106. Construct Binary Tree from Inorder and Postorder Traversal

Given inorder and postorder traversal of a tree, construct the binary tree.

```
class Solution {
    public TreeNode buildTree(int[] inorder, int[] postorder) {

        if (inorder == null || postorder == null || inorder.length !=
postorder.length)
            return null;

        HashMap<Integer, Integer> map = new HashMap<Integer,Integer>();

        for (int i=0;i<inorder.length;++i)
            map.put(inorder[i], i);

        return buildTreePostIn(inorder, 0, inorder.length-1, postorder, 0,
                postorder.length-1,map);
    }

    private TreeNode buildTreePostIn(int[] inorder, int inStart, int inEnd,
                                     int[] postorder, int poStart, int poEnd,
                                     HashMap<Integer,Integer> map){

        if (poStart > poEnd || inStart > inEnd)
            return null;

        TreeNode root = new TreeNode(postorder[poEnd]);
        int ri = map.get(postorder[poEnd]);

        TreeNode leftchild = buildTreePostIn(inorder, inStart, ri-1,
postorder,
                                             poStart, poStart+ri-inStart-1,
map);
        TreeNode rightchild = buildTreePostIn(inorder, ri+1, inEnd, postorder,
                                              poStart+ri-inStart, poEnd-1,
map);
        root.left = leftchild;
        root.right = rightchild;
        return root;
    }
}
```