

## Problem A. Valar Morghulis

Program:            winter.(cpl|java)  
Input:              winter.in  
Balloon Color:     White

Khaleesi has set sails for Kings Landing. With three dragons and tens of thousands in her army, she will be unstoppable. The ships are now getting closer to Kings Landing and she want to start planning for her attack.

The plan for the attack is that the army will be split into two wings: right and left. There is one problem though, Khaleesi's soldiers are from very diverse backgrounds and some of them have rivalries and don't get along (like Martells and Tyrells) so they can't be in the same wing.

Khaleesi knows all the rivalries and can try to resolve some of them. Some rivalries are hard to resolve though and she will have to deal with them. She wants to check if some rivalries are not resolved, will it be possible split the soldiers into two wings where no two soldiers who don't get along are in the same wing. Can you help Khaleesi plan for her attack to sit on the iron throne?

### Input

Your program will be tested on one or more test cases. The first line of the input will be a single integer  $T$ , the number of test cases ( $1 \leq T \leq 10$ )

Each test case starts with a line that contains three space separated integers:

- **N**: The number of soldiers ( $1 \leq N \leq 40,000$ )
- **E**: The number of rivalries ( $1 \leq E \leq 40,000$ )
- **Q**: The number of queries ( $1 \leq Q \leq 40,000$ )

Followed by **E** lines each containing 2 space separated integers **A** and **B** which means that soldier **A** and **B** can not be in the same wing. ( $1 \leq A, B \leq N$ )

Followed by **Q** lines each containing 2 space separated integers **C** and **D** which means that all rivalries between **[C,D]** can not be resolved. ( $1 \leq C, D \leq E$ )

### Output

For each test case print **Q** lines where the  $i^{th}$  line represents the answer for the  $i^{th}$  query. The line should contain "-1 -1"(quotes for clarity) if the army can't be split into two wings or the size of the wings separated by single space otherwise. If there are multiple valid assignments print the one that maximizes the first wing.

## Example

winter.in	standard output
1	-1 -1
9 8 6	-1 -1
5 4	6 3
3 4	7 2
5 6	7 2
5 3	8 1
7 8	
4 8	
5 4	
3 4	
1 8	
1 4	
2 5	
2 4	
4 7	
4 4	