# 1506. Find Root of N-Ary Tree Premium

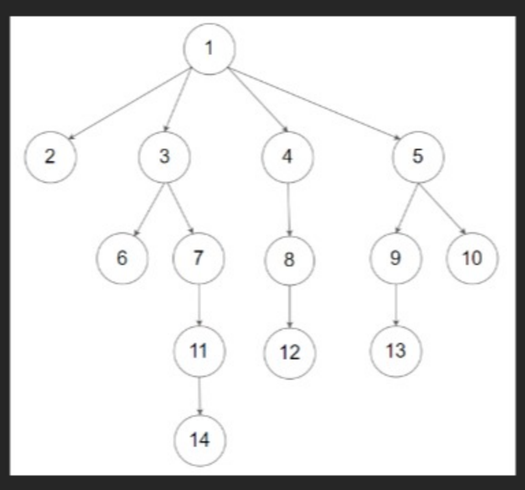
Medium | Topics | Companies | Hint

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).

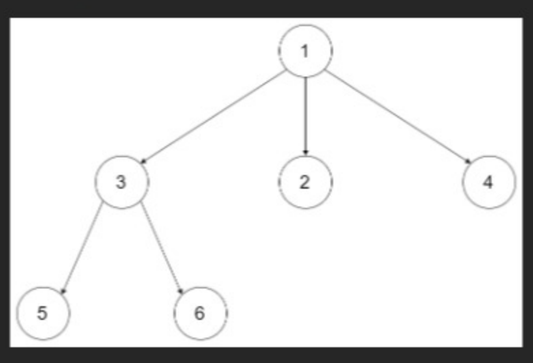



For example, the above tree is serialized as `[1,null,2,3,4,5,null,null,6,7,null,8,null,9,10,null,null,11,null,12,null,13,null,null,14]`.

The testing will be done in the following way:

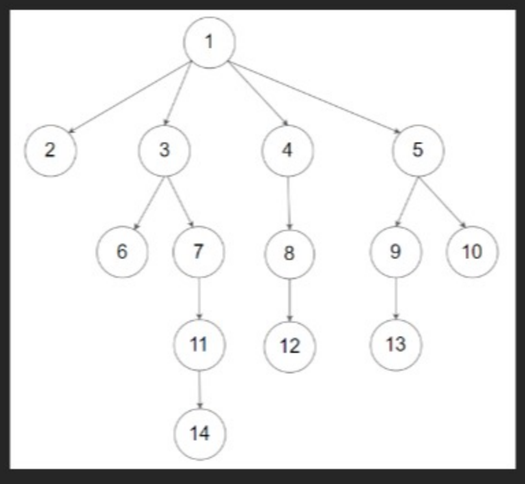
1. The **input data** should be provided as a serialization of the tree.
2. The driver code will construct the tree from the serialized input data and put each `Node` object into an array **in an arbitrary order**.
3. The driver code will pass the array to `findRoot`, and your function should find and 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 $[1, 5 * 10^4]$.
- Each node has a **unique** value.

**Follow up:**

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

Seen this question in a real interview before? 1/5

Yes No

Accepted 36.2K | Submissions 46.2K | Acceptance Rate 78.4%

Topics

Hash Table | Bit Manipulation | Tree | Depth-First Search

Companies

0 - 6 months

Google 2

Hint 1

Node with indegree 0 is the root

Similar Questions

Move Sub-Tree of N-Ary Tree Hard

Discussion (6)