链滴

【xsong 说算法】第三期：手撕力扣二叉树

作者：xsong
原文链接：https://ld246.com/article/1639638850315
来源网站：链滴

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/bing/20190829.jpg?imageView2/1/w/960/h/540/interlace/1/q/100">
<h3 id="3-1-前言">3.1 前言</h3>
看了很多别人总结的算法的笔记，他们都比较推荐在我们对算法无从下手的时候，去刷二叉树
题，因为二叉树类型的题能够培养我们的框架思维，为我们接下来刷动态规划类型打一个很好的基础
首先声明：该文章的侧重点是帮我们建立二叉树算法的类型体系，以便于我们熟能生巧，在面试中发
的更好，对数据结构不做详细讲解。
<blockquote>
二叉树的遍历：递归思想
</blockquote>
<pre><code class="language-java highlight-chroma"><span c
ass="highlight-cl">//System.out.println(root.val);
<span cla
s="highlight-c1">void <span class="highlight-n
">traverse(TreeNod
 root) <span
class="highlight-o">{
    <span class="h
ghlight-c1">// 前序遍历
<span cla
s="highlight-c1">   //System.out.println(root.val);
<span cla
s="highlight-c1">   traverse<span class="highli
ht-o">(root.</span
left)
    <span class="h
ghlight-c1">// 中序遍历
<span cla
s="highlight-c1">   //System.out.println(root.val);
<span cla
s="highlight-c1">   traverse<span class="highli
ht-o">(root.</span
right)
    <span class="h
ghlight-c1">// 后序遍历
<span cla
s="highlight-c1">   //System.out.println(root.val);
<span cla
s="highlight-c1">}
</code></pre>
<h3 id="3-2-验证二叉搜索树的合法性">3.2 验证二叉搜索树的合法性</h3>
<pre><code class="language-java highlight-chroma"><span c
ass="highlight-cl">boolean
sBST(TreeNode</sp
n> root,<span class
"highlight-n">TreeNode min<span class="highl
ght-o">,TreeNode
ax){
    <span class="h
ghlight-c1">//base case
<span cla
s="highlight-c1">   if<span class="highlight-o"
(root == <s
an class="highlight-kc">null){

        <span class=
highlight-k">return ture<span class="highlight-
">;
    <span class="h
ghlight-o">}

    <span class="h
ghlight-k">if(min</
pan> != null <span
lass="highlight-o">& root<span cla
s="highlight-o">.val <span class="highlight-o
><= min)</span
return false<span cl
ss="highlight-o">;
    <span class="h
ghlight-k">if(max</
pan> != null <span
lass="highlight-o">& root<span cla
s="highlight-o">.val <span class="highlight-o
>>= max)</span
return false<span cl
ss="highlight-o">;
    <span class="h
ghlight-k">return isBST<span class="highlight-
">(root.<sp
n class="highlight-na">left,<span class="highli
ht-n">min,root</sp
n>) & <
pan class="highlight-n">isBST(<span class="hig
light-n">root.right
/span>,root<span cl
ss="highlight-o">,max<span class="highlight-o
>);
<span class="high
ight-o">}
</code></pre>
<h3 id="3-3-带你刷-LeetCode-二叉树">3.3 带你刷 LeetCode 二叉树</h3>
<blockquote>
下面的题都是我这周做的，然后写了比较详细的题解，东西基本都在题解里面了。
</blockquote>
<table>
<thead>
<tr>
<th align="center">题目链接</th>
<th align="center">我的题解</th>
</tr>
</thead>
<tbody>
<tr>
<td align="center"><a href="https://ld246.com/forward?goto=https%3A%2F%2Fleetcode-cn
com%2Fproblems%2Finvert-binary-tree%2F" target="_blank" rel="nofollow ugc">【力扣】22
.翻转二叉树</td>
<td align="center"><a href="https://ld246.com/forward?goto=https%3A%2F%2Fleetcode-cn
com%2Fproblems%2Finvert-binary-tree%2Fsolution%2Fer-cha-shu-fan-zhuan-di-gui-chu-li-

y-xs-hhun%2F" target="_blank" rel="nofollow ugc">我的题解</td>
</tr>
<tr>
<td align="center">【力扣】226.翻转二叉树</td>
<td align="center">我的题解</td>
</tr>
<tr>
<td align="center">【力扣】114. 二叉树展开为链表</td>
<td align="center">我的题解</td>
</tr>
<tr>
<td align="center">【力】654. 最大二叉树</td>
<td align="center">我的题解</td>
</tr>
<tr>
<td align="center">【力扣】105. 从前序与中序遍历序列构造二叉树</td>
<td align="center">我的题解</td>
</tr>
<tr>
<td align="center">【力扣】106. 从中序与后序遍历序列构造二叉树</td>
<td align="center"><a href="https://ld246.com/forward?goto=https%3A%2F%2Fleetcode-cn.com%2Fproblems%2Fconstruct-binary-tree-from-inorder-and-postorder-traversal%2Fsolution%2Flao-fang-fa-di-gui-gou-zao-er-cha-shu-by-v4r2%2F" target="_blank" rel="nofollow ugc"我的题解</td>
</tr>
<tr>
<td align="center">【力扣】230. 二叉搜索树中第 K 小的元素</td>
<td align="center">我的题解</td>
</tr>
<tr>
<td align="center"><a href="https://ld246.com/forward?goto=https%3A%2F%2Fleetcode-cn

com%2Fproblems%2Fconvert-bst-to-greater-tree%2F" target="_blank" rel="nofollow ugc">力扣】538. 把二叉搜索树转换为累加树</td>
<td align="center">我的题解</td>
</tr>
</tbody>
</table>
至此，二叉树就讲到这里，如果你把上面的题都做完，并且认真总结的话，相信你一定会有很的收获。上面的题是二叉树的经典题型，如果都掌握的话，建议去深入学习 BST 的一些操作，比如除操作，如何在删除之后依旧维持他的稳定？等等...这里就不再过多讨论了。有什么问题可以私信我者提出来，大家共同进步！加油！
ps:由于临近期末，这周我就更新这一期，计网还有其他专业课该好好复习了，不能挂科。