

DescriptionSolutionDiscuss (122)Submissions

### 1506. Find Root of N-Ary Tree

Medium

👍 287🗨 99

🔖 Add to List

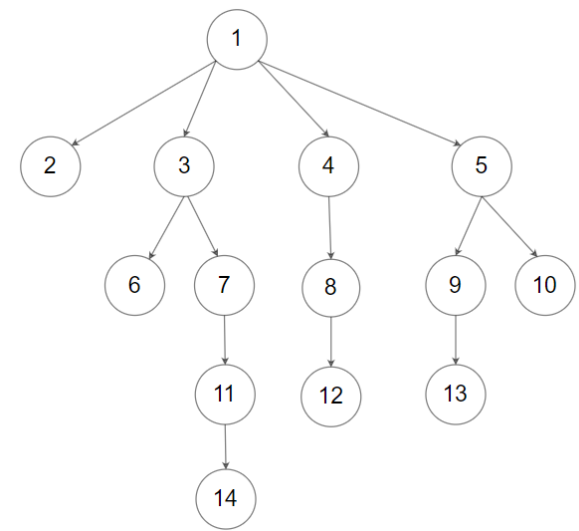
🔗 Share

You are given all the nodes of an *N*-ary tree as an array of `Node` objects, where each node has a **unique value**.

Return the *root* of the *N*-ary tree.

#### Custom testing:

An *N*-ary tree can be serialized as represented in its level order traversal where each group of children is separated by the `null` value (see examples).

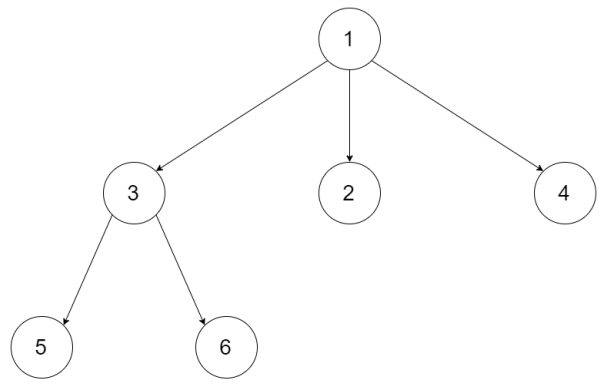


For example, the above tree is serialized as `[1,null,2,3,4,5,null,null,6,7,null,8,null,9,10,null,null,11,null,12,null,13,null,null,14]`

The testing will be done in the following way:

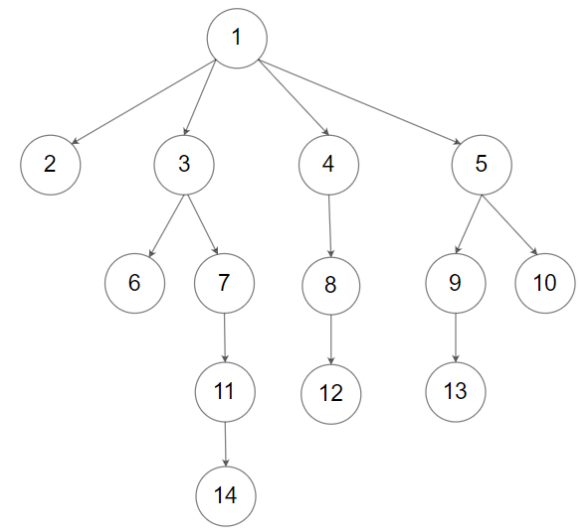
1. The **input data** should be provided as a serialization of the tree.
2. The driver code will construct the tree from the serialized input data and put each `Node` object into an array **in an arbitrary order**.
3. The driver code will pass the array to `findRoot`, and your function should find and return the root `Node` object in the array.
4. The driver code will take the returned `Node` object and serialize it. If the serialized value and the input data are the **same**, the test **passes**.

#### Example 1:



**Input:** `tree = [1,null,3,2,4,null,5,6]`  
**Output:** `[1,null,3,2,4,null,5,6]`  
**Explanation:** The tree from the input data is shown above. The driver code creates the tree and gives `findRoot` the `Node` objects in an arbitrary order. For example, the passed array could be `[Node(5),Node(4),Node(3),Node(6),Node(2),Node(1)]` or `[Node(2),Node(6),Node(1),Node(3),Node(5),Node(4)]`. The `findRoot` function should return the root `Node(1)`, and the driver code will serialize it and compare with the input data. The input data and serialized `Node(1)` are the same, so the test passes.

#### Example 2:



**Input:** `tree = [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:** `[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]`

#### Constraints:

- The total number of nodes is between `[1, 5 * 104]`.
- Each node has a **unique** value.

#### Follow up:

- Could you solve this problem in constant space complexity with a linear time algorithm?

Accepted 18,302 | Submissions 23,322

Seen this question in a real interview before?

YesNo

Companies 🏢

0 ~ 6 months6 months ~ 1 year1 year ~ 2 years

Google

Related Topics

Hash TableBit ManipulationTreeDepth-First Search

Similar Questions

Move Sub-Tree of N-Ary Tree

Hard

Hide Hint 1

Node with indegree 0 is the root

JavaAutocomplete

```
1 class Solution {
2     public Node findRoot(List<Node> tree) {
3
4
5
6         Integer valueSum = 0;
7
8         for (Node node : tree) {
9             // the value is added as a parent node
10            valueSum += node.val;
11            for (Node child : node.children)
12                // the value is deducted as a child node.
13                valueSum -= child.val;
14        }
15
16        Node root = null;
17        // the value of the root node is `valueSum`
18        for (Node node : tree) {
19            if (node.val == valueSum) {
20                root = node;
21                break;
22            }
23        }
24        return root;
25    }
26 }
```