

94. Binary Tree Inorder Traversal

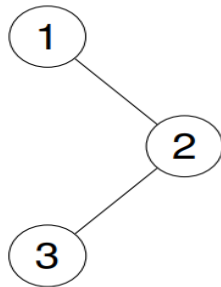
Given the root of a binary tree, return *the inorder traversal of its nodes' values*.

Example 1:

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

Output: [1,3,2]

Explanation:

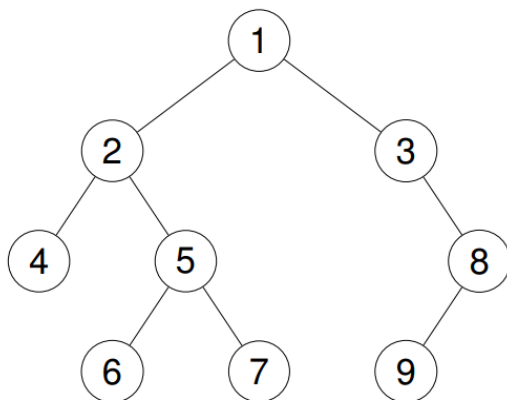


Example 2:

Input: root = [1,2,3,4,5,null,8,null,null,6,7,9]

Output: [4,2,6,5,7,1,3,9,8]

Explanation:



Example 3:**Input:** root = []**Output:** []**Example 4:****Input:** root = [1]**Output:** [1]**Constraints:**

- The number of nodes in the tree is in the range [0, 100].
- $-100 \leq \text{Node.val} \leq 100$

```
# Definition for a binary tree node.
# class TreeNode(object):
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right
class Solution(object):
    def inorderTraversal(self, root):
        def helper(root,result):
            if root != None:
                helper(root.left,result)
                result.append(root.val)
                helper(root.right,result)

        result = []
        helper(root,result)
        return result

    if not root:
        return
    inorderTraversal(root.left)
    arr.append(root.val)
    inorderTraversal(root.right)
    return arr
```