LeetCode 题库分门别类详细解析笔记（第7章）

第7章 二叉树和递归（20题） 1

7-1 从二叉树的角度看递归（2题） 1

 （完成）例题1：LeetCode 第 104 题：给定一个二叉树，找出其最大深度。 1

 （完成）练习1：LeetCode 第 111 题： 1

7-2 一个简单的二叉树引发的血案（5题） 1

 （完成）例题1：LeetCode 第 226 题：反转一棵二叉树（youtube 上缺少视频） 1

 （完成）练习1：LeetCode 第 100 题： 2

 （完成）练习2：LeetCode 第 101 题（典型问题） 2

 （完成）练习3：LeetCode 第 222 题 2

 （完成）练习4：LeetCode 第 110 题 3

7-3 注意递归的终止条件（3题） 3

 （完成）例题1：LeetCode 第 112 题Path Sum 3

 （完成）练习1：LeetCode 第 111 题（回顾） 3

 （完成）练习2：LeetCode 第 404 题（油管视频缺少下载） 3

7-4 如何使用递归函数的返回值（3题） 4

 例题1：LeetCode 第 257 题 4

 （完成）练习1：LeetCode 第 113 题 4

 （完成）练习2：LeetCode 第 129 题 4

7-5 稍复杂的递归逻辑 Path Sum III（1题） 4

 例题1：LeetCode 第 437 题 4

7-6 二分搜索树中的问题（6题） 5

 复习：二分搜索树的相关操作 5

 例题1：LeetCode 第 235 题 5

 练习1：LeetCode 第 98 题 5

 练习2：LeetCode 第 450 题 6

 练习3：LeetCode 第 108 题 7

 练习4：LeetCode 第 230 题：二叉搜索树中第K小的元素 8

 练习5：LeetCode 第 236 题 9

第7章 二叉树和递归（20题）

递归，是使用计算机解决问题的一种重要的思考方式。而二叉树由于其天然的递归结构，使得基于二叉树的算法，均拥有着递归性质。使用二叉树，是研究学习递归算法的最佳入门方式。在这一章里，我们就来看一看二叉树中的递归算法。

7-1 从二叉树的角度看递归（2题）

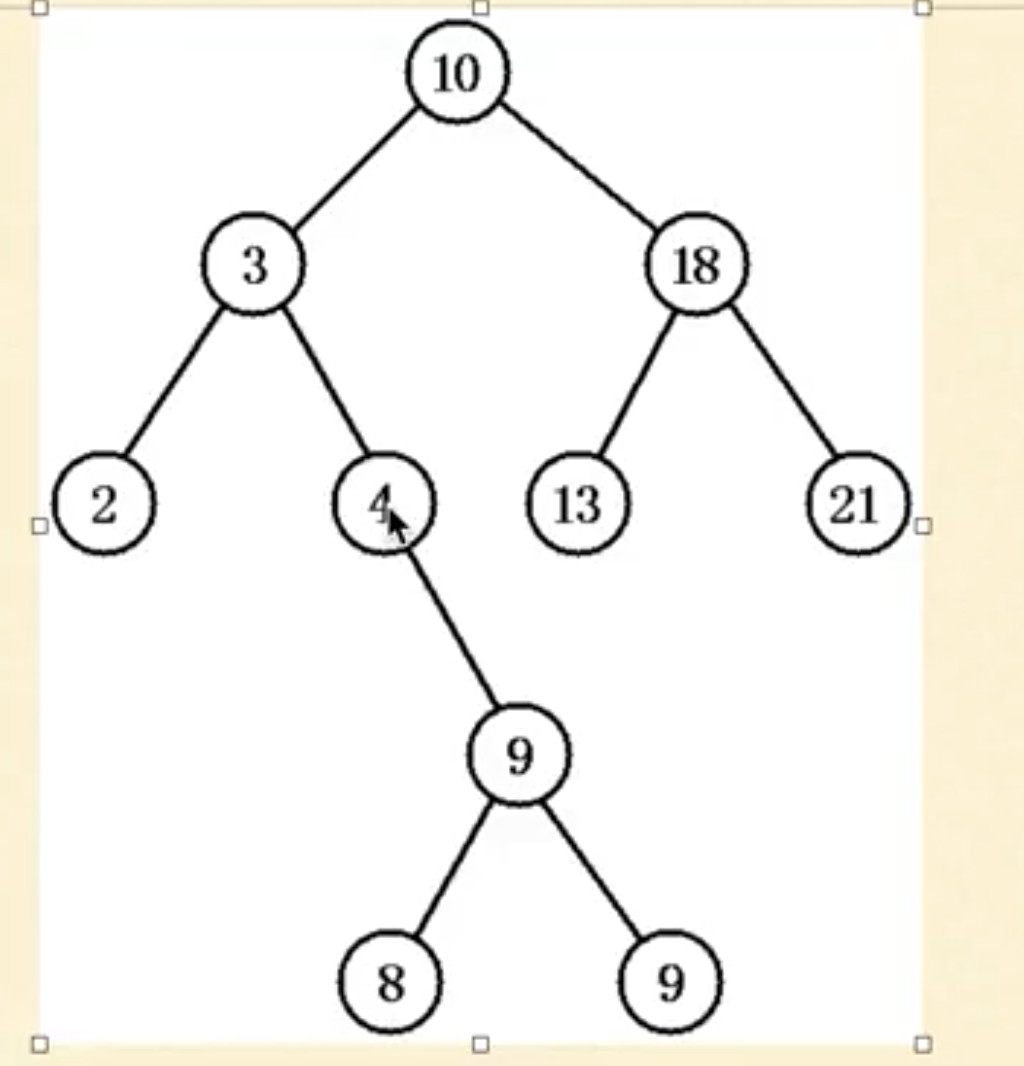
* （完成）例题1：LeetCode 第 104 题：给定一个二叉树，找出其最大深度。

https://leetcode-cn.com/problems/maximum-depth-of-binary-tree/description/

使用递归的方法，考虑清楚递归终止条件，解答是非常简单且清楚的。

* （完成）练习1：LeetCode 第 111 题：

提示：给定一个二叉树，找出其最小深度。最小深度是从根节点到最近叶子节点的最短路径上的节点数量。



这道题，我第一次做是想当然，顺着第 104 题把最大改成最小，但是要注意到上图 4 那个节点，4 的左孩子为空，返回 0 ，右孩子为 9，返回2，按照我们的逻辑就返回 0 ，显然是错误的，所以要针对左右孩子有一个为空的时候，做出分类判断。

