

# 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 :**

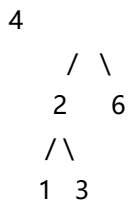
**Input:** root = [4,2,6,1,3,null,null]

**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:



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:

```
1 # Definition for a binary tree node.
2 # class TreeNode:
3 #     def __init__(self, x):
4 #         self.left = None
5 #         self.right = None
6 class Solution:
7     def minDiffInBST(self, root):
8         """
9         :type root: TreeNode
10        :rtype: int
11        """
12        self.ans, self.pre = float('inf'), -float('inf')
13        def inOrder(root):
14            if root.left:
```

```

15         inOrder(root.left)
16         self.ans = min(self.ans, root.val - self.pre)
17         self.pre = root.val
18         if root.right:
19             inOrder(root.right)
20     inOrder(root)
21     return self.ans
22

```

## Solution for C++:

```

1  class Solution {
2  public:
3      int ans = INT_MAX, pre = -1;
4      int minDiffInBST(TreeNode* root) {
5          if (root->left)
6              minDiffInBST(root->left);
7          if (pre >= 0)
8              ans = min(ans, root->val - pre);
9          pre = root->val;
10         if (root->right)
11             minDiffInBST(root->right);
12         return ans;
13     }
14 };

```