

[Previous \(/articles/baseball-game/\)](/articles/baseball-game/)   Next [\(/articles/knight-probability-in-chessboard/\)](/articles/knight-probability-in-chessboard/)

## 687. Longest Univalve Path [↗ \(/problems/longest-univalve-path/\)](/problems/longest-univalve-path/)

Sept. 30, 2017

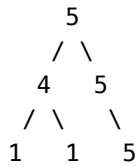
Average Rating: 4.67 (3 votes)

Given a binary tree, find the length of the longest path where each node in the path has the same value. This path may or may not pass through the root.

**Note:** The length of path between two nodes is represented by the number of edges between them.

### Example 1:

Input:

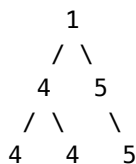


Output:

2

### Example 2:

Input:



Output:

**Note:** The given binary tree has not more than 10000 nodes. The height of the tree is not more than 1000.

---

Approach #1: Recursion [Accepted]

## Approach #1: Recursion [Accepted]

### Intuition

We can think of any path (of nodes with the same values) as up to two arrows extending from it's root.

Specifically, the *root* of a path will be the unique node such that the parent of that node does not appear in the path, and an *arrow* will be a path where the root only has one child node in the path.

Then, for each node, we want to know what is the longest possible arrow extending left, and the longest possible arrow extending right? We can solve this using recursion.

### Algorithm

Let `arrow_length(node)` be the length of the longest arrow that extends from the `node`. That will be `1 + arrow_length(node.left)` if `node.left` exists and has the same value as `node`. Similarly for the `node.right` case.

While we are computing arrow lengths, each candidate answer will be the sum of the arrows in both directions from that node. We record these candidate answers and return the best one.

### Python

```
class Solution(object):
    def longestUnivaluePath(self, root):
        self.ans = 0
```

Articles > 687. Longest Univalue Pa

```
    def arrow_length(node):
        if not node: return 0
        left_length = arrow_length(node.left)
        right_length = arrow_length(node.right)
        left_arrow = right_arrow = 0
        if node.left and node.left.val == node.val:
            left_arrow = left_length + 1
        if node.right and node.right.val == node.val:
            right_arrow = right_length + 1
        self.ans = max(self.ans, left_arrow + right_arrow)
        return max(left_arrow, right_arrow)

    arrow_length(root)
    return self.ans
```

## Java

```
class Solution {
    int ans;
    public int longestUnivaluePath(TreeNode root) {
        ans = 0;
        arrowLength(root);
        return ans;
    }
    public int arrowLength(TreeNode node) {
        if (node == null) return 0;
        int left = arrowLength(node.left)
        int right = arrowLength(node.right);
        int arrowLeft = 0, arrowRight = 0;
        if (node.left != null && node.left.val == node.val) {
            arrowLeft += left + 1;
        }
        if (node.right != null && node.right.val == node.val) {
            arrowRight += right + 1;
        }
        ans = Math.max(ans, arrowLeft + arrowRight);
        return Math.max(arrowLeft, arrowRight);
    }
}
```

## Complexity Analysis

- Time Complexity:  $O(N)$ , where  $N$  is the number of nodes in the tree. We process every node once.
- Space Complexity:  $O(H)$ , where  $H$  is the height of the tree. Our recursive call stack could be

up to  $H$  layers deep.

Articles > 687. Longest Univalued Path

Previous (/articles/baseball-game/)

Next (/articles/knight-probability-in-chessboard/)



Join the conversation

Login to Reply

J

**jkarimi** commented 3 days ago

I do not understand the arrow concept:  
(<https://discuss.leetcode.com/user/jkarimi>)

We can think of any path (of nodes with the same values) as up to two arrows extending from it's root.

could you elaborate?

A


**amirdon** commented 5 days ago

@GuptaCoder007 (<https://discuss.leetcode.com/uid/317567>) I've had the same issue and what I've decided to do is to take some online courses on algorithms & data structures. There are many concepts that's best learned before diving into these algorithmic type problems. Lookup some courses on Udemy/Coursera.

G

**GuptaCoder007** commented 5 days ago

so I started with leetcode 2-3 weeks ago. My efficiency of solving these problems quite low . either I prepare a code with high complexity or I am not able to completely solve the problem or even not able to touch it. I find it easier after looking into the solution, but not able to crack the solution beforehand. efficient way to prepare for coding interviews ?

**terrymyy** commented last week Articles > 687. Longest Univalued Path

---

`arrowLeft += left + 1;` could simply be `arrowLeft = left + 1;`  
(<https://discuss.leetcode.com/user/terrymyy>)

[View original thread \(https://discuss.leetcode.com/topic/105581\)](https://discuss.leetcode.com/topic/105581)

