

Aragorn The King

locked

Problem

Submissions

Time Limit: $C/C++$ (1s) , $Java$ (2s)

Memory Limit: 512MB

In the realm of Middle-earth, a formidable and malevolent force emerged. Sauron, the Lord of the Rings, aspired to subjugate all of Middle-earth and mold it according to his will. To aid in his quest for dominion, he forged the One Ring, a weapon of immense power. Before Sauron's might, Aragorn, the king of Gondor, was defeated and forced to abandon his kingdom.

Having left his homeland behind, Aragorn discovered a new realm consisting of N cities, numbered from 1 to N , separated by water. He established the capital of his new kingdom in the city numbered 1 and began assembling his army. Gradually, he would select any two cities randomly and construct a two-way bridge between them **if any one of these cities were accessible from the capital** as building materials were always supplied from the capital. To save resources, **Aragorn refrained from constructing bridges if it was already possible to reach the capital from both the cities**. Occasionally, he would wander between cities, searching for signs of enemies.

Meanwhile, Sauron remained vigilant, ceaselessly dispatching his Orc forces to these cities with the intent of attacking and annihilating the capital. However, this time, Aragorn possessed the *Aetheric Nullifier*, an ancient weapon capable of swiftly obliterating all designated bridges. Whenever Aragorn encountered enemies within his kingdom, he would resort to destroying the minimum number of bridges necessary to prevent the enemy from reaching the capital.

The battle between Aragorn and Sauron raged on, with each side employing their strategies and weapons. The fate of Middle-earth hung in the balance, as the forces of good and evil clashed in a cataclysmic struggle for supremacy.

Aragorn appointed you to build a system that will perform Q operations of 3 different types specified below:

- **1 $U\ V$ ($1 \leq U, V \leq N$)** - Build a bridge between city U and city V . The bridge shall not be built, if it unveils a path to the capital for the Orc forces sent earlier.
- **2 $K\ A_1\ A_2 \dots A_i \dots A_K$ ($2 \leq A_i \leq N$)** - A list of cities of size K is given where Orc forces start their exploration. Find a minimum number of bridges to destroy such that the enemies of these cities can not reach the capital. After that, destroy those bridges using the Aetheric Nullifier. If there are multiple sets of bridges, destroy only them such that a maximum number of cities stay connected with the capital. Having failed in their quest to seize the capital, the Orc forces now occupy these cities, biding their time in the hope of one day to track down Aragorn.
- **3 $U\ V$ ($1 \leq U, V \leq N$)** - Find the minimum number of cities Aragorn needs to visit including city U if he wishes to travel from city U to city V . Of course he will not visit a city where the Orc forces those are sent earlier may reach and attack him. Print -1 if it is impossible to reach city V from city U .

Input Format

Input starts with an integer T denoting test cases.

For each test case, input starts with 2 integers N, Q — the total number of cities and the number of operations.

Next Q lines contain operations — one operation per line.

Constraints

$$1 \leq T \leq 10000, 2 \leq N \leq 10^5, 1 \leq Q \leq 3 * 10^5$$

The overall summation of N, Q and K is $\leq 4 * 10^6$

Output Format

For each case, print "Case #c:" in a line, where **c** is the test case number.

In the next **Q** lines, print the output of the queries.

- For each query of type **1**, print **1** if it is possible to build a bridge otherwise print **−1** in a line.
- For each query of type **2**, print the total number of bridges to be destroyed in a line.
- For each query of type **3**, print the minimum number of cities Aragorn needs to visit in a line.

Note. Follow the samples for the exact output format.

Sample Input 0

```
2
6 7
1 2 1
1 3 2
1 4 3
1 5 2
1 6 5
3 4 6
2 1 6
7 9
1 2 1
1 3 1
1 4 2
1 5 3
3 5 4
2 2 2 3
1 6 4
3 5 4
3 7 7
```

Sample Output 0

```
Case #1:
1
1
1
1
1
5
1
Case #2:
1
1
1
1
5
2
-1
-1
1
```

Explanation 0

The kingdom for the 2nd case would look like the image below:

Fig: A

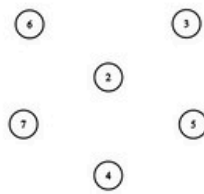


Fig: B

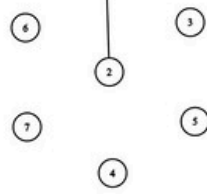


Fig: C

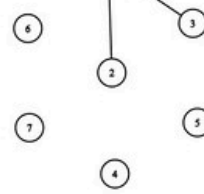


Fig: D

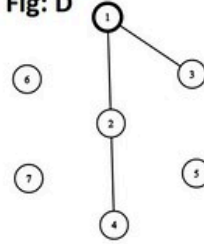


Fig: E

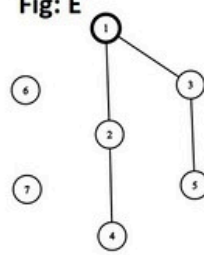
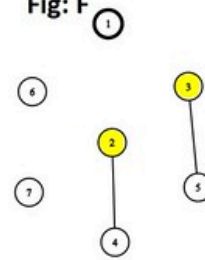


Fig: F



Submissions: 5

Max Score: 1

Rate This Challenge:

[More](#)

C



```

1 #include <stdio.h>
2 #include <string.h>
3 #include <math.h>
4 #include <stdlib.h>
5
6 int main() {
7
8     /* Enter your code here. Read input from STDIN. Print output to STDOUT */
9     return 0;
10 }
  
```

Line: 1 Col: 1

[Upload Code as File](#)
☐ Test against custom input

Run Code

Submit Code