二叉树后序中序转成先序或者层次遍历。

1. 数据结构
   * 1. 两个数组，存储后序和中序序列
     2. vector<int> lev(100000, -1) 用来存储层次遍历的序列。
2. 算法
   * 1. 由后序中序求层次遍历序列，中序提供切分子树序列，后序提供root

root 根节点 start 序列起点 end 序列终点 index是表示层次遍历点的序号

void level(int root,int start,int end,int index){中序root，后序start end

if (start > end) return;//start=end时节点无孩子，递归它时退出

int i = start;

while (i <= end&&in[i] != post[root]) i++;//在中序序列中找根节点的下标

lev[index]=in[i];//lev[index]存储当前序列的根节点

level(root - (end - i) - 1, start, i - 1, index \* 2 + 1);//递归根节点的左子树序列

level(root - 1, i + 1, end, index \* 2 + 2);//递归根节点的右子树序列

}

ii. 由后序中序求先序序列

void level(int root,int start,int end){

if (start > end) return;//start=end时节点无孩子，递归它时退出

int i = start;

while (i <= end&&in[i] != post[root]) i++;//在中序序列中找根节点的下标

cout<<in[i]<<’ ‘;//lev[index]存储当前序列的根节点

level(root - (end - i) - 1, start, i - 1);//递归根节点的左子树序列

level(root - 1, i + 1, end);//递归根节点的右子树序列

}