

DescriptionHintsSubmissionsDiscussionsNotes

One Edge

2 sec

512000KB

100

Difficulty

Time Limit

Memory

Score

80/80 XP

30/30

Description

You have given an undirected graph with n nodes, and m edges between them. The goal is to add exactly one edge between two nodes such that the total number of connected components in the graph decreases.

Your task is to find out the number of ways to add such edge.

Input Format

The first input line has two integers n and m : the number of nodes and edges. The nodes are numbered 1, 2, ..., n .

After that, there are m lines describing the edges. Each line has two integers a and b : there is an edge between those nodes.

An edge always connects two different nodes, and there is at most one edge between any two nodes.

Output Format

Print the number of ways to add such edge, described in the statement.

Constraints

$1 \leq n \leq 10^5$

$1 \leq m \leq 2 \times 10^5$

$1 \leq a, b \leq n$

Sample Input 1

5 4
1 2
2 3
1 3
4 5

Copy

C++1400:00:0012 px

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 #define endl "\n"
4 using lli=long long int;
5 using pp=pair<lli,lli>;
6 lli m = 1e9 + 7;
7 vector<vector<int>>g;
8 vector<int>color;
9 vector<int>vis;
10 vector<int>colNum;
11 void dfs(int node,int col){
12     vis[node]=1;
13     color[node]=col;
14     colNum[col]++;
15     for(auto i:g[node]){
16         if(!vis[i]){
17             dfs(i,col);
18         }
19     }
20 }
21 int main(){
22     ios_base::sync_with_stdio(0);
23     cin.tie(0);
24     cout.tie(0);
25     int n,m;
26     cin>>n>>m;
27     g.resize(n+1);
28     color.resize(n+1);
29     vis.resize(n+1);
```

Sample TestsManual Tests

Test Case 1

Test Case 2

Input

5 4
1 2
2 3

Console

Run on Sample