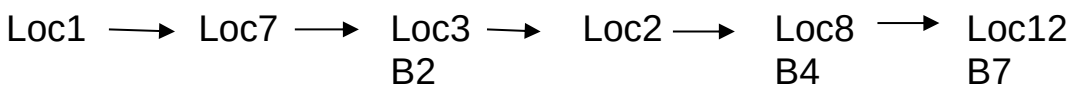
Scavenger Hunt

Filenames: *scavenger*, *scavengersmall* (for partial credit)
Time limit: *2 seconds*

You are participating in a virtual scavenger hunt, with *n* possible locations to visit. Whenever you visit a location, you are given a list of 0 or more choices of other locations to visit. If there are no choices, then your scavenger hunt ends, otherwise, you choose one of the given locations and proceed. Everyone starts at location number 1, and the choices are designed so that you can not visit any location twice, thus, it's guaranteed that your scavenger hunt will end.

In the old system, your score was simply the total number of locations visited. Unfortunately, word leaked that if someone had a list of all the choices from each location in advance, then they could pay some UCF Computer Science students a bit of money to figure out which path would earn the maximum score.

The organizers have added a twist this year. A few of the locations will be designated as bonus locations. If you visit a bonus location, then your score for the portion of the path starting at the bonus location until the end of the scavenger hunt is multiplied by the bonus multiplier assigned to that location, an integer in between 2 and 10, inclusive. Furthermore, bonuses can build upon each other! For example, consider the following path for a scavenger hunt:



Locations 3, 8 and 12, have bonus values of 2, 4 and 7, respectively. The last vertex contributes a value of 2*4*7 points because it benefits from the multipliers on locations 3, 8, and 12, while the first vertex only contributes 1 point because it isn't affected by any multipliers on the path. Therefore, the total value of the path above is {1 + 1 + 2+ 2 + 8 + 56} = 70.

You've been approached by some contestants who have stolen a map of the scavenger hunt. Help them determine the maximum possible score they can achieve in the scavenger hunt.

The Problem

Given the layout of the scavenger hunt, including all of the forwarding locations, which locations are bonus locations and what each bonus location's bonus multiplier is, determine the maximum possible score achievable for the scavenger hunt.

The Input

The first line of input will contain a single positive integer, c ($c \leq 100$), representing the number of input cases to process. The input cases follow. The first line of each input case has two space separated integers, n ($2 \leq n \leq 200$), representing the number of locations for the scavenger hunt, and b ($0 \leq b \leq n$), representing the number of bonus locations. The following b lines contain information about the bonus locations. On each of these lines there will be two space separated integers, v ($1 \leq v \leq n$), representing the location number and m ($2 \leq m \leq 10$), representing the multiplier bonus for that location. The locations that receive bonuses will all be unique.

The following n lines contain information about each of the locations, in order, from location 1 to location n. On the i^{th} of these lines, the first integer on the line, f_i ($0 \leq f_i < n$), represents the number of forwarding locations from location i. The following f_i values will be distinct integers in between 1 and n, inclusive, representing the locations to travel to on the scavenger hunt from location i. The input cases will be constructed such that no case has an answer greater than 10^9.

Partial Credit Input (50%)

The number of locations on the scavenger hunt, n, will be less than or equal to 10.

The Output

For each input case, output a single integer on a line by itself representing the maximum score that can be achieved on the scavenger hunt.

Sample Input

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

Sample Output

```
17
2
```