Problem Name: Nearest Left Node in a Binary Tree

Topic: Trees

Tags: Trees, Graph

Language used: C++

Difficulty: Hard

Problem Statement:

Given a binary tree and a target node, your task is to find the nearest left node on the same level as the target node. If the target node is the leftmost node on its level, return `-1`.

You are given the root of a binary tree and a reference to a node `u` in the tree. Your task is to find and return the nearest node to the left of `u` on the same level. If no such node exists (i.e., `u` is the leftmost node on its level), return `-1`.

Input Format:

- The first line contains an integer `N`, the number of elements in level order traversal.
- The second line contains `N` integers, representing the level order traversal of the tree. Use `-1` for null nodes.
- The third line contains a single integer, the `targetVal` — the value of the target node.

Output Format:

- Output a single integer — the value of the nearest left node of the target node. If it doesn't exist, print `-1`.

Constraints:

- `1 ≤ N ≤ 10^5`
- `-10^9 ≤ Node Value ≤ 10^9`
- The binary tree will not contain duplicate values.
- The target node will always be present in the tree.

Sample Input 1:

```
N=13
root [] = 1 2 3 4 5 6 7 -1 -1 -1 -1 -1 -1
targetVal=5
```

Sample Output 1:

```
4
```

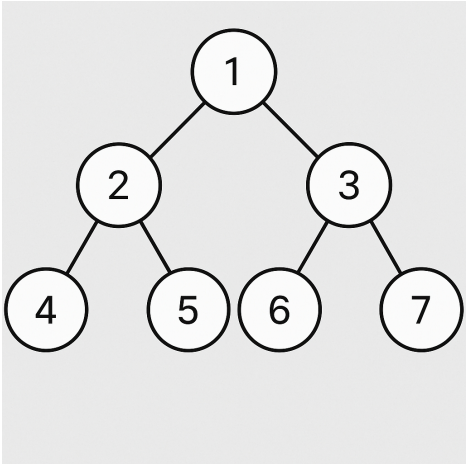
Sample Input 2:

```
N=13
root[]=1 2 3 4 5 6 7 -1 -1 -1 -1 -1 -1
targetVal=4
```

Sample Output 2:

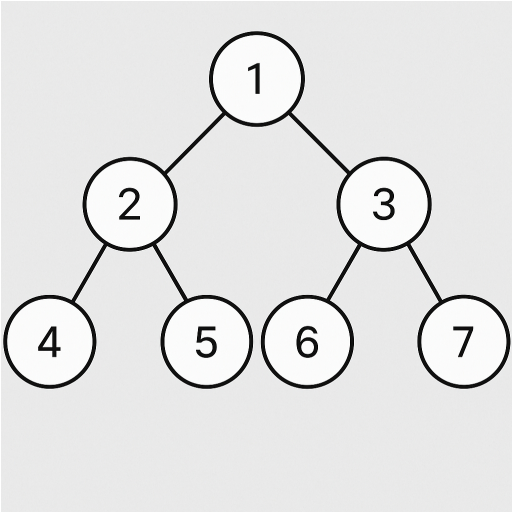
```
-1
```

Explanation of Sample Input 1:



====>> For TargetVal 5 , 4 is Nearest Left Node of Binary tree

Explanation of Sample Input 2:



====>> For TargetVal 4 , 4 it self if Lett most element so we return -1