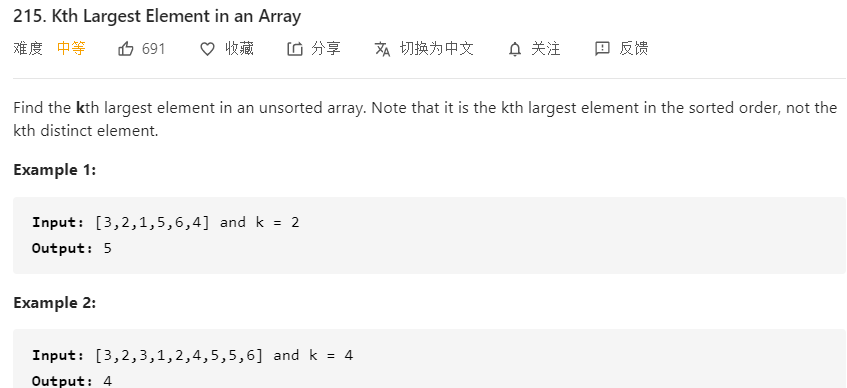
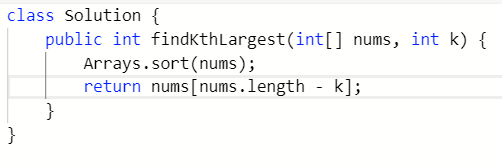
排序 sort

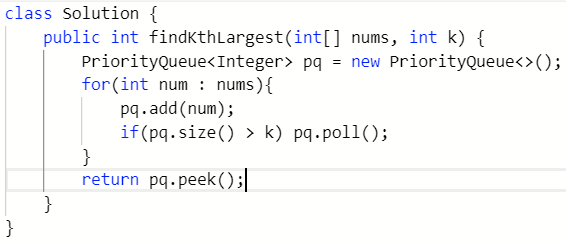
1.



解法1：调库函数 时间NlogN

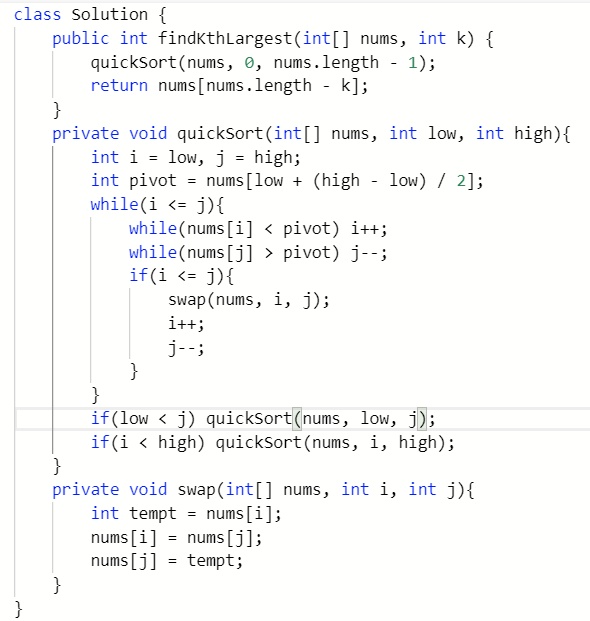


解法2：用优先对立priority queue，每次add都会进行从小到大排序----确保add到正确的位置，poll和peek会把最前面也就是最小的元素返回 时间NlogN



解法3：递归快速排序 时间O(NlogN) 空间O(1)

pivot每次设置为nums的mid数，之后把i和j分别运动到pivot的右边和左边如果i<=j则互换i和j上的两个元素，再进行递归

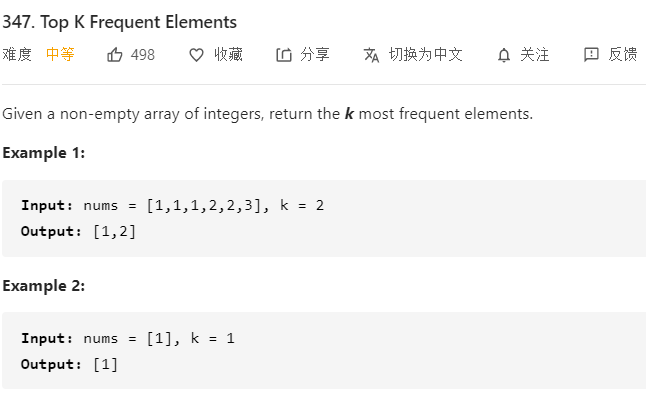


解法4：快速选择 时间O(N) 空间O(1)

比快速排序快因为每次partition完自己左边都是比自己小的数但没有被sort，右边同理。这样看partition完的是不是nums.length - k就行



2.



