



链滴

Leetcode 每日刷题 (106. 从中序与后序遍历序列构造二叉树)

作者: [ludengke95](#)

原文链接: <https://ld246.com/article/1601031514782>

来源网站: 链滴

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



LeetCode的每日抑题系列, LeetCode会每天随机一道题作为签到题.我也是菜鸡,如果今天没A掉,那只证明我离大神又进了一步.

题目链接

题名	通过率	难度
106. 从中序与后序遍历序列构造二叉树 等		70.6%

根据一棵树的中序遍历与后序遍历构造二叉树。

注意:

你可以假设树中没有重复的元素。

例如, 给出

中序遍历 inorder = [9,3,15,20,7]

后序遍历 postorder = [9,15,7,20,3]

解题思路

递归,不说了,递就完事了.(有人玩怪物猎人嘛,解就完事了.)

根据后序遍历找到根节点,把中序和后序切分成左右子树两部分,递归处理

java代码

```
import scala.collection.mutable._
```

```

/**
 * 描述:
 * ${DESCRIPTION}
 *
 * @author ludengke
 * @create
 */
object Solution3 extends App {
  def buildTree(inorder: Array[Int], postorder: Array[Int]): TreeNode = {
    buildTree(inorder,0,inorder.length-1,postorder,0,inorder.length-1)
  }

  def buildTree(inOrder: Array[Int], inStart: Int, inEnd: Int, postOrder: Array[Int], postStart: Int,
ostEnd: Int): TreeNode = {
    if(inEnd < inStart || postStart > postEnd){
      null
    }else{
      val nodeValue = postOrder(postEnd)
      val index = inOrder.indexOf(nodeValue)
      val node = new TreeNode(nodeValue)
      node.left = buildTree(inOrder,inStart,index - 1,postOrder,postStart,index - 1 - inStart + po
tStart)
      node.right = buildTree(inOrder,index + 1,inEnd,postOrder,index - inStart + postStart,post
nd - 1)
      node
    }
  }

  val t = buildTree(Array(9,3,15,20,7),Array(9,15,7,20,3))
  println(t)
}

```

算法or数据结构

1. 递归, 递归, 递归