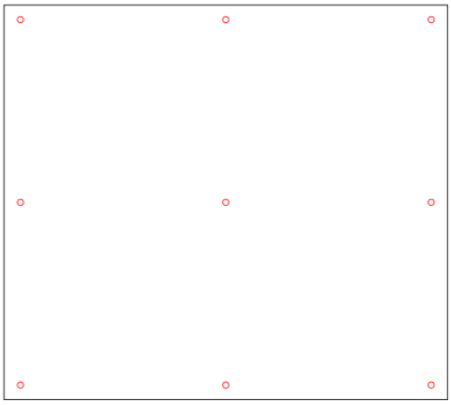
Problem A - Setting up an accelerator lab in the living room

Description

Your roommates are setting up an accelerator lab in the living room with several linear particle colliders (of small size!) in order to do the homework for the Accelerator Physics course. Each linear collider is a straight hollow pipe with two ends: one that fits a given particle source device that generates and injects particles into the collider and the other that fits a given target device where the particles should collide. A collider requires a given source and a given target device. Interestingly enough, each device can be shared with several colliders and work in different manners. For instance, a device can be a source for a collider and be a target for another collider. Well, this is what was written on that Darknet website from where the material was ordered..

Since the living room has only one sofa and a TV (for your roommates to watch "The Big Bang Theory" series) they can put the source and target devices almost anywhere. The only constraint is that colliders should not cross, otherwise, some unexpected interference may occur, for instance, an antimatter black hole in the middle of your living room! Unfortunately, this may not be possible and they are willing to run the risk if you help them to decide where to put the source and target devices in the living room such that the number of crossings between colliders is as small as possible.

Let's see an example. The following figure shows the floor of the living room as well as the nine places where the source and the target devices can be set (in red circles).

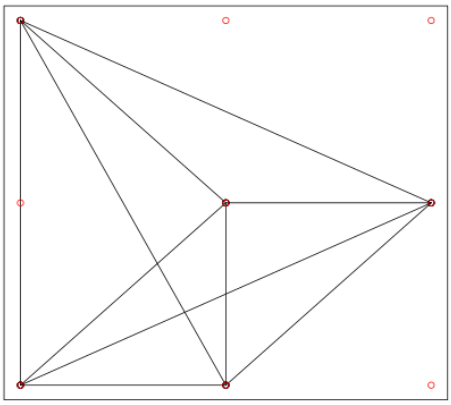


Consider that you have 5 source/target devices and 10 colliders with the following requirements:

- Device 1 is a source for 4 colliders, each of which requires devices 2, 3, 4 and 5, respectively, as targets.
- Device 2 is also a source for 3 colliders, each of which requires devices 3, 4 and 5, respectively, as targets.
- Device 3 is also a source for 2 colliders, each of which requires devices 4 and 5, respectively, as targets.
- Device 4 is also a source for one collider, which requires device 5 as target.

The following figure shows a positioning of the ten colliders in the living room that

minimizes the number of crossings (only three).



Note that a crossing between two colliders occur if both are partially overlapped. However, it does not occur if both only share the same end.

Input

Each test case starts with the number of places in the living room, $1 \leq p \leq 81$. Each of the following p lines contains two integers, the (x,y)-coordinate of a place in the living room. Then, a line with two positive integers follows: the number of devices, $1 \leq n \leq 81$, and the number of colliders, $1 \leq b \leq 28$. Each device is numbered from 1 to n. Each of the following b lines contains the indices of the two devices, working as source and as target, respectively, that must be attached to both ends of a collider.

Output

For each test case, print the minimum number of crossings of colliders that can be obtained. Note that all b colliders must be set for computing the total number of crossings.

Further considerations

Report

The maximum grade in mooshak is 150 points. In case you are able to reach 100 points, you are allowed to submit a 3-page report (in english or in portuguese) to describe your implementation. Use the template that is provided in inforestudante. The maximum grade in the report is 50 points. The grade depends of how well you are able to answer the following questions in the report:

- What is the time complexity of your approach? If your approach is recursive, which is the time complexity of each recursive step?
- What data structures were implemented? What is their purpose?
- Is your approach correct for any input? Prove it, even if informally.
- How does your implementation detect that two colliders are crossing?
- (if applicable) Which speed-up tricks did you implement in order to reach 150 points?
- (if applicable) What could be the reason for not being able to reach 150 points?

Validation

In order to allow us to validate your solutions, implement a function that prints the *(x,y)*-coordinates of both ends of each collider, each per line, to a file named data.out. Give clear instructions on how to use this function. Note that if we are not able to validate your solution, your grade to this problem may be cancelled.

Example 1

Example input:

```
7
5 9
7 8
7 5
6 2
3 3
2 5
3 8
5 4
1 2
1 3
3 5
2 4
```

Example output:

```
0
```

Example 2

Example input:

```
9
3 2
4 2
5 2
3 3
4 3
5 3
3 4
4 4
5 4
5 10
1 2
1 3
1 4
1 5
2 3
2 4
2 5
3 4
3 5
4 5
```

Example output:

```
3
```

A recommendation: use git within the team for collaborative work, but set it up as a private repository.