272. Closest Binary Search Tree Value II

Hard 570 20 Add to List Share

Given a non-empty binary search tree and a target value, find *k* values in the BST that are closest to the target.

Note:

- Given target value is a floating point.
- You may assume *k* is always valid, that is: *k* ≤ total nodes.
- You are guaranteed to have only one unique set of *k* values in the BST that are closest to the target.

Example:

```
Input: root = [4,2,5,1,3], target = 3.714286, and k = 2

    4
   / \
  2   5
 / \
1   3

Output: [4,3]
```

Follow up:
Assume that the BST is balanced, could you solve it in less than *O*(*n*) runtime (where *n* = total nodes)?

Accepted 53,735 | Submissions 108,635

Seen this question in a real interview before? Yes No

Contributor

Companies

Related Topics

Similar Questions

Show Hint 1

Show Hint 2

Show Hint 3

Show Hint 4

Python

Autocomplete

```python
# Definition for a binary tree node.
# class TreeNode(object):
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right
class Solution(object):
    def closestKValues(self, root, target, k):
        """
        :type root: TreeNode
        :type target: float
        :type k: int
        :rtype: List[int]
        """

```

Console Contribute