7-2 一个简单的二叉树引发的血案（5题）

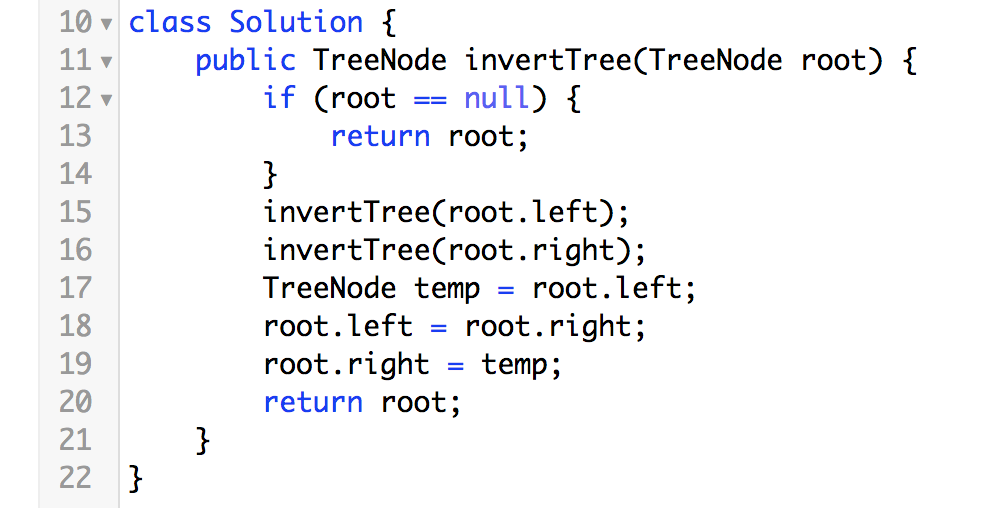
* （完成）例题1：LeetCode 第 226 题：反转一棵二叉树（youtube 上缺少视频）

https://leetcode-cn.com/problems/invert-binary-tree/description/

反转的时候，有两种方式：

（1）可以先把左右子树反转了，再交换左右子树的位置。（如下实现）

（2）先交换左右子树的位置，再反转。（还未实现）



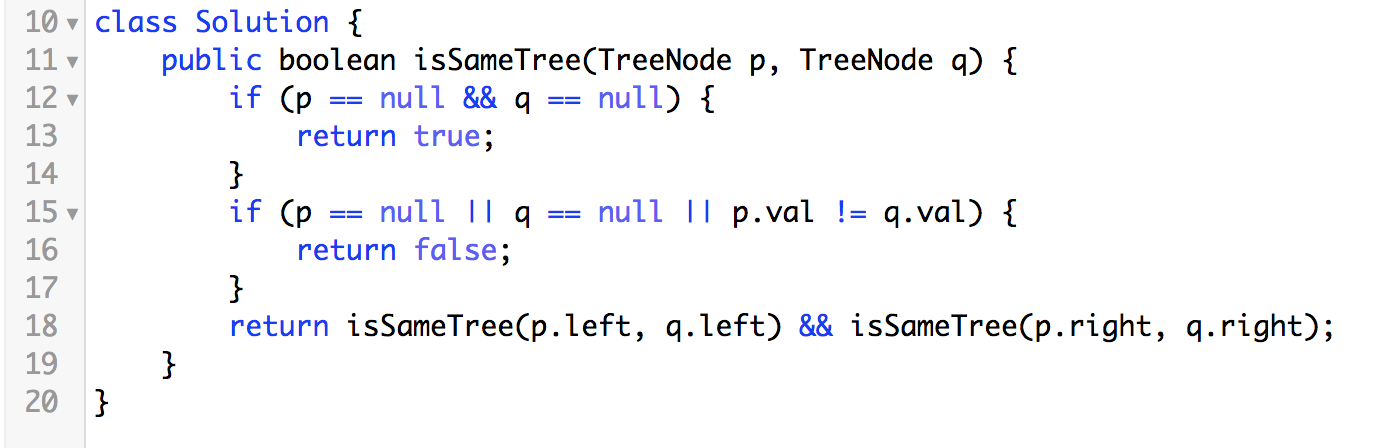
我第一次的写法是这样的。



* （完成）练习1：LeetCode 第 100 题：

https://leetcode-cn.com/problems/same-tree/description/

提示：判断两棵二叉树是否一样。



* （完成）练习2：LeetCode 第 101 题（典型问题）

提示：判断两棵二叉树是否左右对称（镜面对称）。

思路1：先拷贝一棵二叉树，再反转，将反转以后的二叉树和自己比较，看看是否相等。

思路2：递归，这里需要设置一个辅助函数。

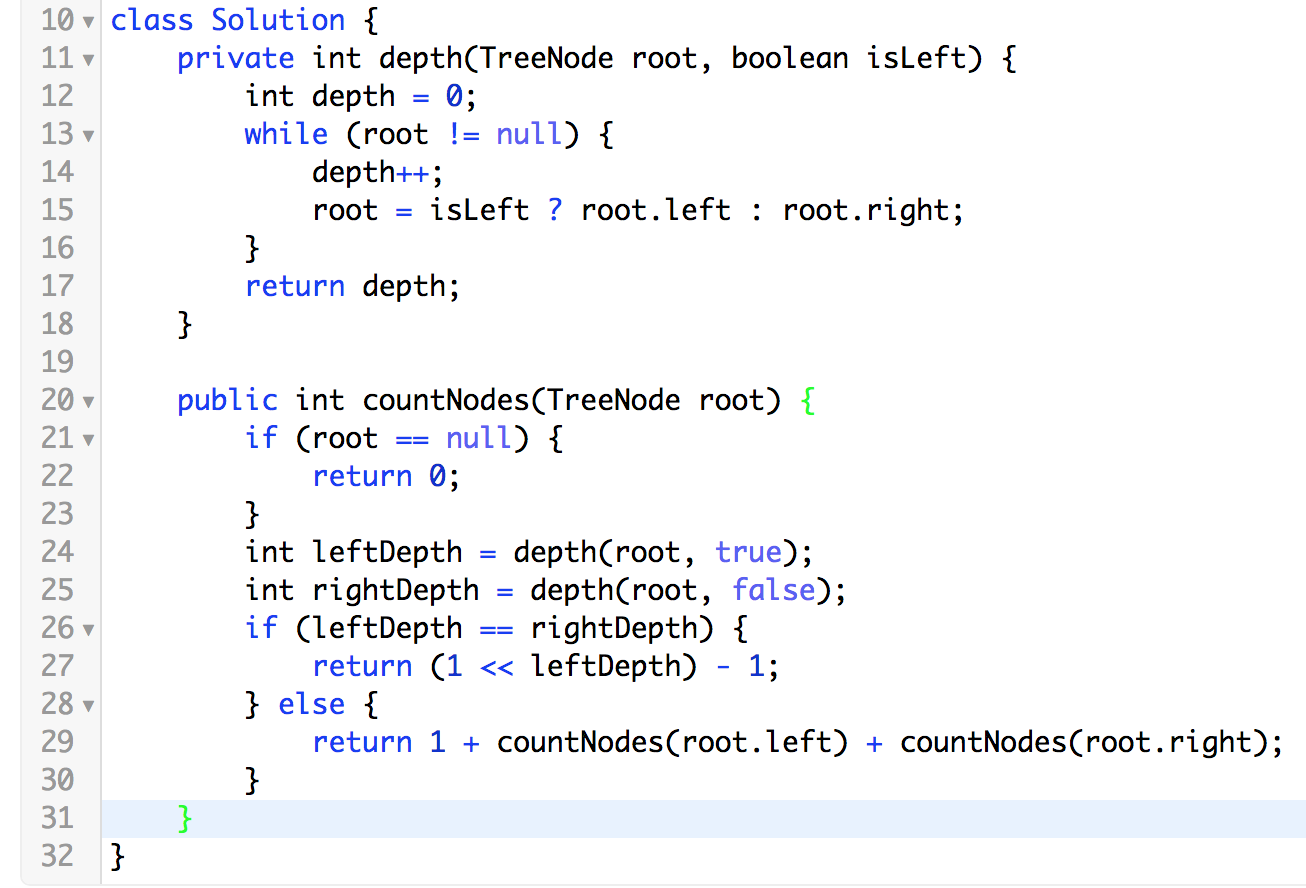
思路3：使用队列，并且是双端队列（链表实现）这个辅助数据结构。画出出队入队的顺序，就很清楚了。

