590. N-ary Tree PostOrder Traversal

Easy

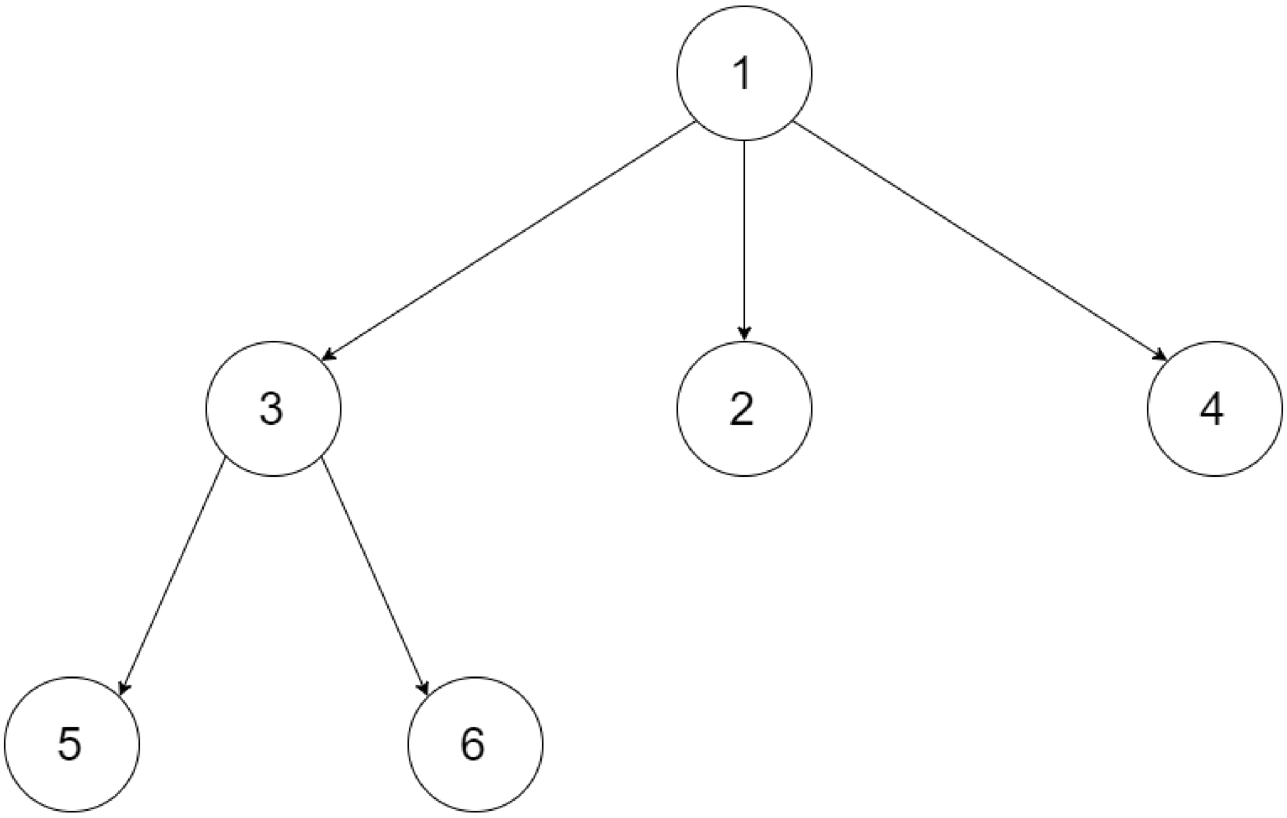
Topics

Companies

Given the `root` of an n-ary tree, return *the postorder traversal of its nodes' values*.

