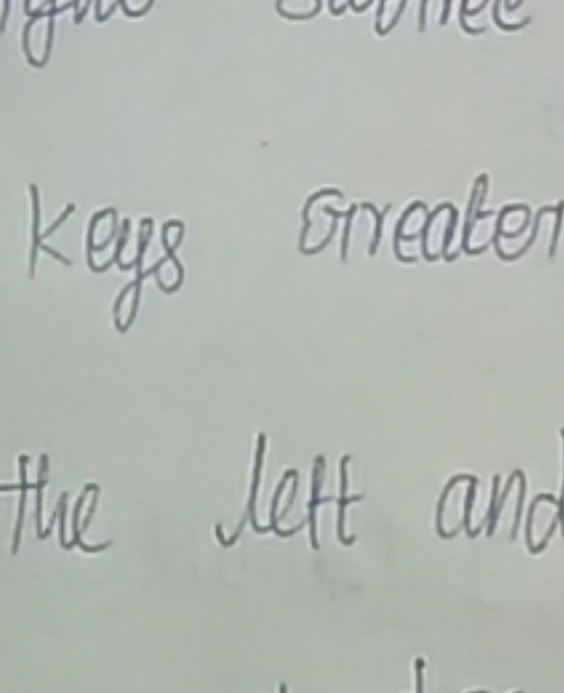
Day 26

Check if binary tree is BST or not

A binary tree data structure is ~~not~~ a node based binary tree data structure which has the following properties.

* The left subtree of node contains only nodes with keys less than nodes key
* The right subtree of node contains only nodes with keys greater than the nodes key
* Both the left and right subtree must also be binary search tree.



A better solution looks at each node only once. the trick is to write a utillity helper function

Implementation :

INT_MAX = 4294967296

INT_MIN = -4294967296

Class Node :

def __init__(self, data):

self.data = data

self.right = None

self.left = None

def isBST(node):

return (isBSTutill(node, INT_MIN, INT_Max))

def isBSTutill(node, mini, maxi):

if node is None:

return True

if node.data < mini or node.data > maxi:

return False

return (isBSTUtil(node.left, mini, node.data - 1)
and isBSTUtil(node.right, node.data + 1, maxi))

Input:

```
root = Node(4)
root.left = Node(2)
root.right = Node(5)
root.left.left = Node(1)
root.left.right = Node(3)
if (isBST(root)):
    print("is BST")
else:
    print("Not a BST")
```

Output:

```
is BST
```

Time Complexity:

O(n)