

# 二叉树：总结篇！（需要掌握的二叉树技能都在这里了）

---

力扣二叉树大总结！

在每一道二叉树的题目中，都使用了**递归三部曲**来分析题目，相信以后看到二叉树，看到递归，都会想：返回值、参数是什么？终止条件是什么？单层逻辑是什么？

而且**几乎每一道题目我都给出对应的迭代法**，可以用来进一步提高自己。

## 二叉树的理论基础

---

- 关于二叉树，你该了解这些！：二叉树的种类、存储方式、遍历方式、定义方式

## 二叉树的遍历方式

---

- 深度优先遍历
  - 二叉树：前中后序递归法：递归三部曲初次亮相
  - 二叉树：前中后序迭代法（一）：通过栈模拟递归
  - 二叉树：前中后序迭代法（二）统一风格
- 广度优先遍历
  - 二叉树的层序遍历：通过队列模拟

## 二叉树的属性

---

- 二叉树：是否对称
  - 递归：后序，比较的是根节点的左子树与右子树是不是相互翻转
  - 迭代：使用队列/栈将两个节点顺序放入容器中进行比较
- 二叉树：求最大深度
  - 递归：后序，求根节点最大高度就是最大深度，通过递归函数的返回值做计算树的高度
  - 迭代：层序遍历
- 二叉树：求最小深度
  - 递归：后序，求根节点最小高度就是最小深度，注意最小深度的定义
  - 迭代：层序遍历
- 二叉树：求有多少个节点
  - 递归：后序，通过递归函数的返回值计算节点数量
  - 迭代：层序遍历
- 二叉树：是否平衡
  - 递归：后序，注意后序求高度和前序求深度，递归过程判断高度差
  - 迭代：效率很低，不推荐
- 二叉树：找所有路径
  - 递归：前序，方便让父节点指向子节点，涉及回溯处理根节点到叶子的所有路径
  - 迭代：一个栈模拟递归，一个栈来存放对应的遍历路径
- 二叉树：递归中如何隐藏着回溯
  - 详解二叉树：找所有路径递归如何隐藏着回溯
- 二叉树：求左叶子之和

- 递归：后序，必须三层约束条件，才能判断是否是左叶子。
  - 迭代：直接模拟后序遍历
- 二叉树：求左下角的值
  - 递归：顺序无所谓，优先左孩子搜索，同时找深度最大的叶子节点。
  - 迭代：层序遍历找最后一行最左边
- 二叉树：求路径总和
  - 递归：顺序无所谓，递归函数返回值为bool类型是为了搜索一条边，没有返回值是搜索整棵树。
  - 迭代：栈里元素不仅要记录节点指针，还要记录从头结点到该节点的路径数值总和

## 二叉树的修改与构造

---

- 翻转二叉树
  - 递归：前序，交换左右孩子
  - 迭代：直接模拟前序遍历
- 构造二叉树
  - 递归：前序，重点在于找分割点，分左右区间构造
  - 迭代：比较复杂，意义不大
- 构造最大的二叉树
  - 递归：前序，分割点为数组最大值，分左右区间构造
  - 迭代：比较复杂，意义不大
- 合并两个二叉树
  - 递归：前序，同时操作两个树的节点，注意合并的规则
  - 迭代：使用队列，类似层序遍历

## 求二叉搜索树的属性

---

- 二叉搜索树中的搜索
  - 递归：二叉搜索树的递归是有方向的
  - 迭代：因为有方向，所以迭代法很简单
- 是不是二叉搜索树
  - 递归：中序，相当于变成了判断一个序列是不是递增的
  - 迭代：模拟中序，逻辑相同
- 求二叉搜索树的最小绝对差
  - 递归：中序，双指针操作
  - 迭代：模拟中序，逻辑相同
- 求二叉搜索树的众数
  - 递归：中序，清空结果集的技巧，遍历一遍便可求众数集合
  - 二叉搜索树转成累加树
  - 递归：中序，双指针操作累加
  - 迭代：模拟中序，逻辑相同

## 二叉树公共祖先问题

---

- 二叉树的公共祖先问题
  - 递归：后序，回溯，找到左子树出现目标值，右子树节点目标值的节点。
  - 迭代：不适合模拟回溯
- 二叉搜索树的公共祖先问题

- ## 二叉搜索树的修改与构造

- ## 最后总结

- 涉及到二叉树的构造，无论普通二叉树还是二叉搜索树一定前序，都是先构造中节点。
- 求普通二叉树的属性，一般是后序，一般要通过递归函数的返回值做计算。
- 求二叉搜索树的属性，一定是中序了，要明白瞎了有序性了。

二叉树专题汇聚为一张图:

