Problem 2096: Step-By-Step Directions From a Binary Tree Node to Another

Problem Information

Difficulty: Medium
Acceptance Rate: 56.37%
Paid Only: No
Tags: String, Tree, Depth-First Search, Binary Tree

Problem Description

You are given the `root` of a **binary tree** with `n` nodes. Each node is uniquely assigned a value from `1` to `n`. You are also given an integer `startValue` representing the value of the start node `s`, and a different integer `destValue` representing the value of the destination node `t`.

Find the **shortest path** starting from node `s` and ending at node `t`. Generate step-by-step directions of such path as a string consisting of only the **uppercase** letters `'L'`, `'R'`, and `'U'`. Each letter indicates a specific direction:

* `'L'` means to go from a node to its **left child** node. * `'R'` means to go from a node to its **right child** node. * `'U'` means to go from a node to its **parent** node.

Return _the step-by-step directions of the**shortest path** from node _`s` _to node_ `t`.

Example 1:

![](https://assets.leetcode.com/uploads/2021/11/15/eg1.png)

Input: root = [5,1,2,3,null,6,4], startValue = 3, destValue = 6 **Output:** "UURL" **Explanation:** The shortest path is: 3 -> 1 -> 5 -> 2 -> 6.

Example 2:

![](https://assets.leetcode.com/uploads/2021/11/15/eg2.png)

Input: root = [2,1], startValue = 2, destValue = 1 **Output:** "L" **Explanation:** The shortest path is: 2 -> 1.

Constraints:

* The number of nodes in the tree is `n`. * `2 <= n <= 105` * `1 <= Node.val <= n` * All the values in the tree are **unique**. * `1 <= startValue, destValue <= n` * `startValue != destValue`

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:
string getDirections(TreeNode* root, int startValue, int destValue) {

}
};
```

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 String getDirections(TreeNode root, int startValue, int destValue) {

}
}
```

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 getDirections(self, root: Optional[TreeNode], startValue: int, destValue:
int) -> str:
```