

Binary Tree, Divide & Conquer

#G面试准备

Validate Binary Search Tree

Validate Binary Search Tree - LeetCode

- 分治法
- 需要一个ResultType 返回该root子树的最小值 最大值 以及该子树是否是BST
- 如果左或者右子树不是BST 就直接返回false
- 比较：
 - root节点值应该大于左子树最大值
 - root节点值应该小于右子树最小值
 - 题目中是
 - The left subtree of a node contains only nodes with keys less than the node's key.
 - The right subtree of a node contains only nodes with keys greater than the node's key
 - 所以不要考虑等于的情况
- 返回：
 - 子树最小值 root值和左的最小比
 - 子树最大值 root值和右的最大比
 - isBST = true

Binary Tree Longest Consecutive Sequence II

Binary Tree Longest Consecutive Sequence II

- 分治法
- helper返回值：一个两位的array 存由root开始的increase sequence和decrease sequence的长度
- 对左和右分别判断：
 - 左不空
 - root.val是否正好比左小1？ 是就是increase sequence 不是的话 左边的inc设为0（后面算总长度就用得上）
 - root.val是否正好比左大1？ 是就是decrease 不是的话左边的dec设为0
 - 右同
- 先更新全局长度 max, 左加+右减+1（root自己），左减+右加+1
- 返回值跟max的长度不一样 要比较从root开始的increase长度是左边长还是右边长，decrease长度同理
- 注意Math.max只能两个参数 WA忘记传入max了 所以就多写一行

100. Same Tree

Same Tree

- 如果两个都是空 返回true
- 如果两个有一个不空 或者两个都不空但是值不一样 返回false
- 分治看左边和右边是返回true或false 最后&&一下两边结果
- 面经里有一题类似 但是

652. Find Duplicate Subtrees

Find Duplicate Subtrees - LeetCode

- 思想: 边做preorder(根左右) traversal 序列化 边把序列化结果存进map里
- 如果别的地方也序列化出同样的结果 就可以把当前的根节点 放进结果
 - 次数正好为2的时候才放进去 map里存<str, int次数> 判断次数

给定一个二叉树，找出这个二叉树中相同的子结构。

基本思想是这样：对这个二叉树的进行一次遍历，得到每个子结构序列化的结果，保存起来，序列化结果相同的结点，就是需要的结果。

序列化最简单的方式就是转换成字符串，但是需要注意的是不同的遍历方式可能得到相同的序列化结果。比如：

前序：DLR

中序：LDR

后序：LRD

没有左孩子的时候 前序 = 中序

没有右孩子的时候 后序 = 中序

只有根节点的时候 后序 = 前序

为了防止这种情况，null值也需要表示出来，代码中用“#”表示。

需要注意的是，这一题不能用set保存出现过的字符串，因为可能有不止一个，所以要将之前出现过一次的字符串保存到结果列表中

<http://www.jianshu.com/p/a135e7ff2e30>