D. Beard Graph

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Let's define a non-oriented connected graph of n vertices and $n - 1$ edges as a *beard*, if all of its vertices except, perhaps, one, have the degree of 2 or 1 (that is, there exists no more than one vertex, whose degree is more than two). Let us remind you that the degree of a vertex is the number of edges that connect to it.

Let each edge be either black or white. Initially all edges are black.

You are given the description of the beard graph. Your task is to analyze requests of the following types:

- paint the edge number i black. The edge number i is the edge that has this number in the description. It is guaranteed that by the moment of this request the i-th edge is white
- paint the edge number i white. It is guaranteed that by the moment of this request the i-th edge is black
- find the length of the shortest path going **only along the black edges** between vertices a and b or indicate that no such path exists between them (a path's length is the number of edges in it)

The vertices are numbered with integers from 1 to n, and the edges are numbered with integers from 1 to $n - 1$.

Input

The first line of the input contains an integer n $(2 \le n \le 10^5)$ — the number of vertices in the graph. Next $n - 1$ lines contain edges described as the numbers of vertices v_i, u_i $(1 \le v_i, u_i \le n, v_i \ne u_i)$ connected by this edge. It is guaranteed that the given graph is connected and forms a beard graph, and has no self-loops or multiple edges.

The next line contains an integer m $(1 \le m \le 3 \cdot 10^5)$ — the number of requests. Next m lines contain requests in the following form: first a line contains an integer $type$, which takes values from 1 to 3, and represents the request type.

If $type = 1$, then the current request is a request to paint the edge black. In this case, in addition to number $type$ the line should contain integer id $(1 \le id \le n - 1)$, which represents the number of the edge to paint.

If $type = 2$, then the current request is a request to paint the edge white, its form is similar to the previous request.

If $type = 3$, then the current request is a request to find the distance. In this case, in addition to $type$, the line should contain two integers a, b $(1 \le a, b \le n$, a can be equal to $b)$ — the numbers of vertices, the distance between which must be found.

The numbers in all lines are separated by exactly one space. The edges are numbered in the order in which they are given in the input.

Output

For each request to "find the distance between vertices a and b" print the result. If there is no path going only along the black edges between vertices a and b, then print "-1" (without the quotes). Print the results in the order of receiving the requests, separate the numbers with spaces or line breaks.

Examples

input

```
3
1 2
2 3
7
3 1 2
3 1 3
3 2 3
2 2
```

```
3 1 2
3 1 3
3 2 3
```

output

```
1
2
1
1
-1
-1
```

input

```
6
1 5
6 4
2 3
3 5
5 6
6
3 3 4
2 5
3 2 6
3 1 2
2 3
3 3 1
```

output

```
3
-1
3
2
```

Note

In the first sample vertices 1 and 2 are connected with edge number 1, and vertices 2 and 3 are connected with edge number 2. Before the repainting edge number 2 each vertex is reachable from each one along the black edges. Specifically, the shortest path between 1 and 3 goes along both edges.

If we paint edge number 2 white, vertex 3 will end up cut off from other vertices, that is, no path exists from it to any other vertex along the black edges.