**先序遍历：https://leetcode-cn.com/problems/binary-tree-preorder-traversal/**

var inorderTraversal = function(root) {

var res=[];

inorder(root,res);

return res;

};

//按照根 左 右顺序遍历

function inorder(root,res){

if(!root) return ;

res.push(root.val);

inorder(root.left,res);

inorder(root.right,res);

}

**中序遍历：https://leetcode-cn.com/problems/binary-tree-inorder-traversal/**

var inorderTraversal = function(root) {

var res=[];

inorder(root,res);

return res;

};

//按照左 根 右顺序遍历

function inorder(root,res){

if(!root) return ;

inorder(root.left,res);

res.push(root.val);

inorder(root.right,res);

}

**后序遍历：https://leetcode-cn.com/problems/binary-tree-postorder-traversal/**

var inorderTraversal = function(root) {

var res=[];

inorder(root,res);

return res;

};

//按照左 右 根顺序遍历

function inorder(root,res){

if(!root) return ;

inorder(root.left,res);

inorder(root.right,res);

res.push(root.val);

}

**层次遍历：https://leetcode-cn.com/problems/binary-tree-level-order-traversal/submissions/**

const levelOrder = root => {

if (!root) return []

let res = [], queue = [root]

while (queue.length) {

let arr = [], temp = []

while (queue.length) {

let curr = queue.shift()

arr.push(curr.val)

if (curr.left) temp.push(curr.left)

if (curr.right) temp.push(curr.right)

}

queue = temp

res.push(arr)

}

return res

}