参考资料：【LeetCode】101. Symmetric Tree 解题报告https://blog.csdn.net/crazy1235/article/details/51541984

* （完成）练习3：LeetCode 第 222 题

提示：求完全二叉树的节点数、满二叉树。使用**递归**可以完成。

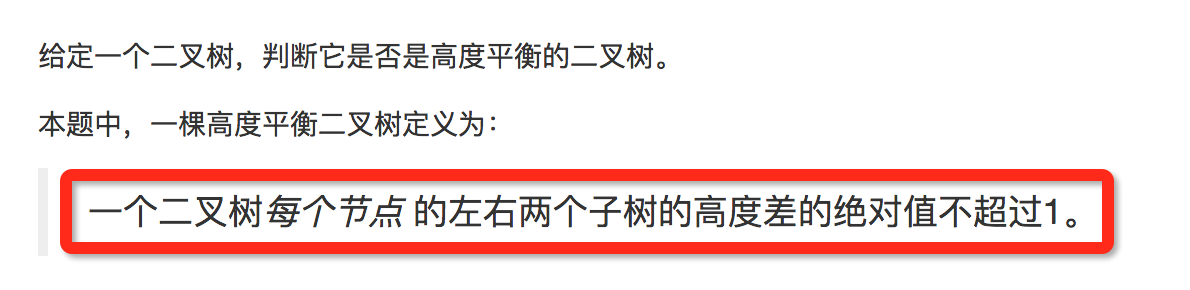
https://leetcode-cn.com/problems/count-complete-tree-nodes/



* （完成）练习4：LeetCode 第 110 题

提示：（1）题意：判断一棵二叉树是否是平衡二叉树。

（2）严格按照定义，使用后序遍历计算每一个节点的度，因为要计算一个整数，所以递归函数应该设计成一个返回 int 的函数。



7-3 注意递归的终止条件（3题）

* （完成）例题1：LeetCode 第 112 题Path Sum

https://leetcode-cn.com/problems/path-sum/description/

提示：想清楚递归关系，解答就变得非常简单了。

* （完成）练习1：LeetCode 第 111 题（回顾）
* （完成）练习2：LeetCode 第 404 题（油管视频缺少下载）

https://leetcode-cn.com/problems/sum-of-left-leaves/description/

提示：计算一棵二叉树的左叶子节点的和。



非递归写法：参考资料：LeetCode 404. Sum of Left Leaves 解题报告

https://blog.csdn.net/camellhf/article/details/52682113

7-4 如何使用递归函数的返回值（3题）

* 例题1：LeetCode 第 257 题

题目要求：得到二叉树的所有路径。

中文网址：https://leetcode-cn.com/problems/binary-tree-paths/description/

求解关键：使用深度优先遍历，遍历到根节点的时候输出结果。

* （完成）练习1：LeetCode 第 113 题

题目要求：给定一棵二叉树，返回所有从根节点到叶子节点的路径，并且要等于指定的和。

中文网址：https://leetcode-cn.com/problems/path-sum-ii/description/

求解关键：发现递归关系。

我的解答：https://gist.github.com/liweiwei1419/26d63eee30548977022d94ac3850445f

* （完成）练习2：LeetCode 第 129 题

题目要求：给定一棵二叉树，每个节点只包含数字 0-9，从根节点到叶子节点的每条路径可以表示成一个数，求这些数的和。

中文网址：https://leetcode-cn.com/problems/sum-root-to-leaf-numbers/description/

求解关键：

解法1：使用 path，递归回溯的常规解法。

解法2（推荐）：（使用递归）使用 cumsum 这个概念，即开始遍历到这个根节点的之前，已经有了 cumsum ，代码写出来也是非常简洁。

7-5 稍复杂的递归逻辑 Path Sum III（1题）

* 例题1：LeetCode 第 437 题

题目要求：

给定一个二叉树，它的每个结点都存放着一个整数值。

找出路径和等于给定数值的路径总数。

**路径不需要从根节点开始，也不需要在叶子节点结束**，但是路径方向必须是向下的（只能从父节点到子节点）。

二叉树不超过1000个节点，且节点数值范围是 [-1000000,1000000] 的整数。

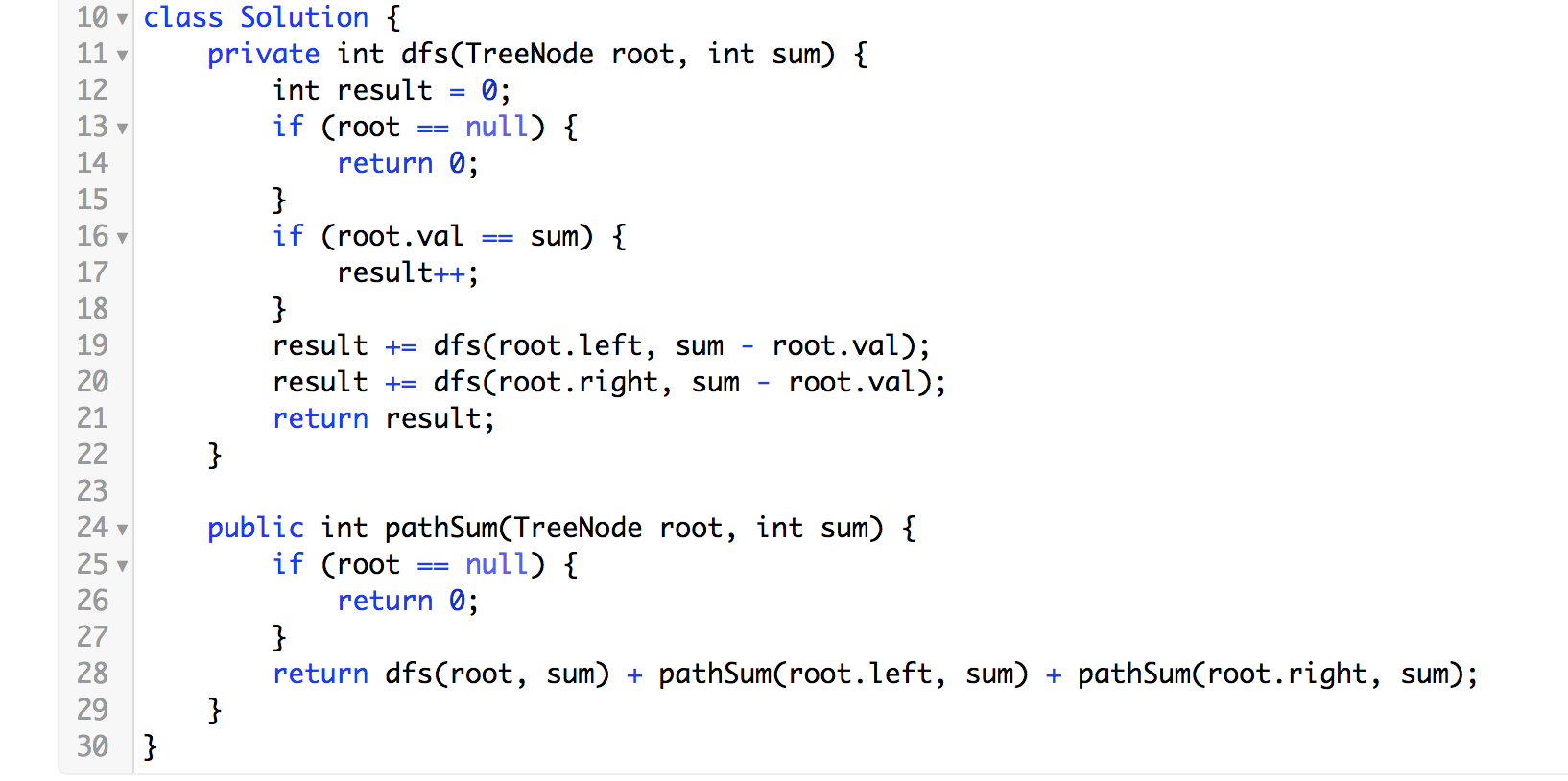
题目难度：**简单**。

英文网址：

中文网址：https://leetcode-cn.com/problems/path-sum-iii/description/

求解关键：还是发现递归关系，这里的递归关系，就是包含 root 节点和不包含根节点的解数之和，就是我们要求的。（**通过这一道题的解法，可以让我们进一步理解递归方法的威力。**）

我的解答：



参考资料：（还有一种使用 HashMap 的方法了解一下。）

7-6 二分搜索树中的问题（6题）

* 复习：二分搜索树的相关操作

题目要求：

题目难度：**简单**。

英文网址：

中文网址：https://leetcode-cn.com/problems/lowest-common-ancestor-of-a-binary-search-tree/description/

求解关键：

我的解答：

* 例题1：LeetCode 第 235 题

题目要求：给定一个二叉搜索树, 找到该树中两个指定节点的最近公共祖先。

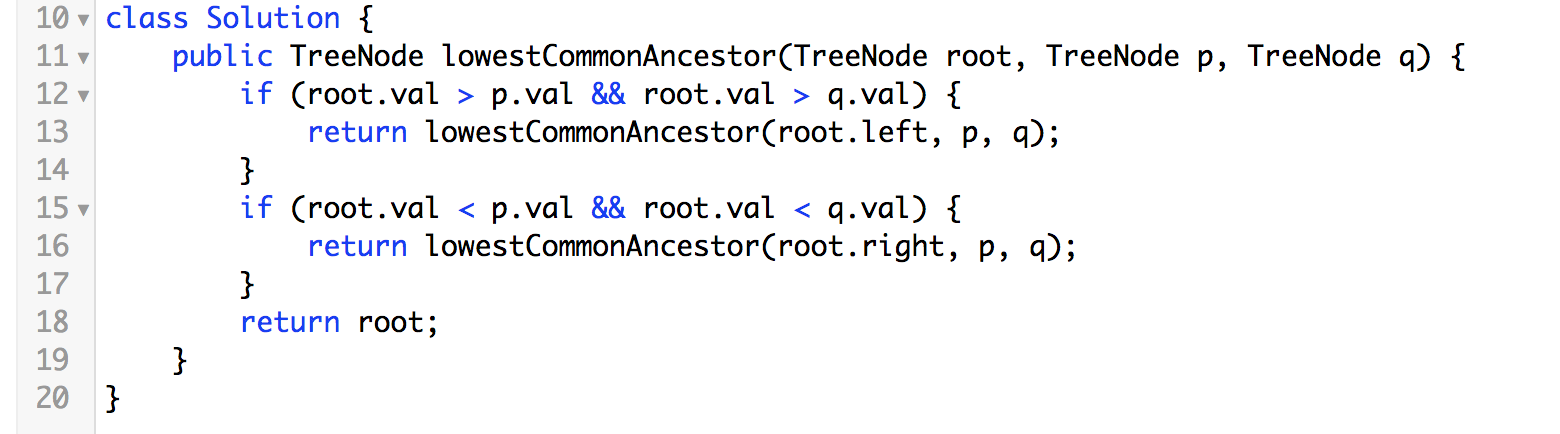
题目难度：**简单**。

英文网址：

中文网址：https://leetcode-cn.com/problems/lowest-common-ancestor-of-a-binary-search-tree/description/

求解关键：通过分类讨论，发现递归关系。在使用递归方法解决问题的时候，我们常常发现，完备且无重复地处理好分类讨论的问题，其实就是问题的求解。

我的解答：



* 练习1：LeetCode 第 98 题

题目要求：给定一个二叉树，判断其是否是一个有效的二叉搜索树。一个二叉搜索树具有如下特征：（1）节点的左子树只包含小于当前节点的数。（2）节点的右子树只包含大于当前节点的数。（3）所有左子树和右子树自身必须也是二叉搜索树。

