i,j}$ ($1 \leq a_{i,j} \leq n$) – the identifiers of stops served by the i -th tram line. Any tram line runs in both directions.

The next line contains a single integer s ($1 \leq s \leq k - 1$).

Then, s lines follow, describing the order in which the tram lines get suspended. Each of those lines contains a single integer s_i ($1 \leq s_i \leq k$) – the identifier of the suspended tram line. Any line can get suspended at most once.

The sums of values of n and k in all test cases do not exceed 1000 each.

Output

For each test case, output $s + 1$ lines, each containing a single integer. The $i + 1$ -th line should denote the largest number of line changes necessary after the i -th suspension event (the first line denoting the answer before any suspensions).

Example

standard input	standard output
1	1
5 4	2
3 1 3 5	0
2 1 4	0
2 2 3	
2 2 4	
3	
1	
4	
3	

Note

Initially, one line change is required to travel, for example, from stop 4 to stop 5 (or vice versa). Such travel is

possible by taking line 2, then line 1. There are no pairs of stops requiring 2 or more changes.

After line 1 gets removed, two line changes are required to travel from stop 1 to stop 3 (or vice versa).

When lines 1 and 4 get removed, the only pairs of stops still reachable one from another are (1, 4) and (2, 3), and in both cases no line changes are required to travel between them.

When lines 1, 4 and 3 get removed, the only pair of stops still reachable one from another is (1, 4). No line changes are required to travel between these stops.