LeetCode Explore Problems Interview New Contest Discuss Store

LeetCoding Challenge + GIVEAWAY!

Description | Solution | Discuss (339) | Submissions

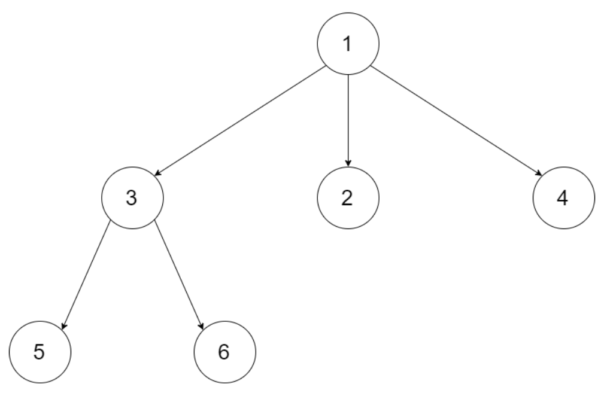
428. Serialize and Deserialize N-ary Tree

Hard 742 43 Add to List Share

Serialization is the process of converting a data structure or object into a sequence of bits so that it can be stored in a file or memory buffer, or transmitted across a network connection link to be reconstructed later in the same or another computer environment.

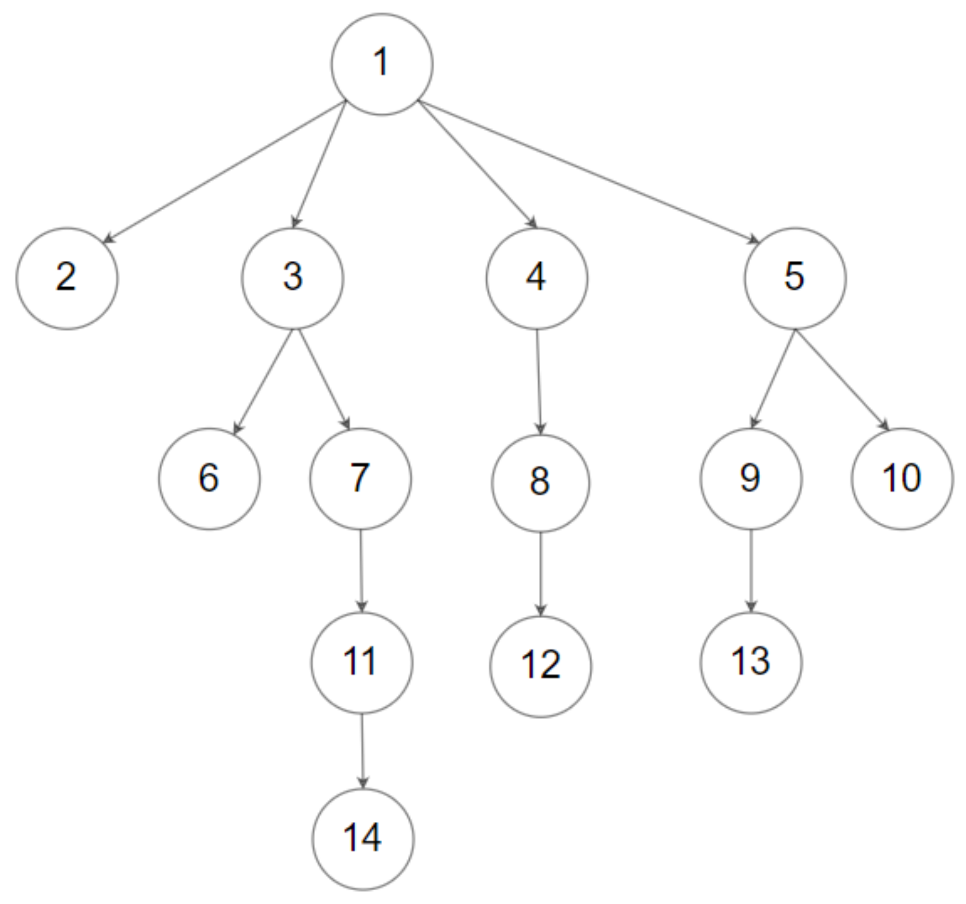
Design an algorithm to serialize and deserialize an N-ary tree. An N-ary tree is a rooted tree in which each node has no more than N children. There is no restriction on how your serialization/deserialization algorithm should work. You just need to ensure that an N-ary tree can be serialized to a string and this string can be deserialized to the original tree structure.

For example, you may serialize the following `3-ary` tree



as `[1 [3[5 6] 2 4]]`. Note that this is just an example, you do not necessarily need to follow this format.

Or you can follow LeetCode's level order traversal serialization format, where each group of children is separated by the null value.



For example, the above tree may be 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`

You do not necessarily need to follow the above-suggested formats, there are many more different formats that work so please be creative and come up with different approaches yourself.

Example 1:

```
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,nul

Output:
[1,null,2,3,4,5,null,null,6,7,null,8,null,9,10,null,null,11,null,12,null,13,null,nul
```

Example 2:

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

Example 3:

```
Input: root = []
Output: []
```

Constraints:

- The number of nodes in the tree is in the range `[0, 10^4]`.
- `0 <= Node.val <= 10^4`
- The height of the n-ary tree is less than or equal to `1000`
- Do not use class member/global/static variables to store states. Your encode and decode algorithms should be stateless.

Accepted 57,044 Submissions 89,800

Seen this question in a real interview before? Yes No

Companies

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

Apple 5	Microsoft 3	Zoom 3	Uber 2

Related Topics

String Tree Depth-First Search Breadth-First Search

Similar Questions

Serialize and Deserialize Binary Tree	Hard
Serialize and Deserialize BST	Medium
Encode N-ary Tree to Binary Tree	Hard

Java Autocomplete

```java
class Codec {

    class WrappableInt {
        private int value;
        public WrappableInt(int x) {
            this.value = x;
        }
        public int getValue() {
            return this.value;
        }
        public void increment() {
            this.value++;
        }
    }

    // Encodes a tree to a single string.
    public String serialize(Node root) {

        StringBuilder sb = new StringBuilder();
        this._serializeHelper(root, sb);
        return sb.toString();
    }

    private void _serializeHelper(Node root, StringBuilder sb) {

        if (root == null) {
            return;
        }

        // Add the value of the node
        sb.append((char) (root.val + '0'));

        // Add the number of children
        sb.append((char) (root.children.size() + '0'));

        // Recurse on the subtrees and build the
        // string accordingly
        for (Node child : root.children) {
            this._serializeHelper(child, sb);
        }
    }

    // Decodes your encoded data to tree.
    public Node deserialize(String data) {
        if(data.isEmpty())
            return null;

        return this._deserializeHelper(data, new WrappableInt(0));
    }

    private Node _deserializeHelper(String data, WrappableInt index) {

        if (index.getValue() == data.length()) {
            return null;
        }

        // The invariant here is that the "index" always
        // points to a node and the value next to it
        // represents the number of children it has.
        Node node = new Node(data.charAt(index.getValue()) - '0', new ArrayList<Node>());
        index.increment();
        int numChildren = data.charAt(index.getValue()) - '0';
        for (int i = 0; i < numChildren; i++) {
            index.increment();
            node.children.add(this._deserializeHelper(data, index));
        }

        return node;
    }
}
```

Problems Pick One Prev /99 Next Console Contribute Run Code Submit