Problem B - Buggy Map

Daniel has decided to change things up for ACM practice, instead of getting pizza delivered from the a single pizzeria he's ordered from before, he decides to order from two pizzerias that he's never ordered from before. Unfortunately, Daniel has already ordered from all pizzerias that deliver, so he has to go to those pizzerias himself to pick up his order.

On the upside, Vancouver is very walkable, and there are many pizzerias nearby. In fact, if we view the map of Vancouver as a graph with intersections as vertices and roads as undirected edges, there are pizzerias at every intersection!

This is the perfect oppurtunity for Daniel to test out a map application on his phone that he wrote. However, his app has a tiny bug he hasn't had time to fix. The app has no problem getting from UBC to a pizzeria and back, but doesn't give the right directions if travelling from pizzeria a to pizzeria b if there are two different paths to go from a to b. A path is a way to get from a to b that has no repeated vertices. Two paths between a and b are different if at least one edge along the respective paths is different.

Daniel doesn't want to head back to UBC between the orders and wants to accomplish it in one trip. Help Daniel figure out how many different pairs of pizzerias he can order from while using his buggy app.

Input

The first line contains a single integer, T specifying the number of test cases.

Each test case begins with two space seperated integers, $1 \leq n \leq 50000$, denoting the number of vertices, and $1 \leq m \leq 150000$, denoting the number of edges in the city. On each of the next m lines will contain a pair of integers $1 \leq x, y \leq n$, denoting that intersection x is connected to intersection y. It is guaranteed that $x \neq y$, and there is at most one edge between any pair x and y.

Output

For each test case, output one line with the integer representing the number of pairs of pizzerias that have a unique path to travel from one to the other. Paths are ways of traversing edges to get from one point to another with no repeated vertices. Two paths from a to b are different if at least one edge along the respective paths is different.

Sample Input

```
2
5 5
1 2
2 3
3 1
2 4
1 5
4 3
1 2
2 3
2 4
```

Sample Output

```
2
6
```