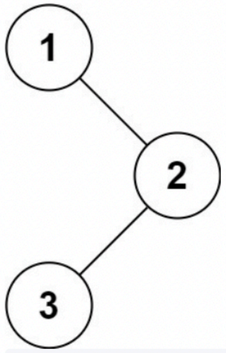
Description | Solution | Discuss (999+) | Submissions

94. Binary Tree Inorder Traversal

Easy 6652 285 Add to List Share

Given the `root` of a binary tree, return *the inorder traversal of its nodes' values*.

Example 1:




```
Input: root = [1,null,2,3]
Output: [1,3,2]
```

Example 2:

```
Input: root = []
Output: []
```

Example 3:

```
Input: root = [1]
Output: [1]
```

Java | Autocomplete

```
 *          this.val = val;
 *          this.left = left;
 *          this.right = right;
 *      }
 * }
 */
class Solution {
    public List<Integer> inorderTraversal(TreeNode root) {

        List<Integer> results = new ArrayList<>();
        Stack<TreeNode> stack = new Stack<>();
        TreeNode node = root;

        while(!stack.isEmpty() || node != null) {
            if(node != null) {
                stack.push(node);
                node = node.left;
            } else {
                node = stack.pop();
                results.add(node.val);
                node = node.right;
            }
        }

        return results;
    }
}
```

Testcase | Run Code Result | Debugger

Accepted Runtime: 0 ms

Your input	[1,null,2,3]
Output	[1,3,2]
Expected	[1,3,2]

Diff