D. Rowena Ravenclaw's Diadem

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Harry, upon inquiring Helena Ravenclaw's ghost, came to know that she told Tom Riddle or You-know-who about Rowena Ravenclaw's diadem and that he stole it from her.

Harry thought that Riddle would have assumed that he was the only one to discover the Room of Requirement and thus, would have hidden it there. So Harry is trying to get inside the Room of Requirement to destroy the diadem as he knows that it is a horcrux.

But he has to answer a puzzle in order to enter the room. He is given n objects, numbered from 1 to n. Some of the objects have a parent object, that has a lesser number. Formally, object i may have a parent object $parent_i$ such that $parent_i < i$.

There is also a type associated with each parent relation, it can be either of type 1 or type 2. Type 1 relation means that the child object is like a special case of the parent object. Type 2 relation means that the second object is **always** a part of the first object and all its special cases.

Note that if an object b is a special case of object a, and c is a special case of object b, then c is considered to be a special case of object a as well. The same holds for parts: if object b is a part of a, and object c is a part of b, then we say that object c is a part of a. Also note, that if object b is a part of a, and object c is a special case of a, then b is a part of c as well.

An object is considered to be neither a part of itself nor a special case of itself.

Now, Harry has to answer two type of queries:

- `1 u v`: he needs to tell if object v is a special case of object u.
- `2 u v`: he needs to tell if object v is a part of object u.

Input

First line of input contains the number n ($1 \le n \le 10^5$), the number of objects.

Next n lines contain two integer $parent_i$ and $type_i$ ($-1 \le parent_i < i$ $parent_i \ne 0$, $-1 \le type_i \le 1$), implying that the i-th object has the parent $parent_i$. (If $type_i = 0$, this implies that the object i is a special case of object $parent_i$. If $type_i = 1$, this implies that the object i is a part of object $parent_i$). In case the i-th object has no parent, both $parent_i$ and $type_i$ are `-1`.

Next line contains an integer q ($1 \le q \le 10^5$), the number of queries.

Next q lines each represent a query having three space separated integers $type_i, u_i, v_i$ ($1 \le type_i \le 2$, $1 \le u, v \le n$).

Output

Output will contain q lines, each containing the answer for the corresponding query as "`YES`" (affirmative) or "`NO`" (without quotes).

You can output each letter in any case (upper or lower).

Examples

input

```
3
-1 -1
1 0
2 0
```

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

output

```
YES
NO
```

input

```
3
-1 -1
1 0
1 1
2
2 2 3
2 3 2
```

output

```
YES
NO
```

Note

In test case 1, as object 2 is a special case of object 1 and object 3 is a special case of object 2, this makes object 3 a special case of object 1.

In test case 2, as object 2 is a special case of object 1 and object 1 has object 3, this will mean that object 2 will also have object 3. This is because when a general case (object 1) has object 3, its special case (object 2) will definitely have object 3.