Problem 1522: Diameter of N-Ary Tree

Problem Information

Difficulty: Medium
Acceptance Rate: 75.38%
Paid Only: Yes
Tags: Tree, Depth-First Search

Problem Description

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.

(_Nary-Tree input serialization is represented in their 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, 104]`.

Code Snippets

C++:

```
/*
// Definition for a Node.
class Node {
public:
int val;
vector<Node*> children;

Node() {}

Node(int _val) {
val = _val;
}

Node(int _val, vector<Node*> _children) {
val = _val;
children = _children;
}
};
*/

class Solution {
public:
int diameter(Node* root) {

}
};
```

Java:

```
/*
// Definition for a Node.
class Node {
public int val;
public List<Node> children;


public Node() {
children = new ArrayList<Node>();
}

public Node(int _val) {
val = _val;
children = new ArrayList<Node>();
}

public Node(int _val,ArrayList<Node> _children) {
val = _val;
children = _children;
}
};
*/

class Solution {
public int diameter(Node root) {

}
}
```

Python3:

```
"""
# Definition for a Node.
class Node:
def __init__(self, val: Optional[int] = None, children:
Optional[List['Node']] = None):
self.val = val
self.children = children if children is not None else []
"""

class Solution:
def diameter(self, root: 'Node') -> int:
"""
```

```
:type root: 'Node'
:rtype: int
"""
```