Path With Given Sequence (medium)

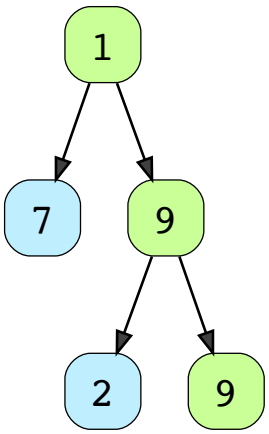
We'll cover the following

Problem Statement

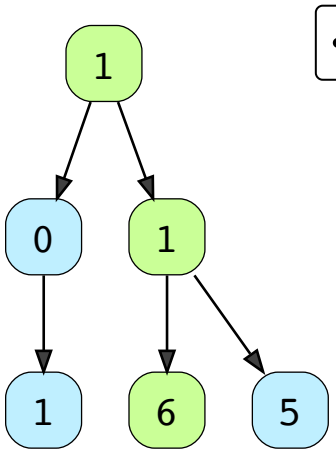
Given a binary tree and a number sequence, find if the sequence is present as a root-to-leaf path in the given tree.

Example 1:

Sequence: [1, 9, 9]
Output: true
Explanation: The tree has a path 1 -> 9 -> 9.



Example 2:



Sequence: [1, 0, 7]
Output: false
Explanation: The tree does not have a path 1 -> 0 -> 7.

Sequence: [1, 1, 6]
Output: true
Explanation: The tree has a path 1 -> 1 -> 6.

Try it yourself

Try solving this question here:

Java | Python3 | JS | C++

```
class TreeNode:
  def __init__(self, val, left=None, right=None)
    self.val = val
    self.left = left
    self.right = right


def find_path(root, sequence):
  # TODO: Write your code here
  if not root:
    return len(sequence)==0
  return find_path_recursive(root,0,sequence)

def find_path_recursive(node,index,sequence):
  if not node:
    return False
  seqLen = len(sequence)
  if index >= seqLen or node.val != sequence[inc
    return False
  if not node.left and not node.right and index
    return True
  return find_path_recursive(node.left,index+1,s
def main():

```

```
    root = TreeNode(1)
    root.left = TreeNode(0)
    root.right = TreeNode(1)
    root.left.left = TreeNode(1)
```

```
Output                                                    0.14s

 Tree has path sequence: False
 Tree has path sequence: True
```

Solution

This problem follows the Binary Tree Path Sum (https://www.educative.io/collection/page/5668639101419520/5671464854355968/5642684278505472/) pattern. We can follow the same **DFS** approach and additionally, track the element of the given sequence that we should match with the current node. Also, we can return `false` as soon as we find a mismatch between the sequence and the node value.

Code

Here is what our algorithm will look like:

Java | Python3 | C++ | JS

```
class TreeNode:
  def __init__(self, val, left=None, right=None)
    self.val = val
    self.left = left
    self.right = right


def find_path(root, sequence):
  if not root:
```

```
    return len(sequence) == 0

 ·return·find_path_recursive(root,·sequence,·0)


def find_path_recursive(currentNode, sequence, s

  if currentNode is None:
    return False

  seqLen = len(sequence)
  if sequenceIndex >= seqLen or currentNode.val
    return False

  # if the current node is a leaf, add it is the
  if currentNode.left is None and currentNode.ri
    return True

  # recursively call to traverse the left and ri
```

```
Output                                                    0.13s

 Tree has path sequence: False
 Tree has path sequence: True
```

Time complexity

The time complexity of the above algorithm is $O(N)$, where 'N' is the total number of nodes in the tree. This is due to the fact that we traverse each node once.

Space complexity

The space complexity of the above algorithm will be $O(N)$ in the worst case. This space will be used to store the recursion stack. The worst case will happen when the given tree is a linked list (i.e., every node has only one

child).


Back

Next

Sum of Path Numbers (medium)

Count Paths for a Sum (medium)

Mark as Completed

Report an Issue

Ask a Question (https://discuss.educative.io/tag/path-with-given-sequence-medium__pattern-tree-depth-first-search__grokking-the-coding-interview-patterns-for-coding-questions)