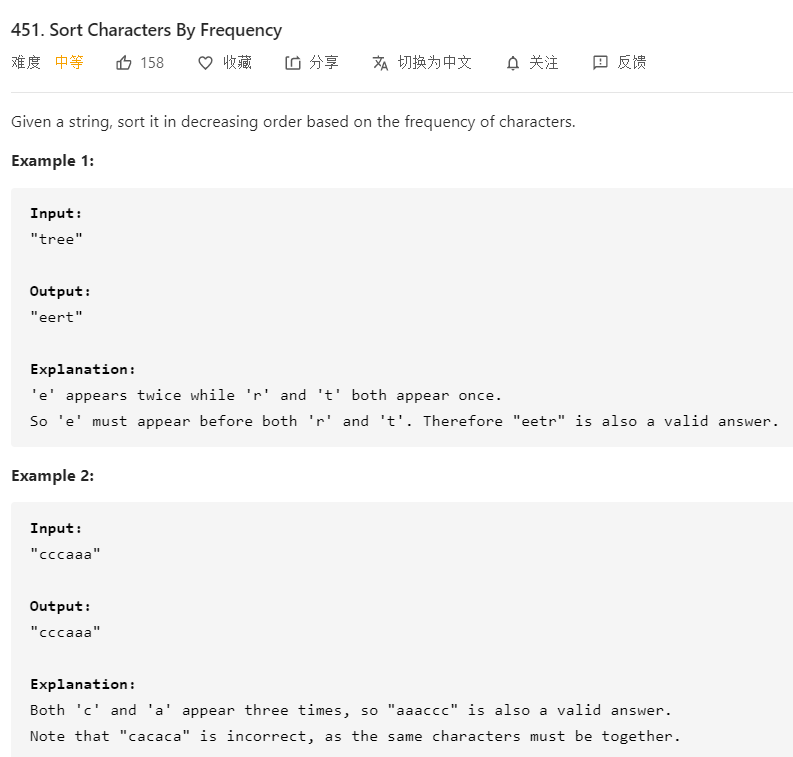
解法：桶排序：用哈希表去记录每个数出现的次数，再用一个ArrayList[]去放根据出现次数放他的key，比如1出现2次，则把1放在index 2。

再用一个数组放返回的元素----用loop去遍历上面的ArrayList[]，如果里面有元素则要根据返回的数组剩下的位置去放。

for loop如果元素放满了或i<0则结束



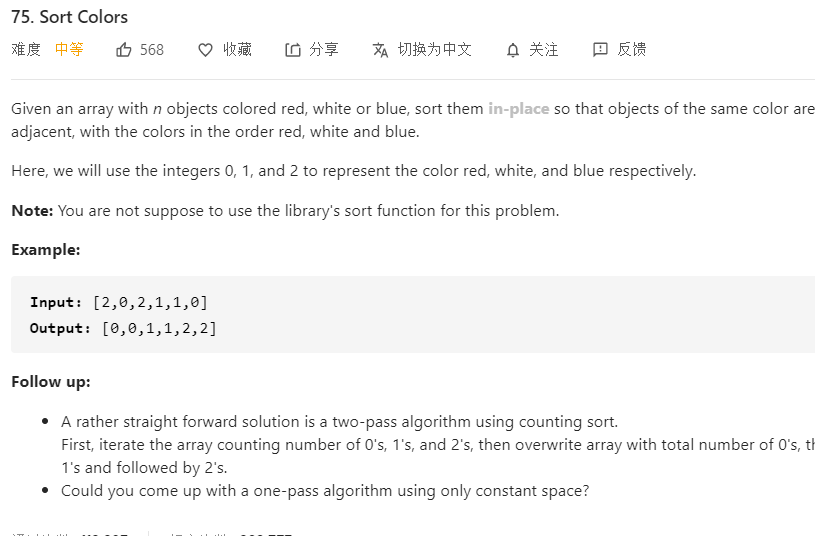
3.



解法：先把每个字母的次数放到hashMap，再根据出现的次数放每个次数，用StringBuild从buckets的后面index往前放里面的字母



4.荷兰国旗问题



解法：三路快排题 用三个指针一个记录左边，一个用于遍历，遍历到0则把元素和左指针的元素的交换，把0放到左边，遇到2则把元素和右指针换，把2换到右边，但要注意的是换完之后遍历的指针要往左退一步这样重新检测一个换过来的元素是0还是1.

当遇到右指针时结束遍历

