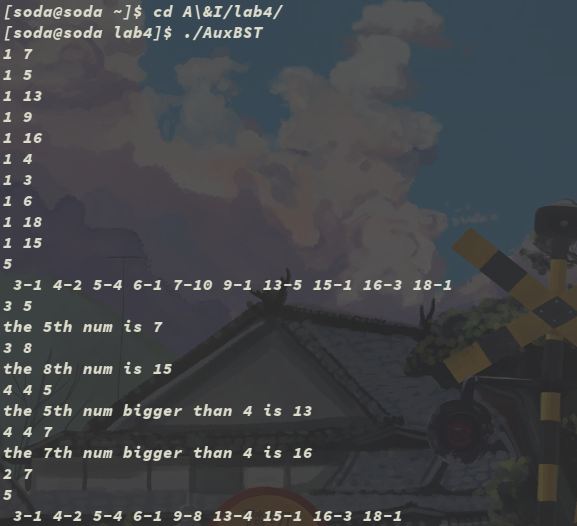
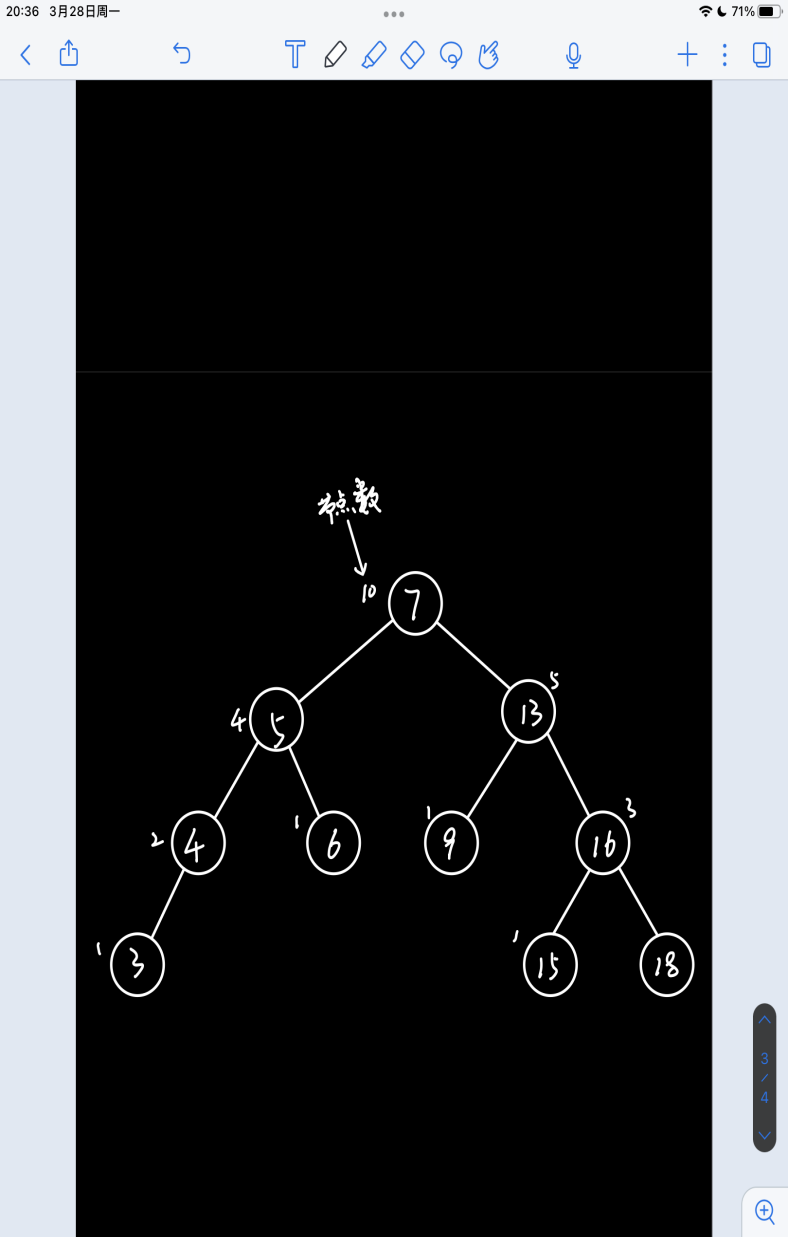
**1.定义ABST：**

**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(logn),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**