

# Problem 783: Minimum Distance Between BST Nodes

## Problem Information

**Difficulty:** Easy

**Acceptance Rate:** 60.82%

**Paid Only:** No

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

## Problem Description

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

**Example 1:**



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

**Example 2:**



**Input:** root = [1,0,48,null,null,12,49] **Output:** 1

**Constraints:**

\* The number of nodes in the tree is in the range `[2, 100]`. \* `0 <= Node.val <= 105`

**Note:** This question is the same as 530:

<<https://leetcode.com/problems/minimum-absolute-difference-in-bst/>>

## 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 minDiffInBST(TreeNode* root) {

    }
};
```

### 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 minDiffInBST(TreeNode root) {
```

```
    }  
}
```

### 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 minDiffInBST(self, root: Optional[TreeNode]) -> int:
```