Construct N-ary Tree & Level Order Traversal

Given an n-ary tree, return the *level order* 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).

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

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

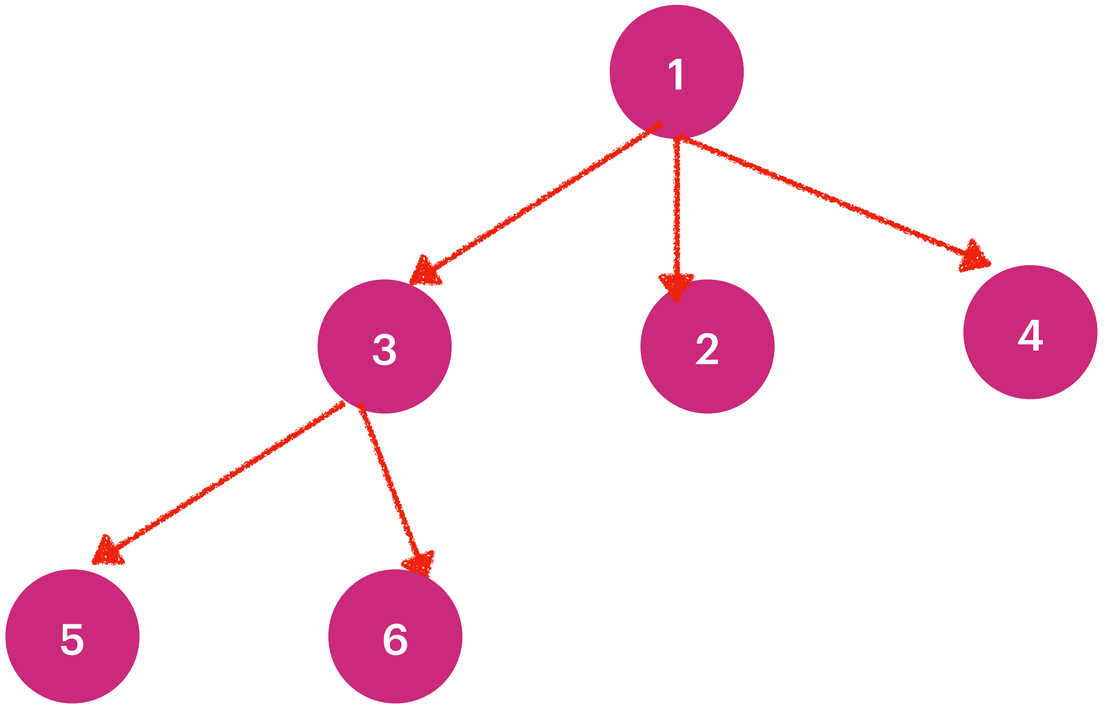
```
class Node {
public int val;
public List<Node> children;
}
```

Constraints:
The height of the n-ary tree is less than or equal to `1000`
The total number of nodes is between `[0, 10⁴]`

```
class Solution {
public List<List<Integer>> levelOrder(Node root) {
}
}
```

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

Construct N-ary Tree & Level Order Traversal



Output: [[1],[3,2,4],[5,6]]

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

Queue[node[1])

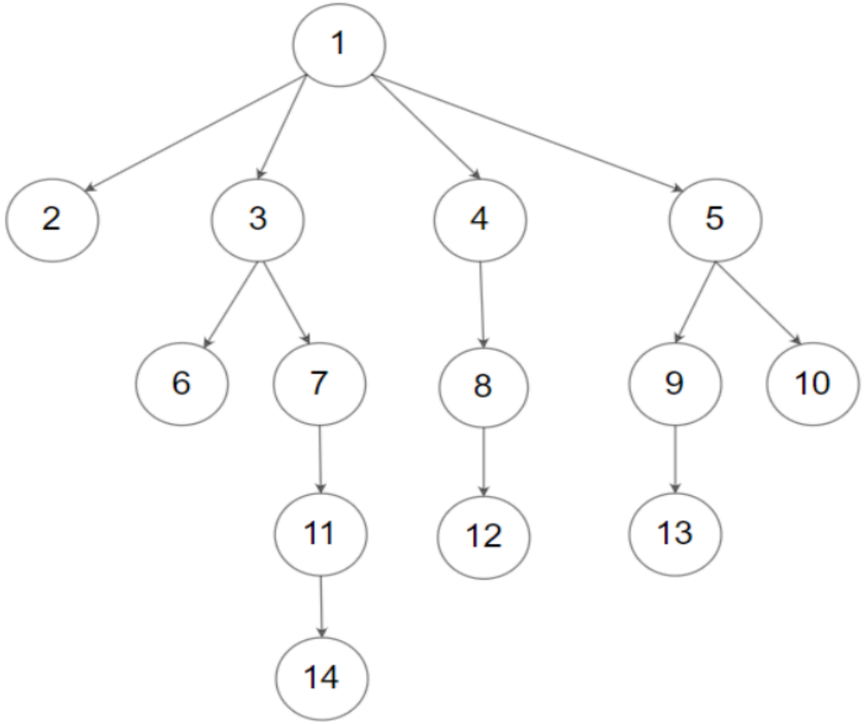
L1 [1]
Queue : [2,3,4,5]

L2[2,3,4,5]
Queue[6,7,8,9,10]

L3[6,7,8,9,10]
Queue[11,12,13]

L4 :[11,12,13]
Queue: [14]

L5[14]
Queue:[]



Construct N-ary Tree & Level Order Traversal

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