题目难度：**中等**。

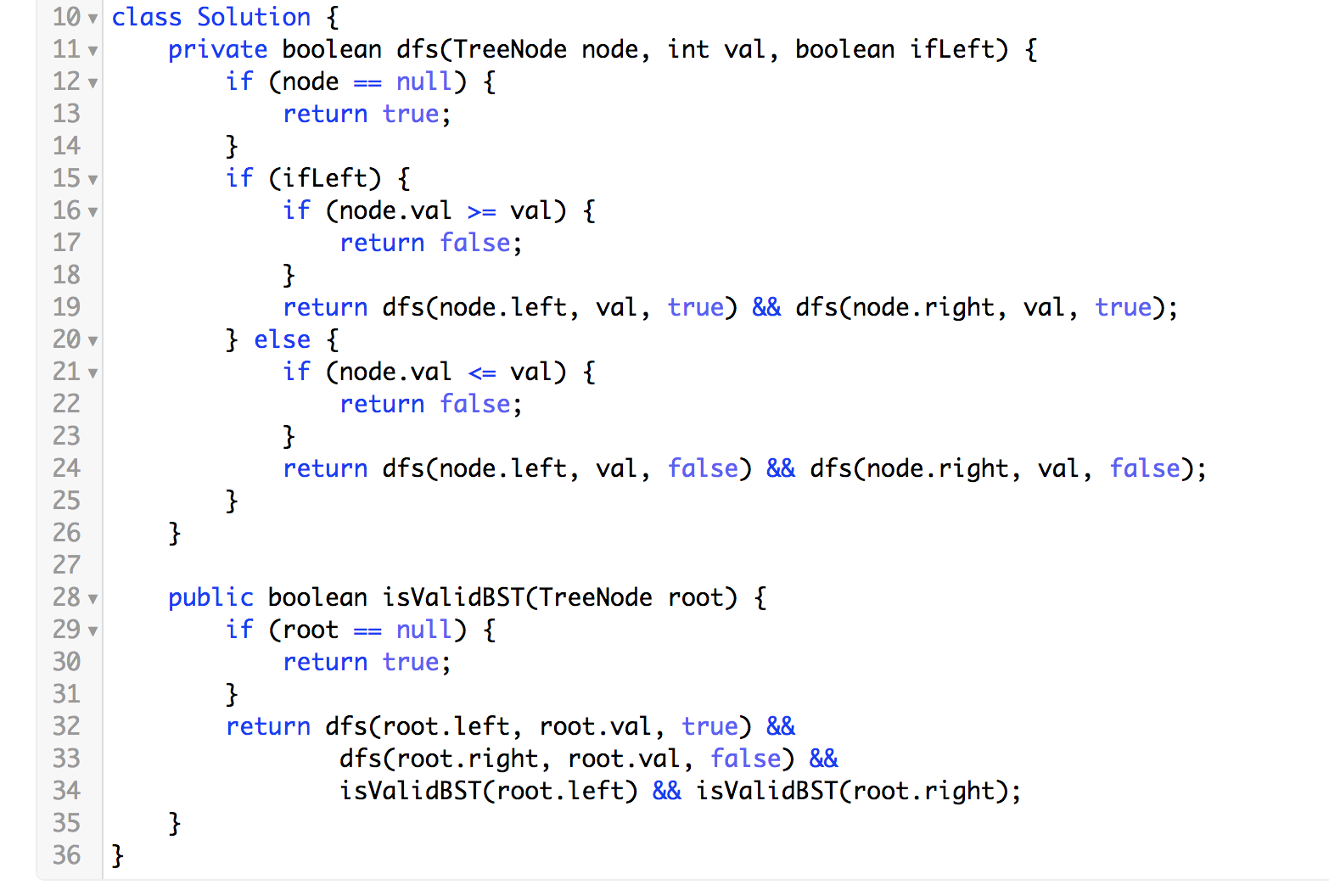
英文网址：

中文网址：https://leetcode-cn.com/problems/validate-binary-search-tree/description/

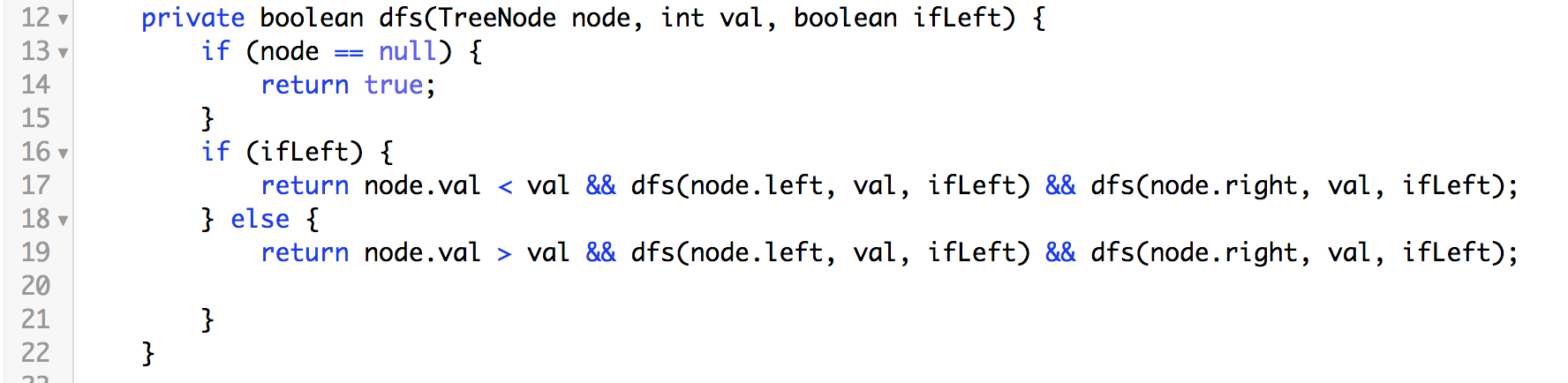
求解关键：

我的解答：

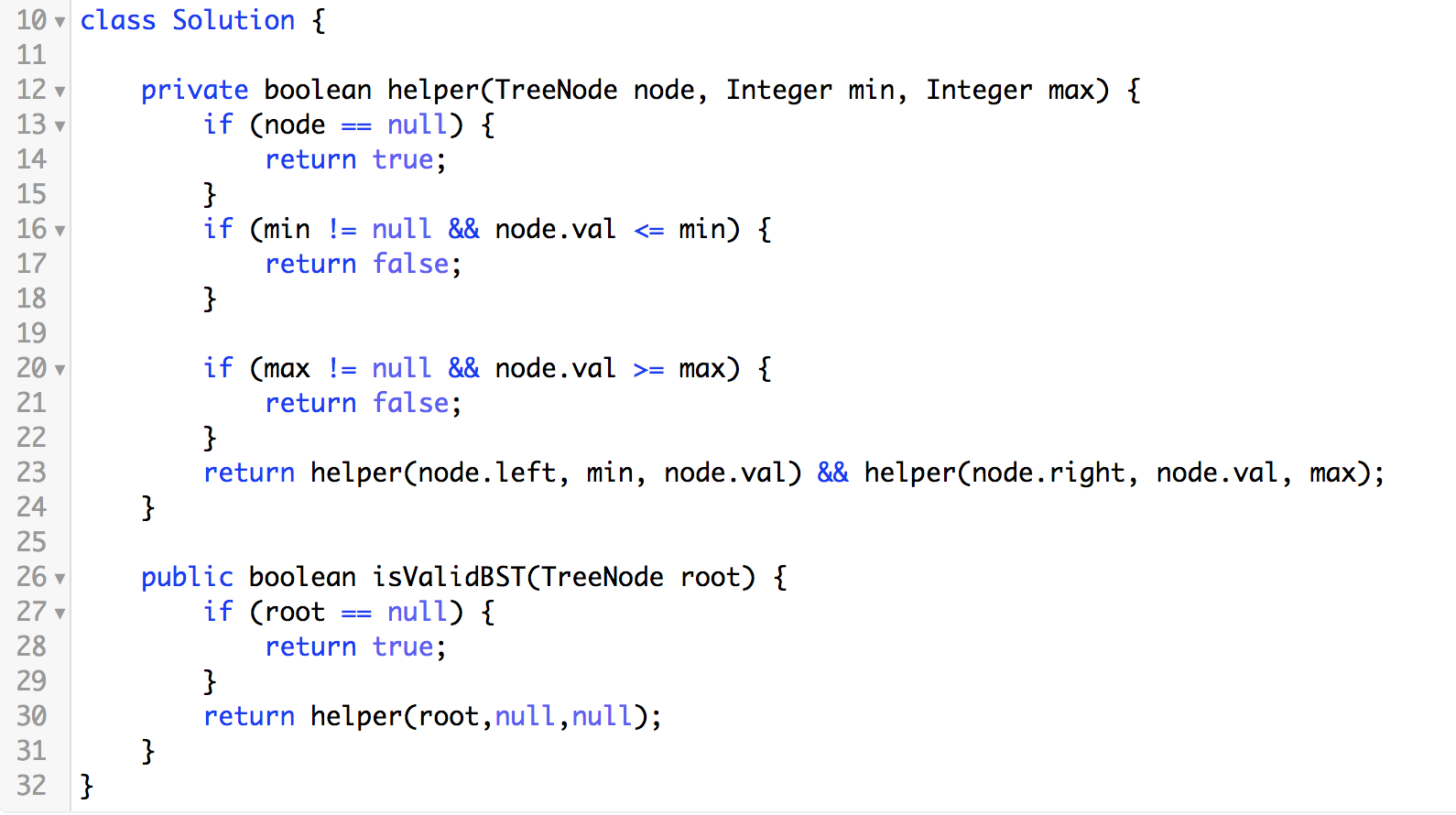
解法1：我一开始的解法是严格根据二分搜索树的定义的（1）（2）（3），提交以后虽然通过了，但是在提交的那一刹那，我觉得还有更好的解法。



