98. Validate Binary Search Tree

Medium

Given a binary tree, determine if it is a valid binary search tree (BST).

Assume a BST is defined as follows:

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

/\*\*

\* Definition for a binary tree node.

\* struct TreeNode {

\* int val;

\* TreeNode \*left;

\* TreeNode \*right;

\* TreeNode(int x) : val(x), left(NULL), right(NULL) {}

\* };

\*/

class Solution {

public:

bool isBST(TreeNode\* node, int min, int max){

if(node==NULL) return true;

if((min!=NULL&&node->val<=min)||(max!=NULL&&node->val>=max)) return false;

return isBST(node->left,min,node->val)&&isBST(node->right,node->val,max);

}

bool isValidBST(TreeNode\* root) {

return isBST(root, NULL, NULL);

}

};