

// <https://leetcode.com/problems/diameter-of-binary-tree/>

LeetCode

ExploreProblemsMockContestDiscussStore

September LeetCode Challenge

DescriptionSolutionDiscuss (999+)SubmissionsC#

543. Diameter of Binary Tree

Easy 3489 211 Add to List Share

Given a binary tree, you need to compute the length of the diameter of the tree. The diameter of a binary tree is the length of the **longest** path between any two nodes in a tree. This path may or may not pass through the root.

Example:
Given a binary tree

```
    1
   /\
  2  3
 /\
4  5
```

Return 3, which is the length of the path [4,2,1,3] or [5,2,1,3].

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

Accepted 374,189 Submissions 769,707

Seen this question in a real interview before?

Companies

Related Topics

Similar Questions

```
1  /**
2   * Definition for a binary tree node.
3   * public class TreeNode {
4   *     public int val;
5   *     public TreeNode left;
6   *     public TreeNode right;
7   *     public TreeNode(int val=0, TreeNode left=null, TreeNode right=null) {
8   *         this.val = val;
9   *         this.left = left;
10 *         this.right = right;
11 *     }
12 * }
13 */
14 public class Solution {
15     public int DiameterOfBinaryTree(TreeNode root) {
16         if(root==null) return 0;
17         int longest=0;
18         DiameterOfTree(root, ref longest);
19         return longest;
20     }
21     public int DiameterOfTree(TreeNode root, ref int longestPath)
22     {
23         if(root==null) return 0;
24
25         int leftTreeDiameter=0,rtTreeDiameter=0;
26         int leftLength = DiameterOfTree(root.left,ref leftTreeDiameter);
27         int rtLength = DiameterOfTree(root.right,ref rtTreeDiameter);
28         longestPath = Math.Max(Math.Max(leftLength+rtLength,leftTreeDiameter),rtTreeDiameter);
29         return Math.Max(leftLength,rtLength)+1;
30     }
31 }
```

Your previous code was restored from your local storage. [Reset to default](#)

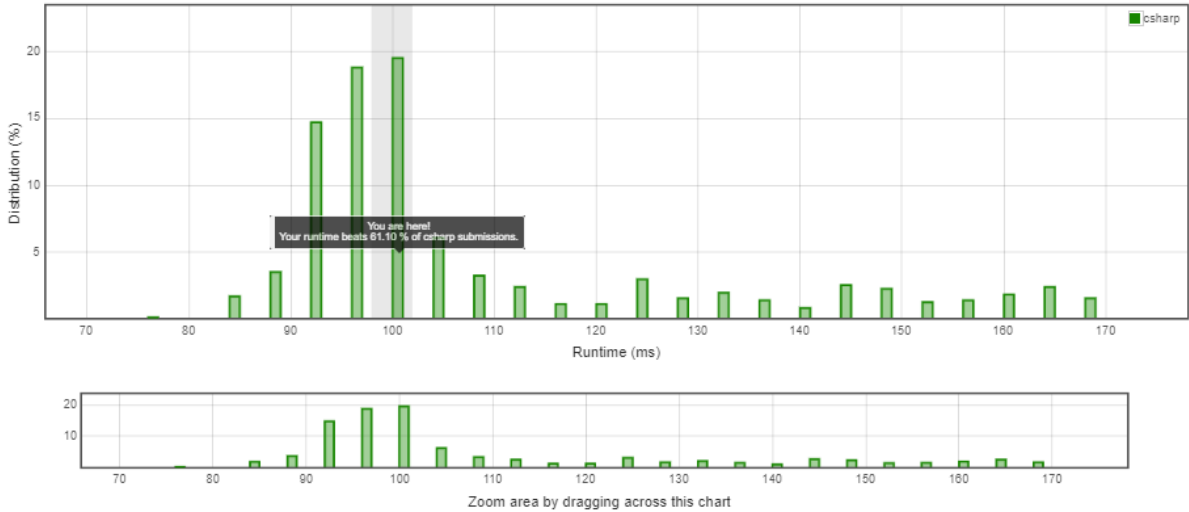
```
public class Solution {
    public int DiameterOfBinaryTree(TreeNode root) {
        if(root==null) return 0;
        int longest=0;
        DiameterOfTree(root, ref longest);
        return longest;
    }
    public int DiameterOfTree(TreeNode root, ref int longestPath)
    {
        if(root==null) return 0;

        int leftTreeDiameter=0,rtTreeDiameter=0;
        int leftLength = DiameterOfTree(root.left,ref leftTreeDiameter);
        int rtLength = DiameterOfTree(root.right,ref rtTreeDiameter);
        longestPath = Math.Max(Math.Max(leftLength+rtLength,leftTreeDiameter),rtTreeDiameter);
        return Math.Max(leftLength,rtLength)+1;
    }
}
```

Submission Detail

106 / 106 test cases passed.	Status: Accepted
Runtime: 100 ms	Submitted: 0 minutes ago
Memory Usage: 25.6 MB	

Accepted Solutions Runtime Distribution



Accepted Solutions Memory Distribution

