



链滴

重建二叉树

作者: [yudake](#)

原文链接: <https://ld246.com/article/1518749171788>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

题目描述

输入某二叉树的前序遍历和中序遍历的结果，请重建出该二叉树。假设输入的前序遍历和中序遍历的结果中都不含重复的数字。例如输入前序遍历序列{1,2,4,7,3,5,6,8}和中序遍历序列{4,7,2,1,5,3,8,6}，则建二叉树并返回。

解题思路

二叉树可以根据

- 前序+中序
- 后序+中序

恢复出来。

很明显可以利用递归：

- 利用中序寻找根节点；
- 前序或后序中根节点的左右即位于根节点的左右子树，设左子树含有N个结点；
- 中序根节点之后的N个结点就是左子树，其中第一个结点就是左子树根节点；
- 左子树之后的所有结点是右子树，其中第一个结点就是右子树根节点。

```
public class Solution {
    public TreeNode reConstructBinaryTree(int [] pre,int [] in) {
        TreeNode root = reConstructBinaryTree(pre, 0, pre.length-1, in, 0, in.length-1);
        return root;
    }

    private TreeNode reConstructBinaryTree(int[] pre, int startPre, int endPre, int[] in, int startIn, int endIn) {
        if (startPre > endPre || startIn > endIn)
            return null;
        TreeNode root = new TreeNode(pre[startPre]);
        for (int i = startIn; i <= endIn; i++) {
            if (in[i] == pre[startPre]) {
                root.left = reConstructBinaryTree(pre, startPre+1, startPre+i-startIn, in, startIn, i-1);
                root.right = reConstructBinaryTree(pre, startPre+i-startIn+1, endPre, in, i+1, endIn);
                break;
            }
        }
        return root;
    }
}
```