（1）写法太啰嗦，可以再简化一下。



解法2：



* 练习2：LeetCode 第 450 题

题目要求：删除节点是一个比较复杂的操作，一定要会。

英文网址：

中文网址：https://leetcode-cn.com/problems/delete-node-in-a-bst/description/

求解关键：分类讨论，并且使用两个递归的辅助函数，辅助函数的定义是很关键的，并且理解“**返回执行了某个操作以后新的二分搜索树的根**”。

我的解答：

* 练习3：LeetCode 第 108 题

**参考思路：利用中序遍历能得到一个二分搜索树所有节点的有序排列。**

题目要求：

英文网址：

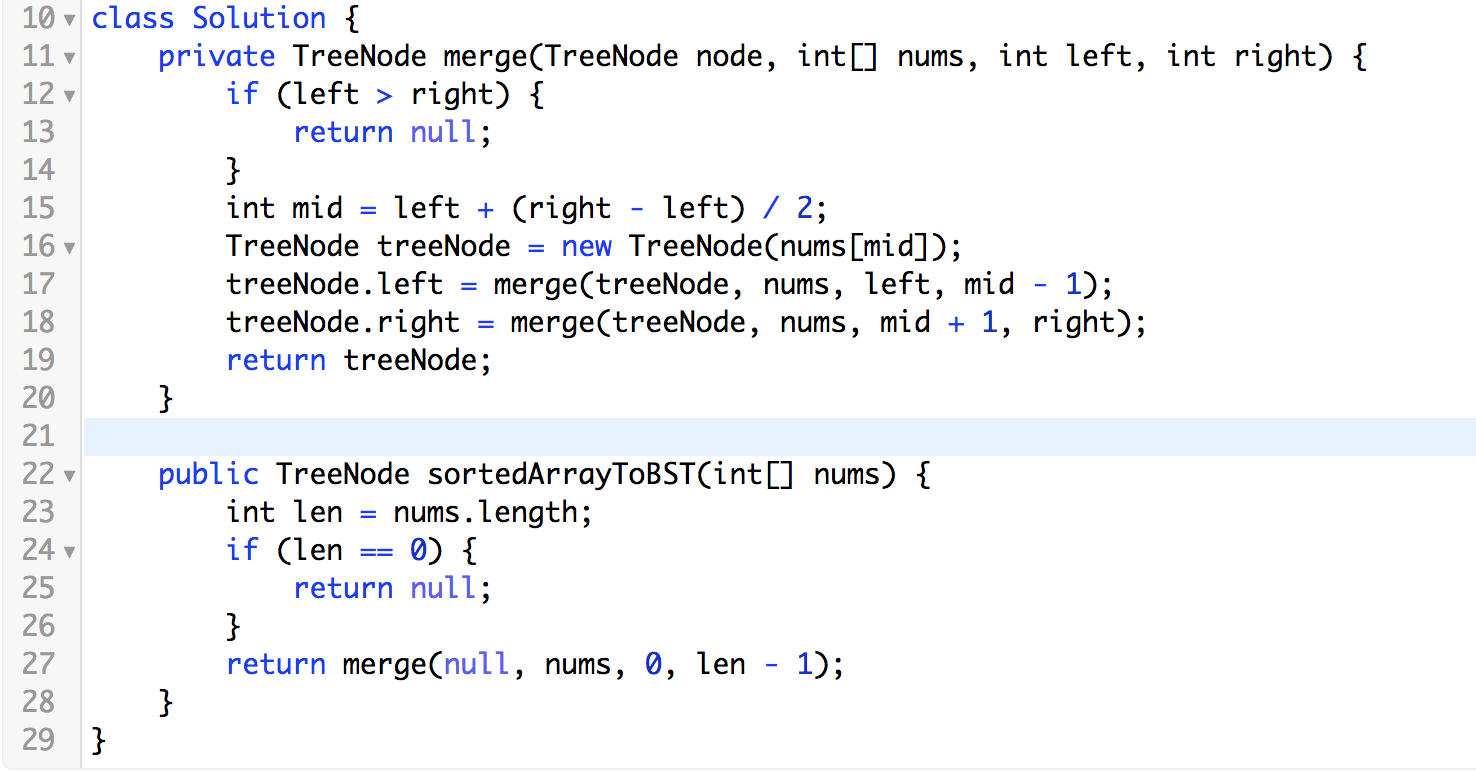
中文网址：

求解关键：

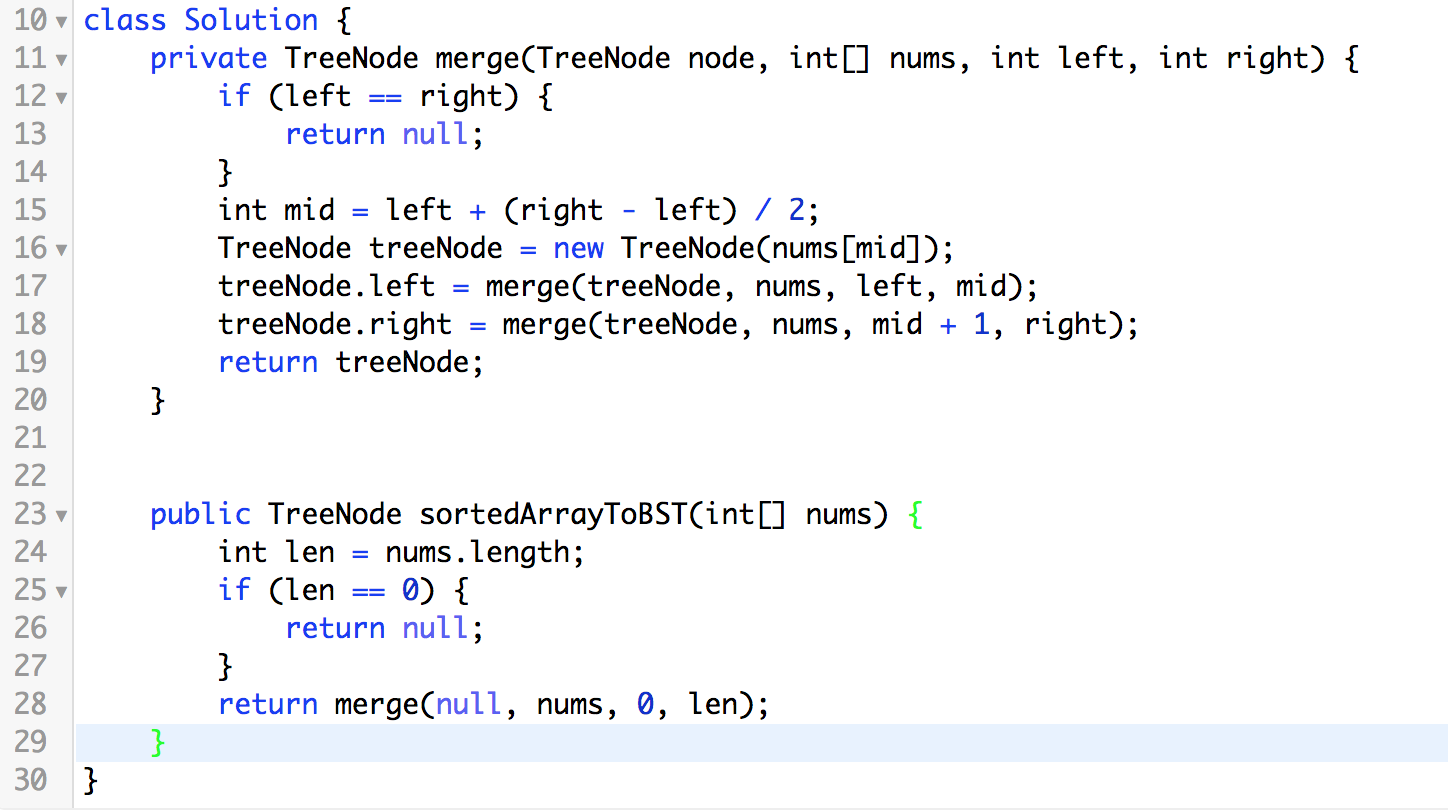
我的解答：

解答1（错误）：简单地认为从中间开始插入元素就可以了，但实际上连最基本的测试用例都没有通过。

