PROBLEM

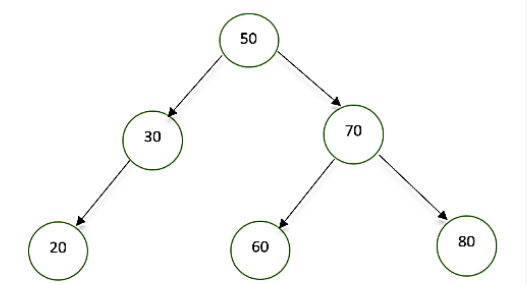
Minimum Absolute Difference In BST

Medium Accuracy: 56.22% Submissions: 17K+ Points: 4

Given a binary search tree having **n (n>1)** nodes, the task is to find the minimum absolute difference between any two nodes.

Example 1:

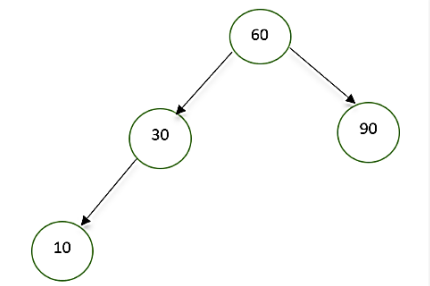
Input:
Input tree



Output:
10

Example 2:

Input:
Input tree



Output:
20

Explanation:
There are no two nodes whose absolute difference is smaller than 20.

Your Task:
You don't have to take any input. Just complete the function **absolute_diff() ,** that takes root as input and return minimum absolute difference between any two nodes.

Expected Time Complexity: O(n)
Expected Auxiliary Space: O(Height of tree)

Constraints:
$2 <= n <= 10^5$
$1 <= Node->data <= 10^9$

CODE

```
# class Node:
#     def _init_(self):
#         self.data = None
#         self.left = None
#         self.right = None

class Solution:
    def absolute_diff(self,root):
        self.prev = None
        self.min_diff = float('inf')
```

```
        def inorder(node):
            if node is None:
                return
            inorder(node.left)
            if self.prev is not None:
                self.min_diff = min(self.min_diff,
node.data - self.prev.data)
            self.prev = node
            inorder(node.right)

        inorder(root)
        return self.min_diff
```