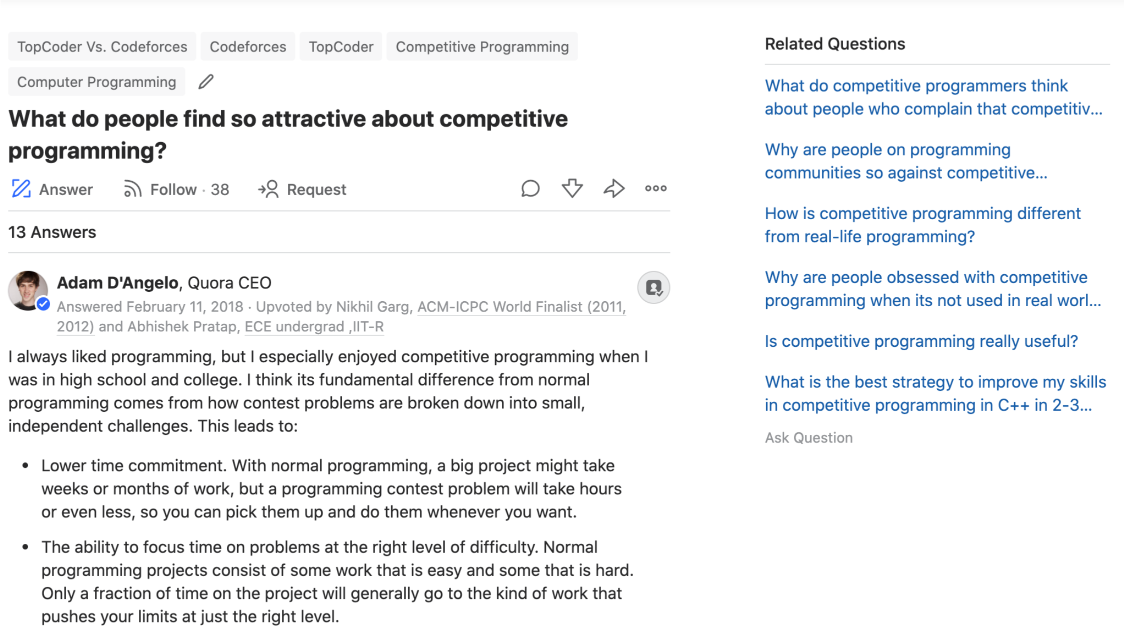
# Help Crawlers
## Problem ID: crawlers

The heart of Quora is questions. When a user lands on a question page, we provide related questions so that they can browse more questions. In order to distribute our content to a broader audience, we can leverage crawlers. When a crawler lands on a question page, it will visit all related questions on that page which it hasn't yet visited.

Given the current mapping from each question to its related questions, determine a minimum set of additional related questions which we could add such that a crawler could reach all other questions starting from any question on Quora.



**What do people find so attractive about competitive programming?**

13 Answers

**Adam D'Angelo**, Quora CEO
Answered February 11, 2018 · Upvoted by Nikhil Garg, ACM-ICPC World Finalist (2011, 2012) and Abhishek Pratap, ECE undergrad ,IIT-R

I always liked programming, but I especially enjoyed competitive programming when I was in high school and college. I think its fundamental difference from normal programming comes from how contest problems are broken down into small, independent challenges. This leads to:

- Lower time commitment. With normal programming, a big project might take weeks or months of work, but a programming contest problem will take hours or even less, so you can pick them up and do them whenever you want.
- The ability to focus time on problems at the right level of difficulty. Normal programming projects consist of some work that is easy and some that is hard. Only a fraction of time on the project will generally go to the kind of work that pushes your limits at just the right level.

**Related Questions**

What do competitive programmers think about people who complain that competitiv...

Why are people on programming communities so against competitive...

How is competitive programming different from real-life programming?

Why are people obsessed with competitive programming when its not used in real worl...

Is competitive programming really useful?

What is the best strategy to improve my skills in competitive programming in C++ in 2-3...

## Input

Your program will receive input from standard input.

The first line of the input contains two integers $n$ and $m$, where $n$ is the number of questions and $m$ is the number of related question pairs. After that, there are $m$ lines. The $i$-th line contains two positive integers $q_{i,1}$ and $q_{i,2}$ indicating that $q_{i,2}$ is a related question of $q_{i,1}$.

## Output

Your program should write to standard output.

The first line should print the minimum number of related question pairs you need to add. After that, print the question pairs in the following lines. Each line should have two questions indicating that the second question is a related question for the first question.

## Constraints

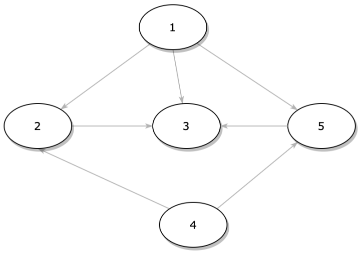
- $1 \leq n \leq 10^6$
- $0 \leq m \leq 2 \cdot 10^6$
- $1 \leq q_{i,1}, q_{i,2} \leq n; q_{i,1} \neq q_{i,2}$
- There is no self edge
- There may be multiple edges

## Subtasks

You will get points for each subtask when you pass all of the testcases of the subtask.

1. $n \leq 10; m \leq 20$ (19 points)
2. $n \leq 10^3; m \leq 2 \cdot 10^3$ (32 points)
3. No additional constraints (49 points)

## Sample Explanation



This image illustrates the sample Input/Output.

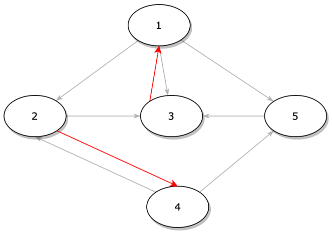
If a crawler lands on question 1, it can reach questions 2, 3, 5.

If a crawler lands on question 2, it can reach question 3.

If a crawler lands on question 3, it cannot reach any other question.

If a crawler lands on question 4, it can reach questions 2, 3, 5.

If a crawler lands on question 5, it can reach question 3.



With two more related question pairs, we can make crawlers reach all questions.

| Sample Input 1 | Sample Output 1 |
|---|---|
| 5 7<br>1 2<br>2 3<br>1 3<br>1 5<br>4 5<br>4 2<br>5 3 | 2<br>3 1<br>2 4 |