BST

#G面试准备

Inorder Successor in BST

Inorder Successor in BST

- 解释: Inorder Successor in Binary Search Tree - GeeksforGeeks

- 返回的是inorder traversal的下一个节点
- 如果他的右树不空 就返回右树的最小值 因为是BST
  - 注意是minValue(p.right) 不是minValue(p)
- 否则从root开始遍历
- 初始化一个successor = null
- BST向左向右遍历
  - 向左: successor是他的root
    - successor = root
    - root = root.left
  - 向右: successor不变 root = root.right
  - 相等 直接退出循环
- 返回successor

230. Kth Smallest Element in a BST

Kth Smallest Element in a BST - LeetCode

- 和上一题很像 其实也是inorder traversal
- 边用循环+stack做inorder traversal边计数

1. 先循环往左遍历 把经过的node都加进stack
2. while stack不空
3. pop stack栈顶的节点
   - 同时k要减一
   - pop的顺序就是数字的顺序 pop k次就能得到第k小的数字
   - 如果当前k减到了0 就输出当前的node
   - 否则继续 当前节点current还要用来看有没有右子树
   - 如果右子树不空 就去到current.right, 并且往左循环遍历节点 加进stack (也可能只加进一个current.right)

- 总结: 整个inorder traversal就是一个不停地往左 往左走不动了就去到最左节点的右子树 看能

不能继续不停地往左的过程 期间所有经过的node都会加入stack 然后pop k次就能得到第k个数

Leetcode Kth Smallest Element in a BST 二叉搜索树第k小节点 - Ethan Li 的技术专栏 - SegmentFault

> 这题的难点其实在于Follow Up：如果我们频繁的操作该树，并且频繁的调用kth函数，有什么优化方法使时间复杂度降低至O(h)？h是树的高度。根据提示，我们可以在TreeNode中加入一个rank成员，这个变量记录的是该节点的左子树中节点的个数，其实就是有多少个节点比该节点小。这样我们就可以用二叉树搜索的方法来解决这个问题了。这个添加rank的操作可以在建树的时候一起完成。