

Police Business

Cops are one of the most fascinating types of people. In the movies they are usually shown as fat and lazy, but they are so much more! Our today's specimen, officer Acab is, for example, very much into philosophy. All the criminals are afraid of Acab so when he appears in a city no criminals can come there. He's often engaged in car chases, so he usually asks himself the following questions: If I know a criminal has to travel from city a to city b, how many cities are there (other than a and b) such that there are no other cops there, and if I come to that city the criminal won't be able to accomplish his trip? If I sort those cities by distance from a, which one will be the k-th in the sorted list? How many roads are there such that there are no other cops on them, and if I'm present on that road the criminal won't be able to travel from a to b? Which is the k-th such road if I sort the available roads by their distance from a?

Given a list of bidirectional roads that connect the cities write a program that will answer Acab's questions. In the beginning we know that there are no other cops in any of the cities. There will be one or more paths between each pair of cities.

Sometimes Acab's cop-friends contact him to tell him they have entered a city or road. There is never more than one cop present in a city or on a road. Thus, when a road or city is reported for the second time, we assume the cop has left there. This only means that after each even report of the same city or a road there are no cops there, and for odd reports there is a cop there.

Note: we define the distance of a road from a city as the minimum of the distances of its endpoints from that city. Even though Acab is a good cop, he doesn't have any special powers such as multilocation, so he can only be present in one city at a time. If you, for some reason, find two cities or roads equidistant from the city a, output the one with the smaller index. **Also, the other cops aren't as good as Acab, so they only block Acab from visiting a city, and not the criminals.**

Input

The first line of input contains two integers N and M ($1 \leq N \leq M \leq 100000$).

The next M lines contain a pair of integers a and b ($1 \leq a, b \leq N$).

All the cities are numbered from 1 to N.

The next line contains a single integer Q ($1 \leq Q \leq 200000$).

The next Q lines contain queries. There are six possible types of queries:

- 1 n - a cop has contacted Acab to let him know he's in city n
- 2 e - a cop has contacted Acab to let him know he's on the road e
- 3 a b - tell Acab how many cities he can block the criminal with
- 4 a b - tell Acab how many roads he can block the criminal with
- 5 a b k - tell Acab which is the k-th city he can block the criminal with
- 6 a b k - tell Acab which is the k-th road he can block the criminal with

Output

For each query of type 3, 4, 5 or 6 output a single line containing the answer. If for a query of type

5 or 6 k is greater than the actual number of possible cities, output -1. Also, in queries 2 and 6 the number of the road is assumed to be its index from the input.

Example

Input:

```
9 10
1 2
1 3
2 4
2 5
4 5
4 6
3 7
3 8
7 8
8 9
10
3 6 9
4 6 9
5 6 9 2
6 6 9 2
1 2
5 6 9 2
2 1
6 6 9 2
1 2
5 6 9 2
```

Output:

```
5
4
2
1
1
2
2
```

Explanation:

In the first query, there are 5 blockable cities: 4, 2, 1, 3, 8.

In the second query there are 4 blockable roads: 6, 1, 2, 10.

In the third query the second city is 2 (4, 2, 1, 3, 8).

In the fourth query the second road is 1 (6, 1, 2, 10).

In the fifth query we've been reported there's a cop in city 2.

In the sixth query the second blockable city is 1 (city 2 already contains a cop).

In the seventh query we've been reported there's a cop on road 1.

In the eighth query the second blockable road is 2 (road 1 already contains a cop).

In the ninth query we've been reported the cop from city 2 has left.

In the tenth query the second city is 2 again.