

# Task: STA

## State Roads

---

UFAM Workshop, contest #3. Source file sta.\* Available memory: 128 MB.

The ancient Kingdom of Byteland had a road network designed for carriage transportation. Byteland was divided into several states. However, the division of the towns among the states was not fixed in time. Historians are now trying to recover the division at different points of time.

The only evidence the historians have about the examined period is which roads were classified as state roads at respective moments. The historians assume that the towns that were connected with a state road belonged to the same state. They are, however, uncertain if every pair of towns from the same state were connected by a state road and even if every pair of such towns were connected by a sequence of state roads (the connection could have used local roads which are not included in the historical data). Help the historians check how accurate are their predictions about states.

### Input

The first line of the input contains two integers  $n$  and  $q$  ( $1 \leq n \leq 1\,000\,000$ ,  $1 \leq q \leq 2\,000\,000$ ), which specify the number of towns in Byteland and the number of events. The towns are numbered from 1 to  $n$ .

Each of the following  $q$  lines describes an event:

- “1  $u$   $v$ ” means that towns  $u$  and  $v$  are now connected with a state road ( $1 \leq u < v \leq n$ ),
- “2  $m$ ” means that the road that became a state road as the  $m$ -th is not a state road anymore ( $m$  is between 1 and the number of events of type 1 so far),
- “3  $k$   $u_1$   $u_2$  ...  $u_k$ ” represents a query whether the set of towns  $\{u_1, u_2, \dots, u_k\}$  could have formed a single state at the considered moment of time ( $1 \leq k \leq n$ ,  $1 \leq u_1 < u_2 < \dots < u_k \leq n$ ).

Each road in Byteland is bidirectional. There is at most one road between any pair of different towns.

Initially there are no state roads. Any road may become a state road several times. No two events of type 2 share the same value of  $m$ . The sum of  $k$  values over all queries does not exceed 2 000 000.

### Output

For each query in the input, your program should output one line with the word YES or NO being the answer to the query.

### Example

For the input data:

```
4 10
3 3 1 3 4
1 1 2
3 3 1 3 4
1 2 3
3 2 1 3
3 3 1 2 3
1 3 4
1 2 4
2 1
3 3 2 3 4
```

the correct result is:

```
YES
NO
NO
YES
YES
```