Problem 1026: Maximum Difference Between Node and Ancestor

Problem Information

Difficulty: Medium
Acceptance Rate: 78.09%
Paid Only: No
Tags: Tree, Depth-First Search, Binary Tree

Problem Description

Given the `root` of a binary tree, find the maximum value `v` for which there exist **different** nodes `a` and `b` where `v = |a.val - b.val|` and `a` is an ancestor of `b`.

A node `a` is an ancestor of `b` if either: any child of `a` is equal to `b` or any child of `a` is an ancestor of `b`.

Example 1:

Input: root = [8,3,10,1,6,null,14,null,null,4,7,13] **Output:** 7 **Explanation:** We have various ancestor-node differences, some of which are given below : |8 - 3| = 5 |3 - 7| = 4 |8 - 1| = 7 |10 - 13| = 3 Among all possible differences, the maximum value of 7 is obtained by |8 - 1| = 7.

Example 2:

Input: root = [1,null,2,null,0,3] **Output:** 3

Constraints:

* The number of nodes in the tree is in the range `[2, 5000]`. * `0 <= Node.val <= 105`

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