## CCO '24 P6 - Telephone Plans

**Time limit:** 4.0s **Memory limit:** 1G

The "Dormi's Fone Service" is now the only telephone service provider in CCOland. There are N houses in CCOland, numbered from 1 to N. Each telephone line connects two distinct houses such that all the telephone lines that ever exist form a forest.

There is an issue where the phone lines are faulty, and each phone line only exists for a single interval of time. Two houses can call each other at a certain time if there is a path of phone lines that starts at one of the houses and ends in the other house at that time.

We would like to process Q queries of the following forms:

- 1 x y: Add a phone line between houses x and y. It is guaranteed that a phone line between houses x and y was never added before.
- 2 x y: Remove the phone line between houses x and y. It is guaranteed that a phone line currently exists between houses x and y.
- 3 t: Compute the number of pairs of different houses that can call each other at some time between the current query and t queries ago. To be more clear, let  $G_q$  be the state of CCOland after the q-th query, where  $G_0$  is the state of CCOland before any queries. If this is the s-th query, then count the number of pairs of houses that are connected in at least one of  $G_{s-t}, G_{s-t+1}, \ldots, G_s$ .

Also, some test cases may be encrypted. For the test cases that are encrypted, the arguments x, y, or t are given as the bitwise xor of the true argument and the answer to the last query of type 3 (if there have been no queries of type 3, then the arguments are unchanged).

#### **Input Specification**

The first line of input will contain E ( $E \in \{0,1\}$ ). E=0 denotes that the input is not encrypted, while E=1 denotes that the input is encrypted.

The second line contains two space-separated integers N and Q, representing the number of houses in CCOland and the number of queries, respectively.

The next Q lines contain queries as specified above (queries are encrypted or not depending on E).

For the q-th query  $(1 \le q \le N)$ , it is guaranteed that (after decrypting if E = 1)  $1 \le x, y \le N$  for type 1 and 2 queries and  $0 \le t \le q$  for type 3 queries.

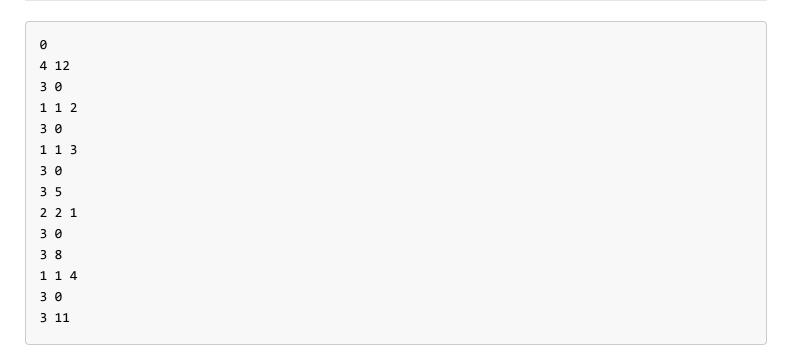
Marks Awarded	Bounds on $N$	Bounds on ${\it Q}$	Encrypted?
3 marks	$1 \leq N \leq 30$	$1 \leq Q \leq 150$	E=0
2 marks	$1 \leq N \leq 30$	$1 \leq Q \leq 150$	E=1
4 marks	$1 \leq N \leq 2000$	$1 \leq Q \leq 6000$	E=0

Marks Awarded	Bounds on $N$	Bounds on ${\it Q}$	Encrypted?
2 marks	$1 \leq N \leq 2000$	$1 \leq Q \leq 6000$	E=1
4 marks	$1 \leq N \leq 100000$	$1 \leq Q \leq 300000$	E=0
5 marks	$1 \leq N \leq 100000$	$1 \leq Q \leq 300000$	E=1
5 marks	$1 \leq N \leq 500000$	$1 \leq Q \leq 1500000$	E=1

# **Output Specification**

For each query of type 3, output the answer to the query on a new line.

## Sample Input 1



## **Sample Output 1**

```
      0

      1

      3

      1

      3

      3

      3

      5
```

#### **Explanation of Output for Sample Input 1**

This test case is not encrypted.

For the  $1^{\rm st}$  query, no pairs of different houses could have called each other.

For the  $3^{\rm rd}$  query, only houses 1 and 2 could have called each other.

For the  $5^{\rm th}$  query, (1,2),(1,3),(2,3) is the set of pairs that could have called each other.

The  $6^{\rm th}$  query is the same.

For the  $8^{\rm th}$  query, only houses 1 and 3 could have called each other.

For the  $9^{\rm th}$  query, there is a point in time where (1,2),(1,3),(2,3) could have called each other.

For the  $11^{th}$  query, the set of pairs that could have called each other is (1,3),(1,4),(3,4).

For the  $12^{th}$  query, the set of pairs that could have called each other at any previous time is (1,2),(1,3),(1,4),(2,3),(3,4).

#### **Explanation of Output for Sample Input 2**

Encrypted version of sample 1.

#### Sample Input 2

```
1
4 12
3 0
1 1 2
3 0
1 0 2
3 1
3 6
2 1 2
3 3
3 9
1 2 7
3 3
3 8
```

#### **Sample Output 2**

0				
1				
3				
3				
1				
3				
_				
3				
5				