

Problem F. Chaleur

Input file: **standard input**
 Output file: **standard output**
 Time limit: 1 second
 Memory limit: 256 megabytes

DreamGrid has n friends which are conveniently numbered from 1 to n . They can be divided into two groups (possibly empty) such that:

- Every pair of friends in the first group have to know each other.
- Every pair of friends in the second group must not know each other.

Now, given the pairs of friends who know each other, DreamGrid would like to know the number of ways to find a group of friends with maximum size such that every pair of friends in the group have to know each other, and he would also like to know the number of ways to find a group of friends with maximum size such that every pair of friends in the group must not know each other.

Input

There are multiple test cases. The first line of input contains an integer T , indicating the number of test cases. For each test case:

The first line contains two integers n and m ($1 \leq n \leq 10^5, 0 \leq m \leq 10^5$) – the number of friends and the number of pairs of friends who know each other.

The i -th of the following m lines contains two integers a_i and b_i ($1 \leq a_i, b_i \leq n, a_i \neq b_i$), which denotes that the a_i -th friend and the b_i -th friend know each other. Note that every unordered pair of (a, b) will appear at most once.

It is guaranteed that neither the sum of all n nor the sum of all m exceeds 2×10^6 .

Output

For each test case, output two integers separated by a single space.

The first integer indicates the number of ways to find a group of friends with maximum size such that every pair of friends in the this group have to know each other.

The second integer indicates the number of ways to find a group of friends with maximum size such that every pair of friends in the group must not know each other.

Example

standard input	standard output
3	2 1
3 2	1 4
1 2	1 2
2 3	
6 6	
1 2	
2 3	
1 3	
1 4	
2 5	
3 6	
4 1	
1 2	