

Description

Editorial

Solutions (1.6K)

Submissions

590. N-ary Tree Postorder Traversal

Easy

2.2K

92

☆

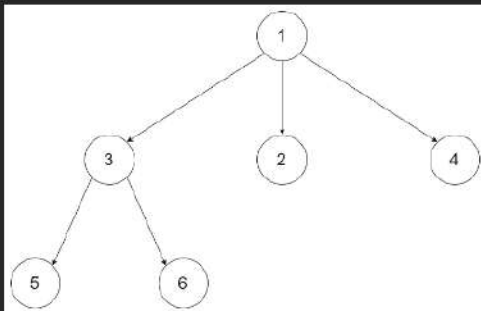
↻

Companies

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

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

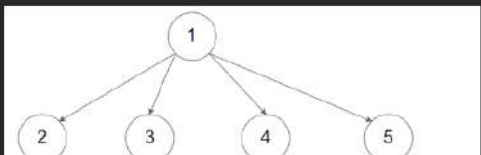
Example 1:



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

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

Example 2:



i Java

• Auto

```
17 };
18 */
19
20 class Solution {
21     List<Integer> ans=new ArrayList<>();
22     public List<Integer> postorder(Node root) {
23         if(root==null) return ans;
24         for(Node n: root.children){
25             postorder(n);
26         }
27         ans.add(root.val);
28         return ans;
29     }
30 }
```

Testcase

Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

`[1,null,3,2,4,null,5,6]`

Output

`[5,6,3,2,4,1]`

Expected

Console

⌵

Run

Submit

664. Strange Printer

Hard 1.6K 151

Companies

There is a strange printer with the following two special properties:

- The printer can only print a sequence of **the same character** each time.
- At each turn, the printer can print new characters starting from and ending at any place and will cover the original existing characters.

Given a string `s`, return the minimum number of turns the printer needed to print it.

Example 1:

Input: `s = "aaabbb"`

Output: 2

Explanation: Print "aaa" first and then print "bbb".

Example 2:

Input: `s = "aba"`

Output: 2

Explanation: Print "aaa" first and then print "b" from the second place of the string, which will cover the existing character 'a'.

Constraints:

i Java Auto

```
1 class Solution {
2     public int strangePrinter(String s) {
3         int n = s.length();
4         char[] sChar = s.toCharArray();
5         int[][] dp = new int[n][n];
6         for(int[] in : dp) Arrays.fill(in, -1);
7         return Util(0, n - 1, sChar, dp);
8     }
9     public int Util(int i, int j, char[] sChar, int[][] dp) {
10         if (i > j) {
11             return 0;
12         }
13
14         if(dp[i][j] != -1) return dp[i][j];
15
16         int firstLetter = sChar[i];
17         // in case, current character is not repeated in the rest of the string
18         int answer = 1 + Util(i + 1, j, sChar, dp);
```

Testcase Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

s =
"aaabbb"

Output

2

Console



Run

Submit