

0285. 二叉搜索树中的中序后继

👤 [ITCharge](#) ⌚ 大约 1 分钟

- 标签：树、深度优先搜索、二叉搜索树、二叉树
- 难度：中等

题目链接

- [0285. 二叉搜索树中的中序后继 - 力扣](#)

题目大意

给定一棵二叉搜索树的根节点 `root`。

要求：按中序遍历顺序将其重新排列为一棵递增顺序搜索树，使树中最左边的节点成为树的根节点，并且每个节点没有左子节点，只有一个右子节点。

解题思路

可以分为两步：

1. 中序遍历二叉搜索树，将节点先存储到列表中。
2. 将列表中的节点构造成一棵递增顺序搜索树。

中序遍历直接按照 左 -> 根 -> 右 的顺序递归遍历，然后将遍历的节点存储到 `res` 中。

构造递增顺序搜索树，则用 `head` 保存头节点位置。遍历列表中的每个节点，将其左右指针先置空，再将其连接在上一个节点的右子节点上。

最后返回 `head.right` 即可。

代码

py

```
class Solution:
    def inOrder(self, root, res):
        if not root:
            return
        self.inOrder(root.left, res)
        res.append(root)
        self.inOrder(root.right, res)

    def increasingBST(self, root: TreeNode) -> TreeNode:
        res = []
        self.inOrder(root, res)

        if not res:
            return
        head = TreeNode(-1)
        cur = head
        for node in res:
            node.left = node.right = None
            cur.right = node
            cur = cur.right
        return head.right
```