

Hello, **2024101067**.

# The Great TA Hunt

[Submit solution](#)[All submissions](#)[Best submissions](#)✓ **Points:** 100 (partial)⌚ **Time limit:** 1.0s📄 **Memory limit:** 256M✍ **Author:**

Om

➤ **Problem type**▼ **Allowed languages**

C, C++

In IIITH ,  $N$  students are enrolled in the DSA course. It's final assignment season, and the submissions are in.

You've been eyeing the position of becoming a TA for this course in the future, and your first challenge comes early. Rumors are that students have copied assignments! Some even from multiple classmates! After digging through MOSS results, you've uncovered  $M$  suspicious copying links. Each link tells you that student  $u$  has copied from student  $v$ .

Importantly, no student copied from someone who eventually copied from them — meaning the links of copying is a Directed Acyclic Graph (DAG).

## Your Mission as a Future TA

To maintain academic integrity , you decide to rank all the students based on how original their assignments were. The rules are:

- If student  $u$  copied from student  $v$ , then  $u$  must be ranked lower (i.e., get a higher number) than  $v$ . Students who copy shouldn't be topping the list.
- Some students might have copied from multiple others, and all of those relationships must be respected.
- Among all possible valid rankings, you want to find the lexicographically smallest one.

## Constraints

- $1 \leq N \leq 10^5$  (Number of students)





Hello, 2024101067.

## Input Format

- The first line contains two integers  $N$  and  $M$  — the number of students and the number of copying links.
- The next  $M$  lines each contain two integers  $u$  and  $v$ , indicating student  $u$  copied from student  $v$ .

## Output Format

- Print  $N$  integers — the rank of each student (from ID 1 to  $N$ ) such that:
  - All copying relationships are respected.
  - The rank list is lexicographically smallest among all valid possibilities.

## Examples

Sample 1:

```
10 8
2 7
5 4
9 1
4 8
5 6
7 1
10 1
9 7
```

Copy

Output 1:

```
5 1 6 8 7 9 3 10 2 4
```

Copy

Sample 2:

```
9 5
2 6
7 1
9 8
7 6
5 1
```

Copy

Hello, **2024101067**.

```
3 4 5 6 1 7 2 9 8
```

Copy

Sample 3:

```
5 8
1 3
5 2
2 1
2 3
4 2
5 1
4 1
5 3
```

Copy

Output 3:

```
4 3 5 1 2
```

Copy

In the first test case, we have 10 students and 8 copying relationships:

- Student 2 copied from Student 7
- Student 5 copied from Student 4
- Student 9 copied from Student 1
- Student 4 copied from Student 8
- Student 5 copied from Student 6
- Student 7 copied from Student 1
- Student 10 copied from Student 1
- Student 9 copied from Student 7

To create a valid ranking, we need to ensure that any student who copied must be ranked lower than the student they copied from. Looking at our constraints:

- Student 1 must be ranked higher than students 9, 7, and 10
- Student 7 must be ranked higher than students 2 and 9
- Student 4 must be ranked higher than student 5
- Student 8 must be ranked higher than student 4
- Student 6 must be ranked higher than student 5

The output ranks students in the order: 5, 1, 6, 8, 7, 9, 3, 10, 2, 4 This means:

Student 1 gets rank 5 Student 2 gets rank 1 Student 3 gets rank 6 And so on...

This ordering satisfies all copying relationships while being lexicographically smallest.



Hello, **2024101067**.

[Request Clarification](#)

No clarifications have been made at this time.