Problem I. Directed Acyclic Graph

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

Recently, Rikka showed great interest in the data structures for directed acyclic graphs (DAGs). She dreams that extending classic tree-based algorithms like "weighted-chain decomposition" to their counterparts based on DAGs will be perfectly coooooool!

Now, she came up with a simple problem, and she would like to invite you to solve this problem with her.

You are given an n-node m-edge DAG G. Each node u has a non-negative integer value val_u. All values are set to 0 initially.

Rikka wants to perform q operations of three types described below:

1. Given u and x, set val_v to x for all v reachable from u;
2. Given u and x, set val_v to $\min\{val_v, x\}$ for all v reachable from u;
3. Given u, print its current value val_u.

Can you perform all these operations fast enough?

A node v is said to be *reachable* from u if there is a path starting in u and ending in v. A *path* is a node sequence $p_1, p_2, \ldots, p_k$ satisfying $(p_i, p_{i+1}) \in G$ for each $i = 1, 2, \ldots, k-1$.

Input

The first line of input contains three integers n, m, q ($1 \le n, m, q \le 10^5$).

Then m lines follow. Each of them contains two integers x and y, representing a directed edge (x, y) in the graph ($1 \le x, y \le n$). The input graph is guaranteed to be a DAG.

Then q lines follow. Each of them contains two or three integers in one of the following three formats:

- "`1` u x" indicating the first type of operation;
- "`2` u x" indicating the second type of operation;
- "`3` u" indicating the third type of operation.

All parameters in the operations above satisfy $1 \le u \le n$ and $0 \le x \le 10^9$.

Output

For each operation of the third type, print a single line containing an integer: the current value of val_u.

Example

standard input	standard output
4 4 7 1 2 1 3 3 4 2 4 1 1 5 1 2 1 3 3 3 4 2 1 3 3 2 3 3	5 1 1 3