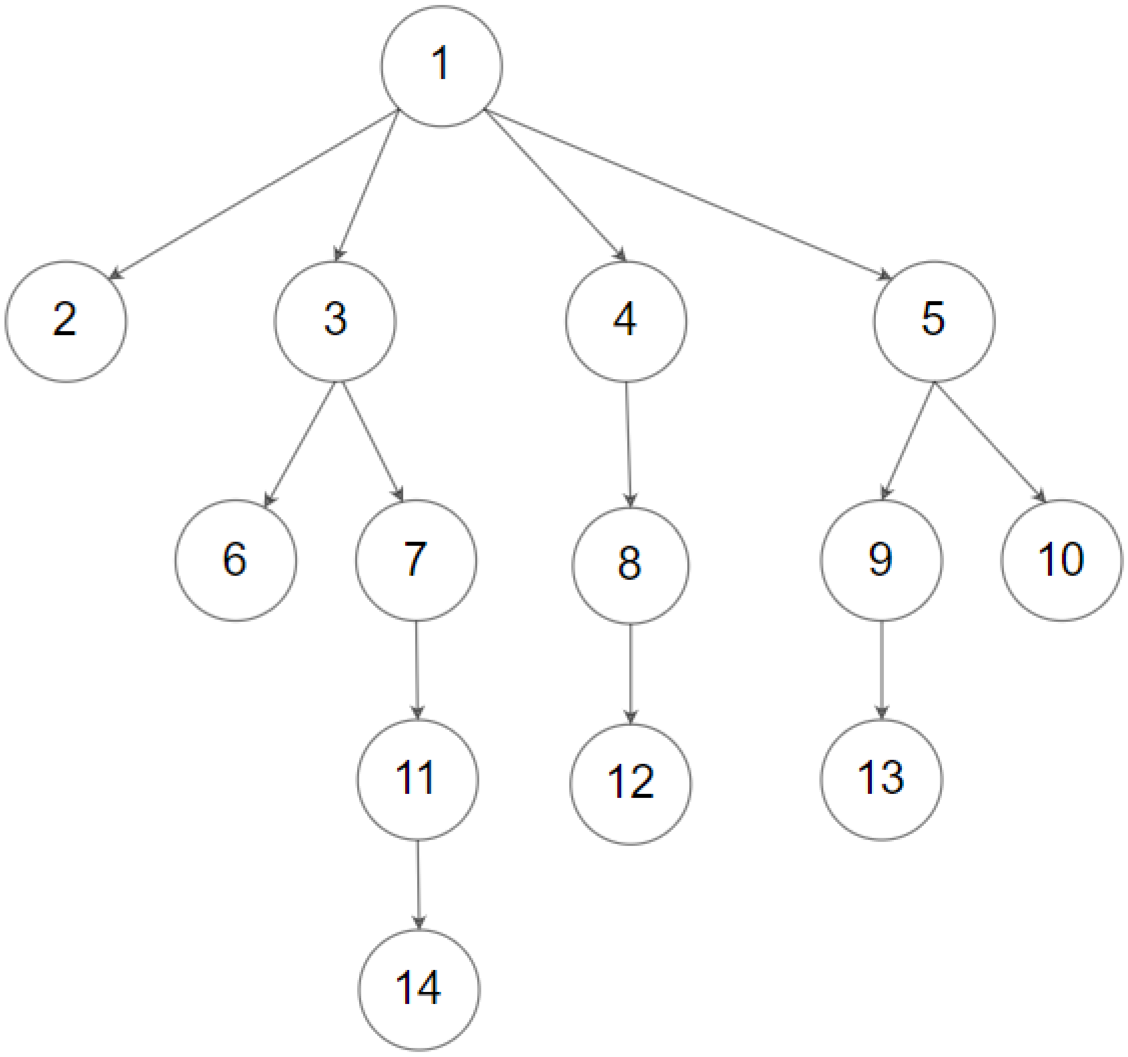
Nary-Tree input serialization is represented in their level order traversal. Each group of children is separated by the null value (See examples)

Example 1:



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

Example 2:



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: [2,6,14,11,7,3,12,8,4,13,9,10,5,1]

Constraints:

- The number of nodes in the tree is in the range `[0, 104]`.
- `0 <= Node.val <= 104`
- The height of the n-ary tree is less than or equal to `1000`.

Solution:

```
/*

// Definition for a Node.

class Node {

    public int val;

    public List<Node> children;

```

```java
    public Node() {}


    public Node(int _val) {

        val = _val;

    }


    public Node(int _val, List<Node> _children) {

        val = _val;

        children = _children;

    }

};

*/


class Solution {

    public List<Integer> postorder(Node root) {

        if(root == null){

            return new ArrayList<>();

        }


        List<Integer> res = new ArrayList<>();

        dfs(root, res);


        return res;

    }
    private void dfs(Node root, List<Integer> res){

        for(Node child: root.children){

            dfs(child, res);

        }



        res.add(root.val);

    }

}
```