

Problem 2385: Amount of Time for Binary Tree to Be Infected

Problem Information

Difficulty: Medium

Acceptance Rate: 64.77%

Paid Only: No

Tags: Hash Table, Tree, Depth-First Search, Breadth-First Search, Binary Tree

Problem Description

You are given the `root` of a binary tree with **unique** values, and an integer `start`. At minute `0`, an **infection** starts from the node with value `start`.

Each minute, a node becomes infected if:

- * The node is currently uninfected.
- * The node is adjacent to an infected node.

Return _the number of minutes needed for the entire tree to be infected._

Example 1:

Input: root = [1,5,3,null,4,10,6,9,2], start = 3 **Output:** 4 **Explanation:** The following nodes are infected during:
- Minute 0: Node 3
- Minute 1: Nodes 1, 10 and 6
- Minute 2: Node 5
- Minute 3: Node 4
- Minute 4: Nodes 9 and 2
It takes 4 minutes for the whole tree to be infected so we return 4.

Example 2:

Input: root = [1], start = 1 **Output:** 0 **Explanation:** At minute 0, the only node in the tree is infected so we return 0.

****Constraints:****

- * The number of nodes in the tree is in the range `[1, 105]`.
- * `1 <= Node.val <= 105`
- * Each node has a **unique** value.
- * A node with a value of `start` exists in the tree.

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 amountOfTime(TreeNode* root, int start) {
    }
};
```

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 amountOfTime(TreeNode root, int start) {

}
}
```

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