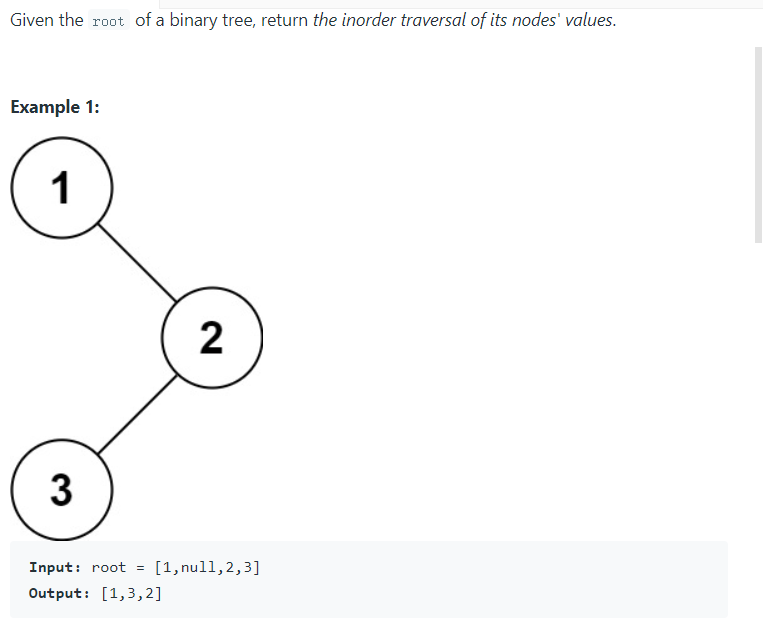
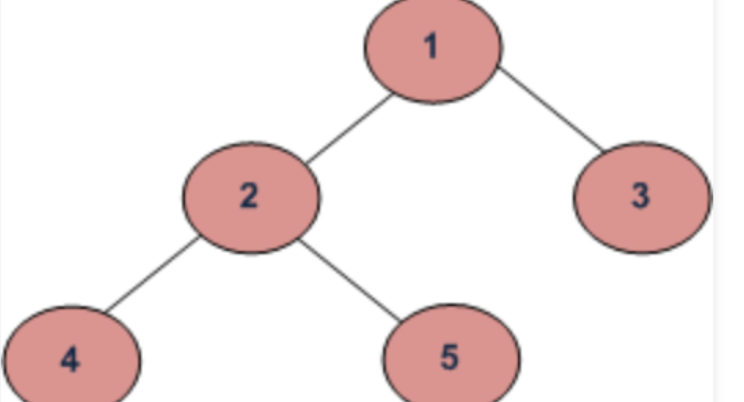
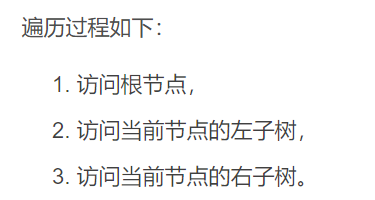
**94. Binary Tree Inorder Traversal**

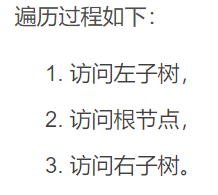


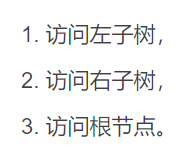
其实就是基础知识：

preorder:



.1.2.4.5.3

Inorder中序 42513

postorder后序45231

这里就是中序

/\*\*

\* Definition for a binary tree node.

\* public class TreeNode {

\* int val;

\* TreeNode left;

\* TreeNode right;

\* TreeNode() {}

\* TreeNode(int val) { this.val = val; }

\* TreeNode(int val, TreeNode left, TreeNode right) {

\* this.val = val;

\* this.left = left;

\* this.right = right;

\* }

\* }

\*/

class Solution {

public List<Integer> inorderTraversal(TreeNode root) { //其实没啥可说的，因为我们要return 一个List，所以要新建一个function

List<Integer> result=new ArrayList<Integer>();

inorder(result,root);

return result;

}

public void inorder(List<Integer> result, TreeNode node){ //注意这里的参数，虽然实际上是ArrayList,但我们initial的时候是List,所以这里是List ，多态，我们不运行的时候不知道他是不是ArrayList

if(node==null) return;

inorder(result,node.left);

result.add(node.val);

inorder(result,node.right);

}

}