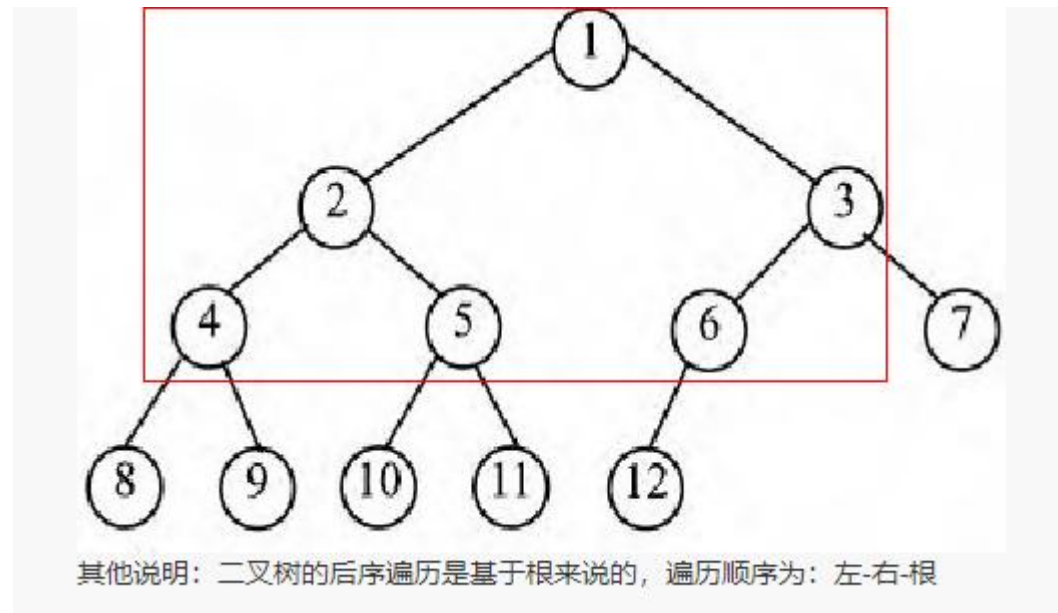


Java-数组树递归-给定一个以顺序储存结构存储整数值的完全二叉树序列

题目描述：

给定一个以顺序储存结构存储整数值的完全二叉树序列（最多 1000 个整数），请找出此完全二叉树的所有非叶子节点部分，然后采用后序遍历方式将此部分树（不包含叶子）输出。

- 1、只有一个节点的树，此节点认定为根节点（非叶子）。
- 2、此完全二叉树并非满二叉树，可能存在倒数第二层出现叶子或者无右叶子的情况



输入描述：

一个通过空格分割的整数序列字符串

输出描述：

非叶子部分树结构的后序遍历结果

补充说明：

输出数字以空格分隔

示例 1

输入：

1 2 3 4 5 6 7

输出：

2 3 1

说明：

找到非叶子部分树结构，然后采用后续遍历输出

import java.util.*;

// 注意类名必须为 Main, 不要有任何 package xxx 信息

public class Main {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

// 注意 hasNext 和 hasNextLine 的区别

String s = in.nextLine();

String[] ss = s.split(" ");

if(ss.length == 1) {

```

        System.out.println(ss[0]);
        return ;
    }
    helper(ss, 1);
}

public static boolean helper(String[] ss, int root){
    // System.out.print(root);
    int left = 2 * root, right = left + 1;
    // System.out.println(" " + left + " " + right) ;
    // if(right - 1 >= ss.length ){
    //     return false;
    // }
    // if(left - 1 >= ss.length){
    //     return false;
    // }
    if(root - 1 < ss.length){
        boolean t = helper(ss, left);
        t = helper(ss, right) || t;
        if(t == true){
            System.out.print(ss[root-1] + " ");
        }
        return true;
    }
    if(left - 1 < ss.length){
        return true;
    }
    return false;
}
}

```