# Problem 1430: Check If a String Is a Valid Sequence from Root to Leaves Path in a Binary Tree

## Problem Information

**Difficulty:** Medium
**Acceptance Rate:** 47.40%
**Paid Only:** Yes
**Tags:** Tree, Depth-First Search, Breadth-First Search, Binary Tree

## Problem Description

Given a binary tree where each path going from the root to any leaf form a **valid sequence** , check if a given string is a **valid sequence** in such binary tree.

We get the given string from the concatenation of an array of integers `arr` and the concatenation of all values of the nodes along a path results in a **sequence** in the given binary tree.

**Example 1:**

**![](https://assets.leetcode.com/uploads/2019/12/18/leetcode_testcase_1.png)**

**Input:** root = [0,1,0,0,1,0,null,null,1,0,0], arr = [0,1,0,1] **Output:** true **Explanation:** The path 0 -> 1 -> 0 -> 1 is a valid sequence (green color in the figure). Other valid sequences are: 0 -> 1 -> 1 -> 0 0 -> 0 -> 0

**Example 2:**

**![](https://assets.leetcode.com/uploads/2019/12/18/leetcode_testcase_2.png)**

**Input:** root = [0,1,0,0,1,0,null,null,1,0,0], arr = [0,0,1] **Output:** false **Explanation:** The path 0 -> 0 -> 1 does not exist, therefore it is not even a sequence.

**Example 3:**

**![](https://assets.leetcode.com/uploads/2019/12/18/leetcode_testcase_3.png)**

**Input:** root = [0,1,0,0,1,0,null,null,1,0,0], arr = [0,1,1] **Output:** false **Explanation:** The path 0 -> 1 -> 1 is a sequence, but it is not a valid sequence.

**Constraints:**

* `1 <= arr.length <= 5000` * `0 <= arr[i] <= 9` * Each node's value is between [0 - 9].

## 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:
bool isValidSequence(TreeNode* root, vector<int>& arr) {

}
};
```

**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 boolean isValidSequence(TreeNode root, int[] arr) {

}
}
```

### 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 isValidSequence(self, root: Optional[TreeNode], arr: List[int]) -> bool:
```