Czech Technical University in Prague
ICPC Foundation
Czech ACM Chapter

Central Europe Regional Contest 2020

Screamers

`screamers.c`, `screamers.cpp`, `Screamers.java`, `screamers.py`

The police is preparing a huge sting in which they hope to jail most of the prominent criminal figures in the city. The flow of information on the side of the police has to be as tight as possible to prevent any leaks. Each detective officer (DO) taking part in the sting is under a strict regulation.

The information shared among DOs is in the form of so-called drops. A drop is always spoken, it must not be recorded on any medium, electronics, paper, etc. Any DO can share a drop only with selected DOs with which he shares a bidirectional connection. Each DO is obliged to pass the drop, as soon as possible and without any change, to all his buddies with which he shares a connection, except for the DO from which he received the drop.

The Chieft Inspector (CI) has to choose which pairs of DOs will share a connection. This final set of connections is called the final group. In this final group (FG) an additional FG-rule holds: A situation when a drop returns to a DO who passed it to his buddies some time ago must not happen. It would mean there are too many unnecessary connections in the FG network.

There is a stack of folders, each folder describing one connection between a particular pair of DOs. The selection of FG is done in two steps. First, CI chooses two integer values S and T, which may be sometimes the same, and removes from the stack all folders above the S-th folder and all folders below the T-th folder.

Next, with the reduced folders, CI repeats the operation. He chooses two integer values U and V, which may be sometimes the same, and removes from the reduced stack all folders above the U-th folder and all folders below the V-th folder.

CI wants to use all the connections in the remaining folders in the FG. However, it is not guaranteed that connections can form FG, due to the additional FG-rule. CI tends to forget this rule quite often.

One cannot change the professional habits of CI. His assistant tries to address the issue diplomatically by employing a programmer who would gradually computerize the process. His first task is to compute the number of different FGs that may be selected by CI after he has chosen the first two values S and T. This computation must be efficient for many different values of S and T.

Input Specification

The first input line contains two numbers, N and M ($1 \leq N, M \leq 10^5$), the number of DOs and the number of folders in the CI's stack respectively. The DOs are identified by integers $1 \ldots N$. Next, there are M lines, each represents one folder and it contains two integers A and B ($1 \leq A < B \leq N$), pair of DOs whose connection is described in the folder. The order of lines corresponds to the order of folders in the stack from top to bottom.

The next line contains one number Q ($1 \leq Q \leq 10^5$), the number of queries. Next, there are Q lines, each represents one query and it contains two positive integers S and T ($1 \leq S \leq T \leq M$), the numbers chosen by CI in the first step of FG selection.

Output Specification

For each of the Q query input lines print the number of different FGs which can be formed in the second step of the selection process.

Sample Input 1

```
4 6
1 2
2 3
1 3
1 4
3 4
2 4
4
1 1
1 3
2 4
1 6
```

Output for Sample Input 1

```
1
5
6
13
```