DAG Queries

by zemen

Problem | Submissions | Leaderboard | Discussions | Editorial

You are given a Directed Acyclic Graph (DAG) with n vertices and m edges. Each vertex v has an integer, a_v, associated with it and the initial value of a_v is 0 for all vertices. You must perform q queries on the DAG, where each query is one of the following types:

1. `1 u x`: Set a_v to x for all v such that there is a path in the DAG from u to v.
2. `2 u x`: Set a_v to x for all v such that there is a path from u to v and $a_v > x$.
3. `3 u`: Print the value of a_u on a new line.

Input Format

The first line contains three space-separated integers describing the respective values of n (the number of vertices in the DAG), m (the number of edges in the DAG), and q (the number of queries to perform).
Each of the m subsequent lines contains two space-separated integers describing the respective values of u and v (where $1 \leq u, v \leq n, u \neq v$) denoting a directed edge from vertex u to vertex v in the graph.
Each of the q subsequent lines contains a query in one of the three formats described above.

Constraints

- $2 \leq n \leq 10^5$
- $1 \leq m, q \leq 10^5$
- $0 \leq x \leq 10^9$
- $0 \leq a_v \leq 10^9$
- It's guaranteed that the graph is acyclic, but there may be more than one edge connecting two nodes.

Output Format

For each query of type 3 (i.e., `3 u`), print the value of a_u on a new line.

Sample Input 0

```
6 5 18
1 2
1 3
3 4
2 4
5 6
1 1 3
3 1
3 2
3 3
3 4
1 2 2
3 1
3 2
3 3
3 4
2 6 7
```

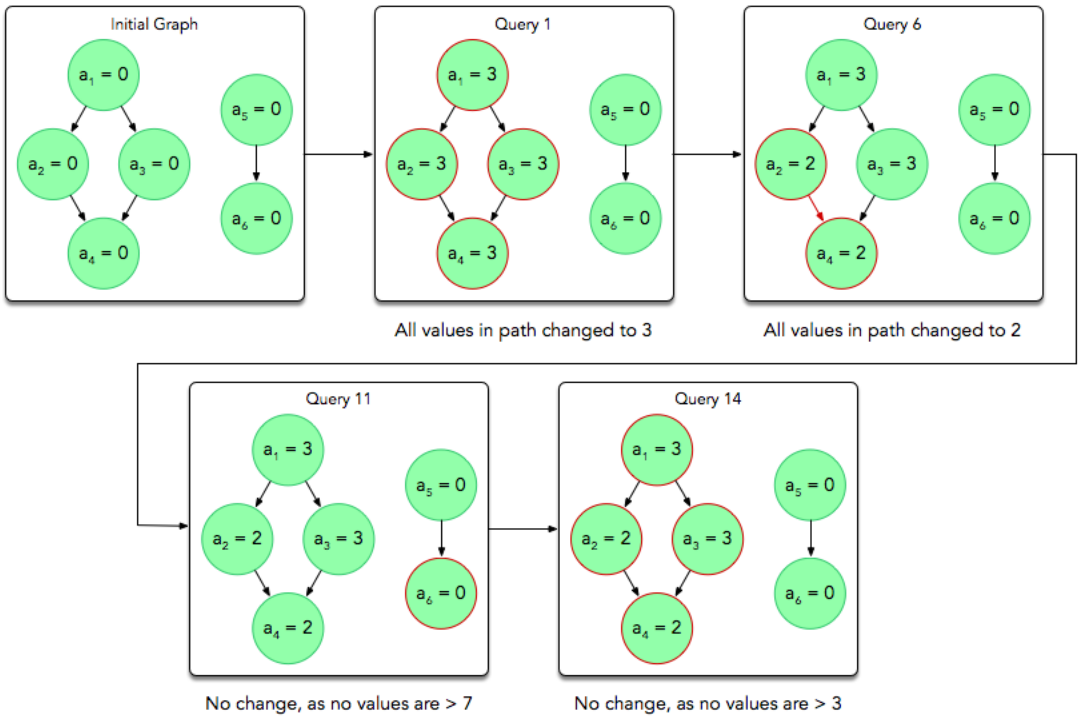
```
3 5
3 6
2 1 3
3 1
3 2
3 3
3 4
```

Sample Output 0

```
3
3
3
3
3
2
3
2
0
0
3
2
3
2
```

Explanation 0

The diagram below depicts the changes to the graph after all type 1 and type 2 queries:



Submissions:86

Max Score:80

Difficulty: Expert

Rate This Challenge:

More

Current Buffer (saved locally, editable) Java 7

```java
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
    }
}
```

Line: 1 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code