递归思想的感悟Recursion

# 递归Recursion

## 递归思想的两个关键点

写递归算法的时候，只需要找到两个关键点就可以：

一是：递归终止条件；

二是：在下一级已经返回结果的假设下(递归)，如何做好当前的操作。

## 示例1：阶乘

利用递归实现阶乘：

首先，写递归终止条件：

if(n < 0) return -1; if(n == 0|| n == 1) return 1;

其次，在递归调用下一级已经返回结果的前提下，应当的操作：

return n\*jiecheng(n-1);

**阶乘算法**的实现代码如下：

**public int** jiecheng(**int** n){  
 **if**(n < 0) **return** -1;  
 **if**(n == 0|| n == 1) **return** 1;  
 **return** n\*jiecheng(n-1);  
}

## 示例2：二叉树的前序遍历

/\*\*

\* Definition for a binary tree node.

\* public class TreeNode {

\* int val;

\* TreeNode left;

\* TreeNode right;

\* TreeNode(int x) { val = x; }

\* }

\*/

class Solution {

public List<Integer> preorderTraversal(TreeNode root) {

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

**preOrder(root,result);**

return result;

}

private void preOrder(TreeNode root,List<Integer> result){

**if(root == null) return;//递归终止条件**

result.add(root.val);

preOrder(root.left,result);

preOrder(root.right,result);

}

}