

# Mock CCC '18 Contest 2 S5 - A Link/Cut Tree Problem

---

Given a graph, support the following two operations:

`Query(ai, bi, wi)`: Does there exist a path from `ai` to `bi` using only edges with weight at least `wi`?

`Update(mi, xi)`: Update the weight of edge `mi` to be `xi`.

## Constraints

---

For 2 marks, there will be no update operations.

For 3 additional marks,  $M \leq 10^3$  and  $Q \leq 10^3$ .

## Input Specification

---

The first line will contain two space-separated integers,  $N(1 \leq N \leq 10^3)$  and  $M(1 \leq M \leq 5000)$ , indicating respectively the number of vertices and the number of edges in the graph.

The next  $M$  lines will contain three space-separated integers  $u_i(1 \leq u_i \leq N)$ ,  $v_i(1 \leq v_i \leq N, u_i \neq v_i)$  and  $z_i(1 \leq z_i \leq 10^9)$ , indicating that edge  $i$  is an undirected weighted edge between vertices  $u_i$  and  $v_i$  with weight  $z_i$ . There may be multiple edges between two vertices.

The next line contains a single integer  $Q(1 \leq Q \leq 10^5)$ , the number of operations to support.

Each of the next  $Q$  lines will contain the description of either a query or an update.

An update operation, which can happen at most 2000 times, will take the form `1 mi xi` ( $1 \leq m_i \leq M, 1 \leq x_i \leq 10^9$ ).

A query will take the form `2 ai bi wi` ( $1 \leq a_i, b_i \leq N, 1 \leq w_i \leq 10^9, a_i \neq b_i$ ).

Note that the operations happen in the order specified in the input.

## Output Specification

---

For each query, print on a separate line `1` if the answer to the query is yes, and `0` otherwise.

## Sample Input

---

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

## Sample Output

---

```
0
1
1
0
```