

2057 Buggy Sat

Discovery Co. Ltd. builds a satellite using a new kind of an intelligent camera. The camera has special software to detect cities and roads from an image, and is also able to detect every region (which is a connected part of surface), bounded by a series of connected roads with no other region inside. Using this technology, the satellite is able to compress the picture before sending it. The compressed format of a picture is the city locations and its regions.

Discovery has launched the satellite, without testing the software completely. So, after a while, they received some buggy pictures which includes one more extra region: the outer region. The outer region is the region of the plane enclosing every other region (which has infinite area). Further analysis shows that all images sent have the following properties:

1. All cities, in the image, have at least two roads to the other cities.
2. There is a path connecting every pair of cities.
3. There is at most one road between each pair of cities.
4. Roads do not cross each other except at the cities.

You are to write a program to read a buggy image and report the outer region.

Input

The first line of the input consists of a single integer N ($1 \leq N \leq 20$), which is the number of test cases. The test cases appear with no blank lines in between. The first line of each test case consists of the number of cities (between 1 and 50) followed by pairs of integers (x, y) which are location of cities (each pair in one line), followed by number of faces in a separate line (between 1 and 50), followed by face information on each line. Face information consists of number of cities making the face and the city numbers in clockwise (or counterclockwise) order.

Output

There should be a single line containing the boundary face number for each test case, with no blank lines in between.

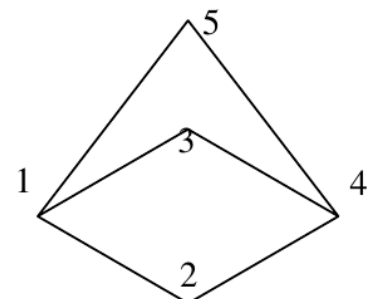
Note: The Figure on the right shows a image of the Sample Input below.

Sample Input

```

1
5
2 6
4 4
4 7
8 6
4 10
3
4 1 2 4 3

```



4 1 3 4 5
4 1 2 4 5

Sample Output

3