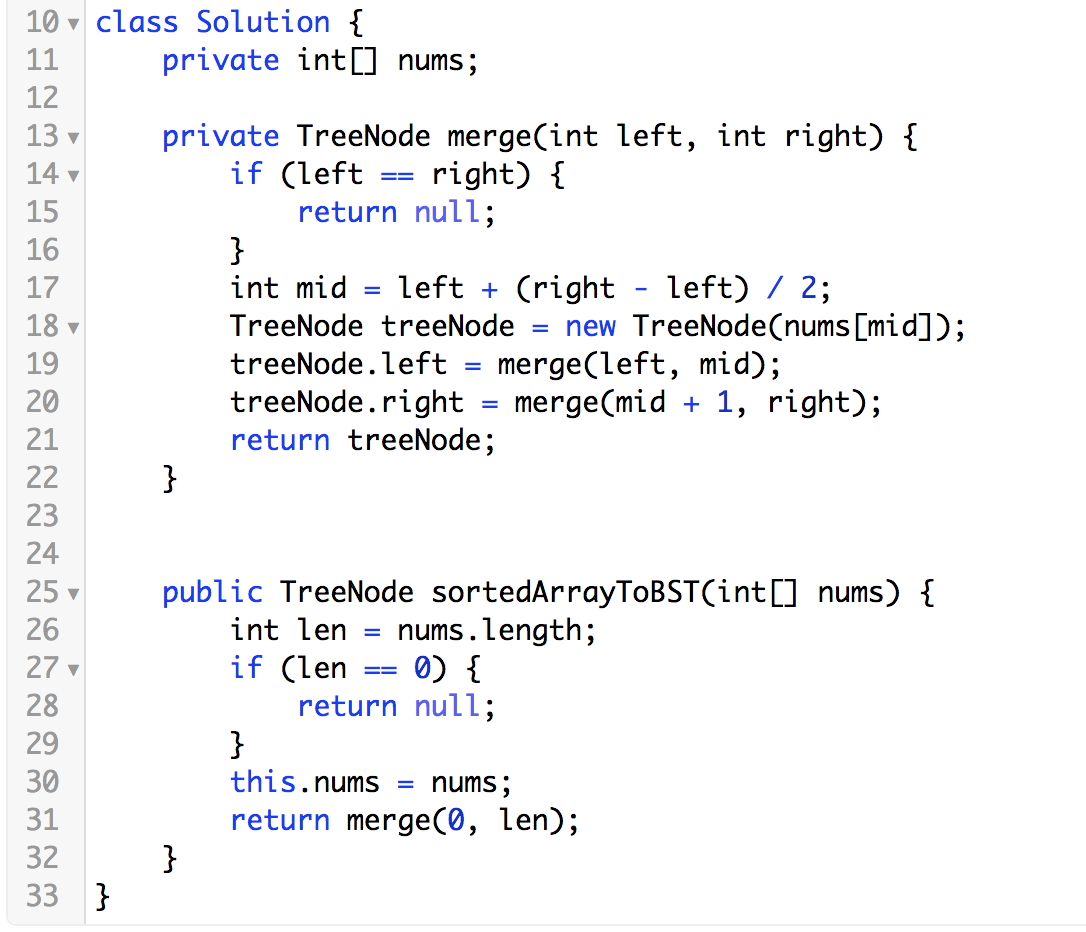
解答2（正确）：使用分而治之的策略，借鉴了归并排序的思想。

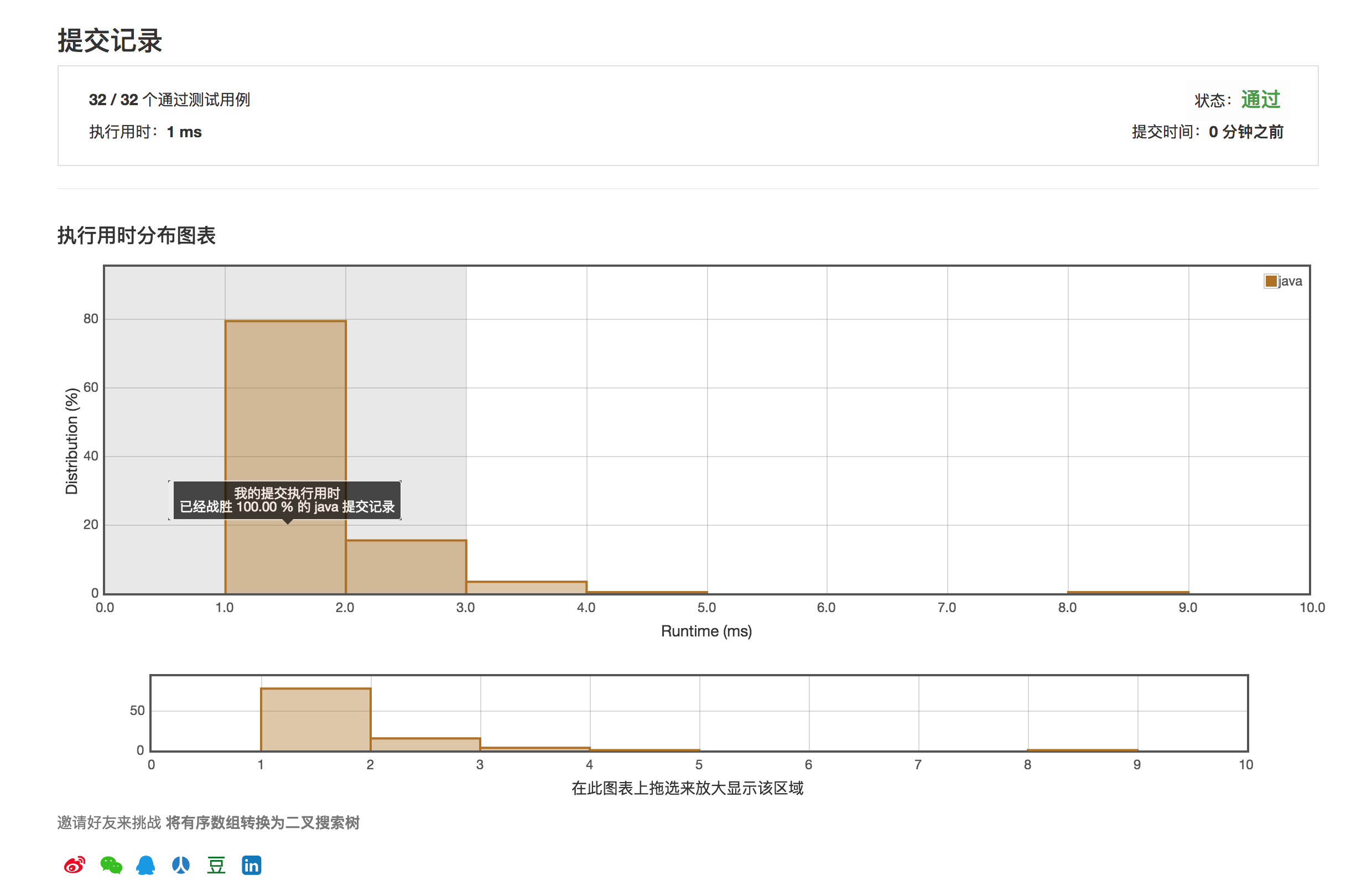


下面是一个相同的解法，只不过定义区间的方式不一样：



最后，整理了一下，发现 TreeNode 节点是没有必要传进来的，并且 nums 数组可以设置成成员变量，也不用在方法中传递参数。





* 练习4：LeetCode 第 230 题：二叉搜索树中第K小的元素

提示：因为二分搜索树具有顺序性，所以我们可以用类似快速排序的 partition 操作来完成。（这个提示感觉有点奇怪，直接使用中序遍历就可以完成。）

题目要求：给定一个二叉搜索树，编写一个函数kthSmallest来查找其中第 k 个最小的元素。

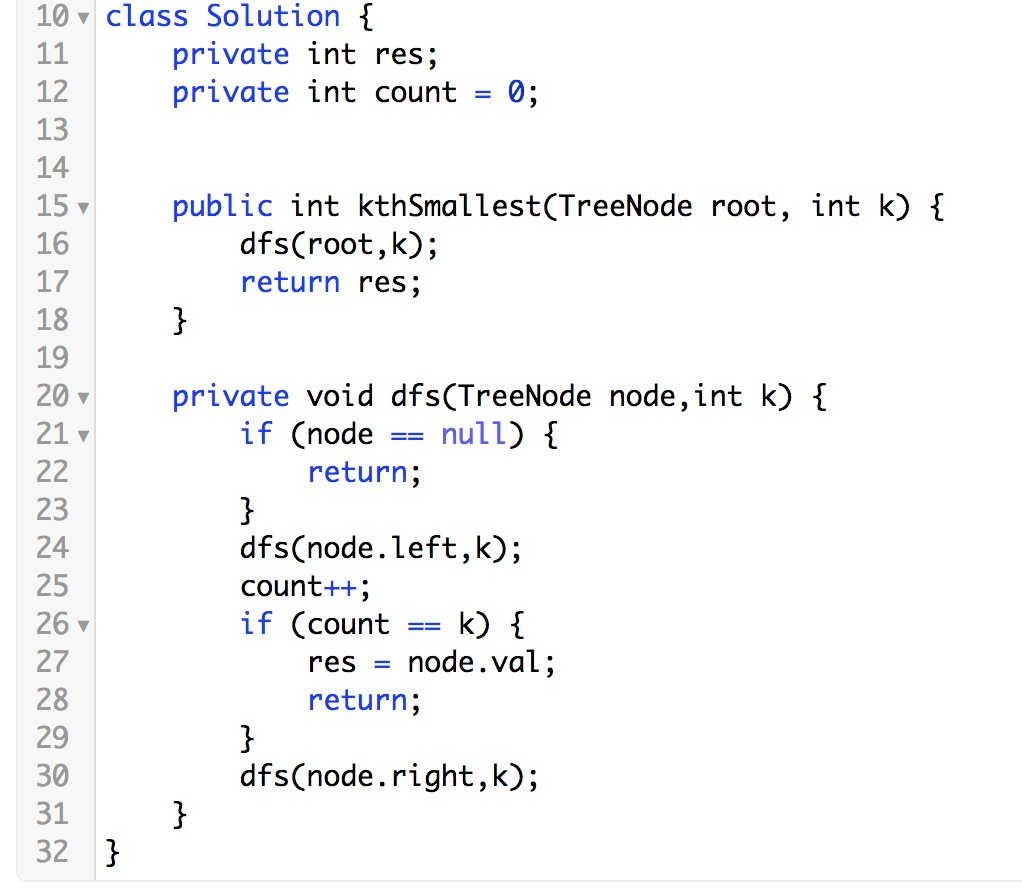
题目难度：**中等**。（其实就是有陷阱，一点都不难。）

英文网址：

中文网址：https://leetcode-cn.com/problems/kth-smallest-element-in-a-bst/description/

求解关键：**这里一定要把计数的变量设置成“成员变量”，如果设置成局部变量，就得不到正确的结果（涉及的知识点是最最基础的“变量的作用域”）。**

我的解答：



* 练习5：LeetCode 第 236 题

题目要求：二叉树的最近公共祖先。给定一棵二叉树, 找到该树中两个指定节点的最近公共祖先。百度百科中最近公共祖先的定义：“对于有根树T的两个结点u、v，最近公共祖先表示一个结点x，满足x是u、v的祖先且x的深度尽可能大。”（一个节点也可以是它自己的祖先）。（著名的 LCA 问题）。

题目难度：**中等**。

英文网址：

中文网址：https://leetcode-cn.com/problems/lowest-common-ancestor-of-a-binary-tree/description/

求解关键：分治法。其实最终的解法写出来就是一棵二叉树的**后序遍历，在后序遍历的过程中，做相应的逻辑判断**。

我的解答：

