

1522. Diameter of N-Ary Tree

Medium

👍 315🗨 5

🤍 Add to List

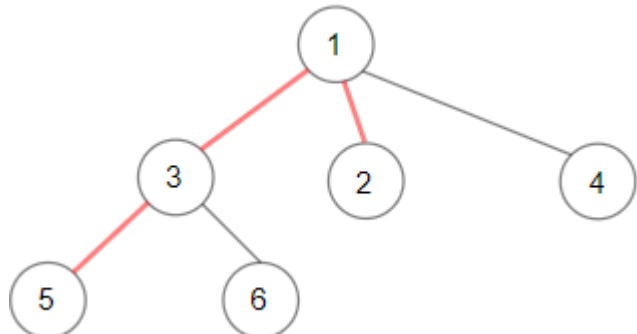
🔗 Share

Given a `root` of an N-ary tree, you need to compute the length of the diameter of the tree.

The diameter of an N-ary tree is the length of the **longest** path between any two nodes in the tree. This path may or may not pass through the root.

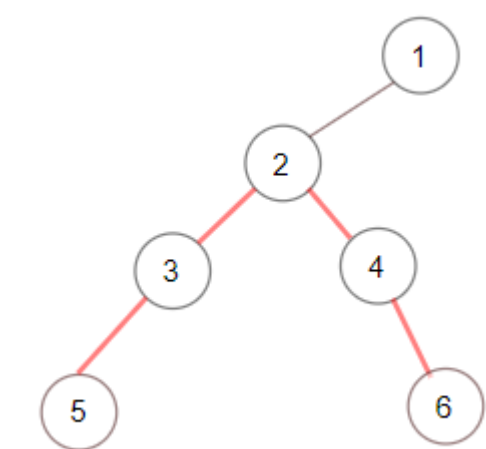
(N-ary-Tree input serialization is represented in its level order traversal, each group of children is separated by the null value.)

Example 1:



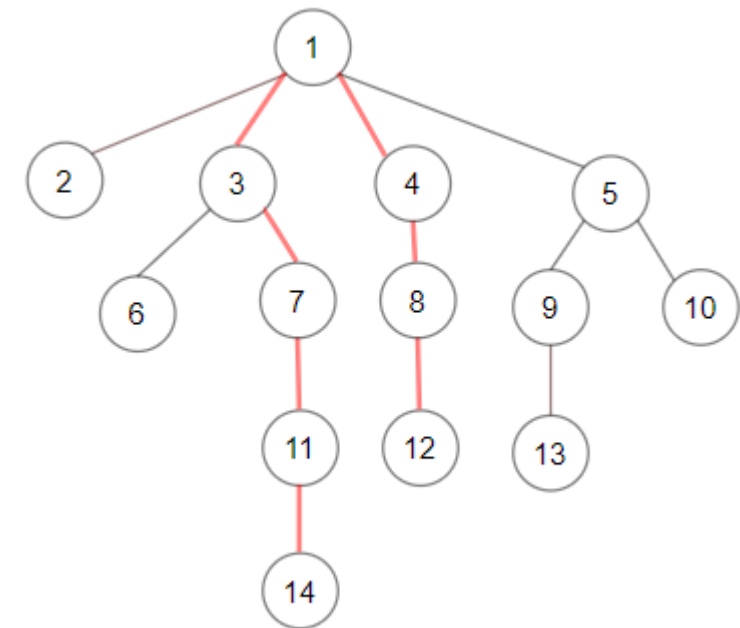
Input: root = [1,null,3,2,4,null,5,6]
Output: 3
Explanation: Diameter is shown in red color.

Example 2:



Input: root = [1,null,2,null,3,4,null,5,null,6]
Output: 4

Example 3:



Input: root = [1,null,2,3,4,5,null,null,6,7,null,8,null,9,10,null,null,11,null,12,null,13,null,null,14]
Output: 7

Constraints:

- The depth of the n-ary tree is less than or equal to 1000.
- The total number of nodes is between $[1, 10^4]$.

Accepted 20,564 | Submissions 28,505

Seen this question in a real interview before?

☒ Yes ☐ No

Companies 🏢 /

0 ~ 6 months 6 months ~ 1 year 1 year ~ 2 years

Facebook | 8

Related Topics

Tree Depth-First Search

Similar Questions

Diameter of Binary Tree

Easy

Hide Hint 1

For the node i , calculate the height of each of its children and keep the first and second maximum heights ($max1_i, max2_i$).

Hide Hint 2

Check all nodes and return $\max(2 + max1_i + max2_i)$.

```
1 *
2 *
3 *
4 *
5
6
7
8
9
10
11
12 *
13 *
14
15
16
17
18
19
20
21
22 *
23
24 *
25
26
27 *
28
29
30
31
32
33
34
35
class Solution {
    public int diameter(Node root) {
        if (root == null || root.children.size() == 0) {
            return 0;
        }

        int[] maxDiameter = new int[1];
        diameterHelper(root, maxDiameter);
        return maxDiameter[0];
    }

    private int diameterHelper(Node root, int[] maxDiameter) {
        if (root.children.size() == 0) {
            return 0;
        }

        // Setting below maximums to -1 helps in the case if there is only one child
        // node of this root node.
        int maxHeight1 = -1;
        int maxHeight2 = -1;

        for (Node child : root.children) {
            int childHeight = diameterHelper(child, maxDiameter);
            if (childHeight > maxHeight1) {
                maxHeight2 = maxHeight1;
                maxHeight1 = childHeight;
            } else if (childHeight > maxHeight2) {
                maxHeight2 = childHeight;
            }
        }

        maxDiameter[0] = Math.max(maxDiameter[0], maxHeight1 + maxHeight2 + 2);
        return maxHeight1 + 1;
    }
}
```