783 Minimum Distance Between BST Nodes

2018年5月4日　17:34

Given a Binary Search Tree (BST) with the root node root, return the minimum difference between the values of any two different nodes in the tree.

Example :

Input: root = [4,2,6,1,3,null,null]

Output: 1

Explanation:

Note that root is a TreeNode object, not an array.

The given tree [4,2,6,1,3,null,null] is represented by the following diagram:

```
4
      /  \
     2     6
    / \
   1   3
```

while the minimum difference in this tree is 1, it occurs between node 1 and node 2, also between node 3 and node 2.

Note:

1. The size of the BST will be between 2 and 100.
2. The BST is always valid, each node's value is an integer, and each node's value is different.

来自 <https://leetcode.com/problems/minimum-distance-between-bst-nodes/description/>

给定一个二叉搜索树的根结点 root, 返回树中任意两节点的差的最小值。

注意:

1. 二叉树的大小范围在 2 到 100。
2. 二叉树总是有效的，每个节点的值都是整数，且不重复。

Solution for Python3:

```python
# Definition for a binary tree node.
# class TreeNode:
#     def __init__(self, x):
#         self.left = None
#         self.right = None
class Solution:
    def minDiffInBST(self, root):
        """
        :type root: TreeNode
        :rtype: int
        """
        self.ans, self.pre = float('inf'), -float('inf')
        def inOrder(root):
            if root.left:
               inOrder(root.left)
           self.ans = min(self.ans, root.val - self.pre)
           self.pre = root.val
           if root.right:
               inOrder(root.right)
       inOrder(root)
        return self.ans

```

Solution for C++:

```
class Solution {
public:
    int ans = INT_MAX, pre = -1;
    int minDiffInBST(TreeNode* root) {
        if (root->left)
            minDiffInBST(root->left);
        if (pre >= 0)
            ans = min(ans, root->val - pre);
        pre = root->val;
        if (root->right)
            minDiffInBST(root->right);
        return ans;
    }
};
```