Problem C: Ideas

A unique feature of ideas is that they are not consumed when used. A good idea can benefit arbitrarily many people without diminishing the value of the idea. An idea can even serve as the basis for creative people to derive even better ideas. Each person relies on a specific set of ideas to build on and create new ideas.

In order to realize these benefits, ideas need to be communicated to the people who use them. People have developed an extensive worldwide communication network to satisfy this need. The network is composed of a series of tubes connecting people. The tubes are inhabited by curious creatures called packets which carry ideas from one person to another. In order to avoid collisions between packets, the tubes are all unidirectional. Each person can have zero or more incoming and zero or more outgoing tubes.

All of the packets start at the same person, and each packet follows the following algorithm:

1. Learn and remember the ideas created by the current person.
2. If there are no outgoing tubes from the current person, stop executing the algorithm.
3. Otherwise, choose an arbitrary outgoing tube and use it to travel to a new person.
4. Tell the new person all of the ideas that he or she needs from other people.
5. Go back to step 1.

The input data is such that it is possible for a packet to reach every person from person 0, and whenever a person P relies on a given idea, every path a packet could have taken to reach P will have visited at least one person who created that idea. It is possible for more than one person to independently create the same idea.

To ease the strain on each packet, you would like to minimize the number of ideas that it remembers at any given time by directing the packet to forget certain ideas in certain tubes. However, in so doing, you must ensure that no matter what path the packet takes, every time it visits a person, the packet knows all of the ideas that the person needs.

Input Specification

The first line of input contains a single integer, the number of test cases to follow. Each test case begins with a line containing three integers N, M, I, the number of people, tubes, and ideas, respectively. Each of these integers is between 1 and one thousand, inclusive. People are numbered from 0 to N-1, ideas from 0 to I-1. All of the packets start at person 0. 2N lines follow, two lines for each person in order from 0 to N-1. Each of these lines contains some number of integers separated by spaces. For each person, the first line lists the ideas that the person needs, and the second line lists the ideas that the person creates. A given idea will never appear on both lines for the same person. M more lines follow, each describing a tube using two integers: the source and destination person of the tube.

Sample Input

```
1
3 3 2

1
1
0
0
1
0 1
1 2
2 1
```

Output Specification

For each test case, output M lines, each corresponding to one of the tubes in the same order as in the input. For each tube, output a list of integers: the minimal set of ideas the packet must have in its memory when travelling through the given tube. Output the ideas in each set in increasing order.

Output for Sample Input

```
1
0
1
```

Ondřej Lhoták