Check whether BST contains Dead End

Given a Binary search Tree that contains positive integer values greater then 0. The task is to complete the function **isDeadEnd** which returns true if the BST contains a dead end else returns false. Here Dead End means, we are not able to insert any element after that node.

Input :
  8
/ \
5 9
/ \
2 7
/
1

Output : Yes
Explanation : Node "1" is the dead End because after that
  we cant insert any element.

Input :
  8
/ \
7 10
/ /
2 9 13

Output : Yes
Explanation : We can't insert any element at
node 9.

```
def isdeadEnd(root):
    # Code here
    seen = set()
    seen.add(0)
    return helper(root,seen)

def helper(root,seen):
    if root:
        if root.left is root.right:
            if root.data+1 in seen and root.data-1 in seen:
                return True
            seen.add(root.data)
        seen.add(root.data)
        return helper(root.left,seen) or helper(root.right,seen)
```