

Validate if Tree is a BST (98)

A binary tree is valid if its inorder traversal is in ascending ~~order~~ (strictly) order.

One way is to get inorder traversal using recursion and check if it is strictly increasing.

Other way is to add one line that checks a parent and right child to the code of iterative inorder traversal.

CODE

```
bool isValidBST(TreeNode* root) {  
    if (root == NULL) return true;  
    stack<TreeNode*> s;  
    TreeNode* prev = -NULL;  
    while (root != NULL || !s.empty()) {  
        while (root != NULL) {  
            s.push(root);  
            root = root->left;  
        }  
        root = s.top(); s.pop();  
        if (prev != NULL && prev->val >= root->val) {  
            return false; }  
        prev = root;  
        root = root->right;  
    }  
    return true; }  
}
```

Time complexity $\rightarrow O(n)$

space complexity $\rightarrow O(n)$