

👤

yorkshire

★ 684

Last Edit: October 15, 2018 4:04 AM

1.0K VIEWS

24

The left child of a binary node is the subtree encoding all the children of the corresponding n-ary node.

The right child of a binary node is a chain of the binary root nodes encoding each sibling of the n-ary node.

Hence the root node has no right binary child, because the root has no siblings.

🔖

```
class Codec:

    def encode(self, root):
        if not root:
            return None

        binary = TreeNode(root.val)                # create a binary root
        if not root.children:
            return binary

        binary.left = self.encode(root.children[0]) # left child of binary is the encoding of all n-ary children,
        node = binary.left                         # starting with the first child.
        for child in root.children[1:]:            # other children of n-ary root are right child of previous child
            node.right = self.encode(child)
            node = node.right

        return binary

    def decode(self, data):
        if not data:
            return None

        nary = Node(data.val, [])                  # create n-ary root
        node = data.left                           # move to first child of n-ary root
        while node:                                 # while more children of n-ary root
            nary.children.append(self.decode(node)) # append to list
            node = node.right                       # and move to next child

        return nary
```

Type comment here... (Markdown is supported)

Post

🔌

feng-zhe

★ 77

September 16, 2018 6:50 AM

Nice and clear solution!

👍 0

🔍

🔍 Reply