# 重建二叉树

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}，则重建二叉树并返回。

## 解题思路

1. 通过前序遍历找到 root 节点
2. 那么在 中序遍历中 root 节点的左侧则是左子树，右侧是右子树
3. 依次类推，递归生成节点的左子树和右子树
4. 构建过程由下往上

public TreeNode reConstructBinaryTree(int[] pre, int[] in) {
 Map<Integer, Integer> preIndex = new HashMap<>();
 for (int i = 0; i < pre.length; i++) {
 preIndex.put(pre[i], i);
 }

 return buildTree(preIndex, in, 0, in.length - 1);
}

private TreeNode buildTree(Map<Integer, Integer> preIndex, int[] in, int start, int end) {
 if (start == end) {
 return new TreeNode(in[start]);
 }
 int indexOfRoot = start;
 for (int i = start; i <= end; i++) {
 if (preIndex.get(in[i]) < preIndex.get(in[indexOfRoot])) {
 indexOfRoot = i;
 }
 }
 TreeNode root = new TreeNode(in[indexOfRoot]);
 if (start <= indexOfRoot - 1) root.left = buildTree(preIndex, in, start, indexOfRoot - 1);
 if (indexOfRoot + 1 <= end) root.right = buildTree(preIndex, in, indexOfRoot + 1, end);
 return root;
}