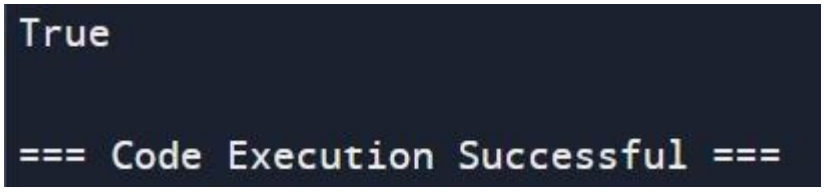
5. Check If a String Is a Valid Sequence from Root to Leaves Path in a Binary Tree Given a binary tree where each path going from the root to any leaf form a valid sequence, check if a given string is a valid sequence in such binary tree. We get the given string from the concatenation of an array of 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: 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

PROGRAM:- from typing import List

```
# 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: TreeNode, arr: List[int]) -> bool:
        def dfs(node, arr, index):
            if not node:
                return False
            if index >= len(arr) or node.val != arr[index]:
                return False
            if index == len(arr) - 1:
                return not node.left and not node.right
            return dfs(node.left, arr, index + 1) or dfs(node.right, arr, index + 1)

        if not root:
            return len(arr) == 0
        return dfs(root, arr, 0)

# Example usage
root = TreeNode(0)
root.left = TreeNode(1)
root.right = TreeNode(0)
root.left.left = TreeNode(0)
root.left.right = TreeNode(1)
root.right.left = TreeNode(0)
root.left.left.right = TreeNode(1)
root.left.right.left = TreeNode(0)
root.left.right.right = TreeNode(0)

arr = [0, 1, 0, 1]

sol = Solution()
print(sol.isValidSequence(root, arr))  # Output: True
```

OUTPUT:-



TIME COMPLEXITY:-O(n)