0270. 最接近的二叉搜索树值

ITCharge 大约 2 分钟

- 标签：树、深度优先搜索、二叉搜索树、二分查找、二叉树
- 难度：简单

题目链接

- 0270. 最接近的二叉搜索树值 - 力扣

题目大意

描述：给定一个不为空的二叉搜索树的根节点，以及一个目标值 $target$。

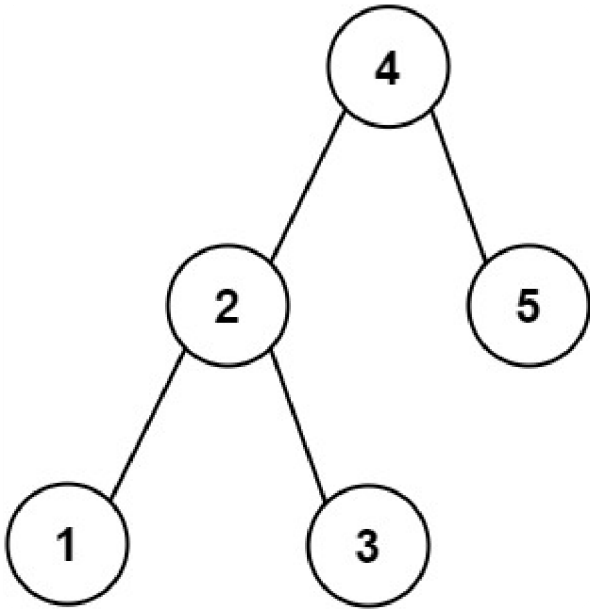
要求：在二叉搜索树中找到最接近目标值 $target$ 的数值。

说明：

- 树中节点的数目在范围 $[1, 10^4]$ 内。
- $0 \leq Node.val \leq 10^9$。
- $-10^9 \leq target \leq 10^9$。

示例：

- 示例 1：



```py
输入：root = [4,2,5,1,3], target = 3.714286
输出：4
```

- 示例 2：

```py
输入：root = [1], target = 4.428571
输出：1
```

解题思路

思路 1：二分查找算法

题目中最接近目标值 $target$ 的数值指的就是与 $target$ 相减绝对值最小的数值。

而且根据二叉搜索树的性质，我们可以利用二分搜索的方式，查找与 $target$ 相减绝对值最小的数值。具体做法为：

- 定义一个变量 $closest$ 表示与 $target$ 最接近的数值，初始赋值为根节点的值 $root.val$。
- 判断当前节点的值域 $closet$ 值哪个更接近 $target$，如果当前值更接近，则更新 $closest$。
- 如果 $target$ < 当前节点值，则从当前节点的左子树继续查找。
- 如果 $target$ ≥ 当前节点值，则从当前节点的右子树继续查找。

思路 1：代码

```py
class Solution:
    def closestValue(self, root: TreeNode, target: float) -> int:
        closest = root.val
        while root:
            if abs(target - root.val) < abs(target - closest):
                closest = root.val
            if target < root.val:
                root = root.left
            else:
                root = root.right
        return closest
```

思路 1：复杂度分析

- **时间复杂度**：$O(\log n)$，其中 n 为二叉搜索树的节点个数。
- **空间复杂度**：$O(1)$。