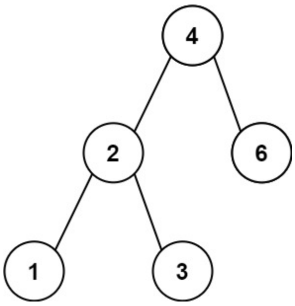
530. Minimum Absolute Difference in BST

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

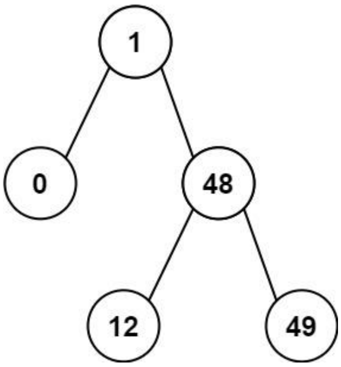
Example 1:



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, 10^4].
- 0 <= Node.val <= 10^5

```
# 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 getMinimumDifference(self, root):
        def traverse(node,low,high):
            if not node:
                return high-low
            left = traverse(node.left, low ,node.val)
            right = traverse(node.right, node.val, high)
            return min(left,right)

        return traverse(root, float('-inf'),float('inf'))
```