

Problem 270: Closest Binary Search Tree Value

Problem Information

Difficulty: Easy

Acceptance Rate: 49.39%

Paid Only: Yes

Tags: Binary Search, Tree, Depth-First Search, Binary Search Tree, Binary Tree

Problem Description

Given the `root` of a binary search tree and a `target` value, return the value in the BST that is closest to the `target`. If there are multiple answers, print the smallest.

Example 1:



Input: `root = [4,2,5,1,3]`, `target = 3.714286` **Output:** 4

Example 2:

Input: `root = [1]`, `target = 4.428571` **Output:** 1

Constraints:

* The number of nodes in the tree is in the range `[1, 104]`.
* `0 <= Node.val <= 109`
* `-109 <= target <= 109`

Code Snippets

C++:

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
```

```

* int val;
* TreeNode *left;
* TreeNode *right;
* TreeNode() : val(0), left(nullptr), right(nullptr) {}
* TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
* TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left),
right(right) {}
* };
*/

class Solution {
public:
int closestValue(TreeNode* root, double target) {

}
};

```

Java:

```

/**
 * Definition for a binary tree node.
 * public class TreeNode {
 *     int val;
 *     TreeNode left;
 *     TreeNode right;
 *     TreeNode() {}
 *     TreeNode(int val) { this.val = val; }
 *     TreeNode(int val, TreeNode left, TreeNode right) {
 *         this.val = val;
 *         this.left = left;
 *         this.right = right;
 *     }
 * }
 */

class Solution {
public int closestValue(TreeNode root, double target) {

}
}

```

Python3:

```
# Definition for a binary tree node.
# class TreeNode:
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right
class Solution:
    def closestValue(self, root: Optional[TreeNode], target: float) -> int:
```