

二叉搜索树的修改和构造

二叉搜索树应时刻想到中序遍历，并注意什么场景下用递归合适

二叉搜索树的插入操作

方法：充分利用二叉搜索树的性质，如果要插入的值小于当前遍历节点cur的值，则将cur往左，大于则往右，直至cur为空；注意设置temp储存上一次遍历的节点，循环结束后判断插入的值和temp节点值的大小关系，大于temp则作为temp的右节点，小于则作为左节点

题目：

701. 二叉搜索树的插入操作（中等）

删除二叉搜索树中的节点

两大核心：找到要删除的节点 & 进行删除操作，采用递归法解决

递归三部曲：

(1) 递归参数和返回类型：参数即为二叉树节点root和删除的节点值key，返回类型为节点，因此直接将主函数作为递归函数

(2) 递归终止条件：分为两种大情况，找到和没找到

a. 没找到：即当root节点为空时，此时返回空节点root

b. 找到了：即root节点为删除节点，分为四种小情况

i. root既没有左子节点也没有右子节点，此时直接删除，返回null（可以包含在ii、iii中，不用单独写这种情况）

ii. root有左子节点但无右子节点，此时返回root.left

iii. root无左子节点但有右子节点，此时返回root.right

iv. root既有左节点也有右子节点，此时需要将root左节点整体移到root右节点中最左节点下方，作为这个最左节点的左子节点，然后将root变为root.right并返回root

(3) 递归逻辑：分为两种情况，最后返回root

a. 如果删除节点的值小于root节点的值，则对root.left向左递归

b. 如果删除节点的值大于root节点的值，则对root.right向右递归

题目：

450. 删除二叉搜索树中的节点（中等）

修剪二叉搜索树

修剪规则：

- （1）如果节点小于区间下界，那么它和它的左子树应该全部被修剪掉，仅保留它的右子树
- （2）如果节点大于区间上界，那么它和它的右子树应该全部被修剪掉，仅保留它的左子树

具体步骤：采用迭代法

- （1）先**从根节点开始**按修剪规则进行修剪，确定新的根节点，循环条件是节点不为空且节点值不在区间内
- （2）**以新的根节点为起点**，对它的左子树进行for循环处理，考虑根节点的左节点，如果它的值小于区间下界，则把它的右节点作为根节点的左节点，然后往左遍历做判断，直至遍历节点为空
- （3）按（2）的方式对根节点的右子树进行处理，最后返回根节点

***此题也可采用递归（详情看递归代码），但是迭代的空间复杂度更优**

题目：

450. 修剪二叉搜索树（中等）

将有序数组转换为二叉搜索树

思路：**二分法+递归**

（本题思路不难想到，每次以**二分法求得的中间值作为当前根节点**）

递归三部曲：

- （1）**递归参数及返回类型**：由于要二分法，因此需要左边界和右边界left和right，同时需要传递数组参数，返回类型为节点。因此**递归函数需要自行编写**，然后在主函数中调用
- （2）**递归终止条件**：左边界大于右边界时停止递归，因此在二分法程序中的while循环外直接返回空

(3) 递归逻辑： $\text{left} \leq \text{right}$ 时按 $\text{middle} = (\text{left} + \text{right})/2$ 的方式确定 root 节点，递归时 root.left 的左右边界分别为 left 和 middle - 1，root.right 的左右边界分别为 middle + 1 和 right，两次递归后循环内直接返回 root 结束程序

题目：

108. 将有序数组转换为二叉搜索树（简单）

把二叉搜索树转换为累加树

累加树是如何形成的： 是对二叉树进行**中序遍历**后，**从后往前累加**得到

由于是从后往前累加，因此本题对二叉树进行**反中序遍历（右中左）**，每次需要存储上一个弹出栈的节点值

题目：

538. 把二叉搜索树转换为累加树（中等）