

1. 定义 ABST :

1)left:左孩子 2)right:右孩子 3)parent:父节点 4)element:节点值 5)size:节点数

2)结果测试:

对于右侧的 BST,左侧为测试情况(1=插入 2=删除 3=kselect 4=find(z,k) 5=print)



2. 添加 size 属性之后

k-select :

每次只需要在 $O(1)$ 时间内检查当前节点是否符合

1)节点为 kth 节点,直接返回

2)节点不符合,递归在左子树或者右子树中寻找

易知,时间复杂度为 $O(\log n)$,相比于普通 BST 遍历查找($O(n)$)复杂度有所提升

比 z 大的第 k 个数:

易知为中序遍历找到第一个大于 z 的值 v 及其所排的次序 $th(v)$,时间复杂度为 $O(n)$,调用

$select(th(v)+k)$ 即可得到大于 z 的第 k 个值,时间复杂度为 $O(n)$,与普通 bst 无差别