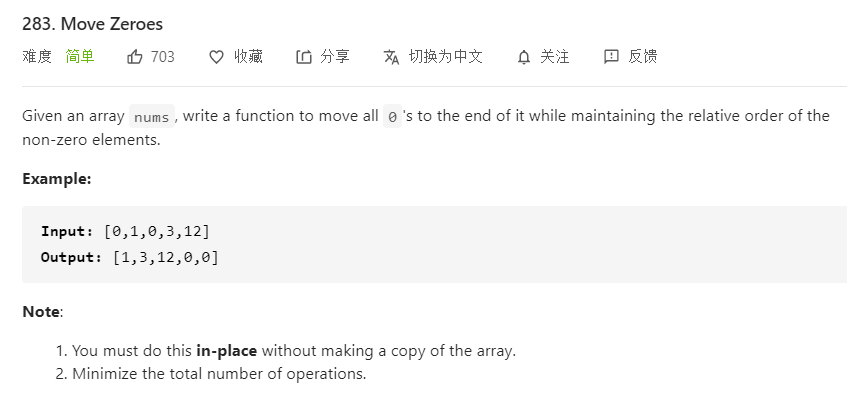
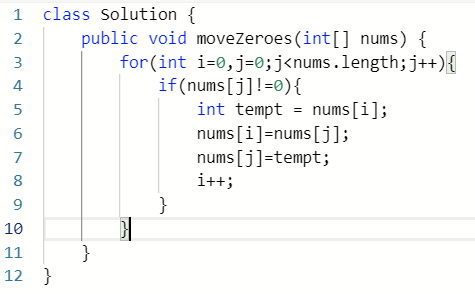
Array

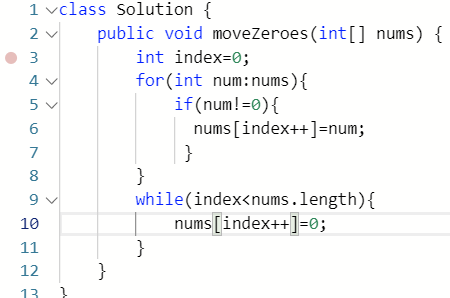
1.



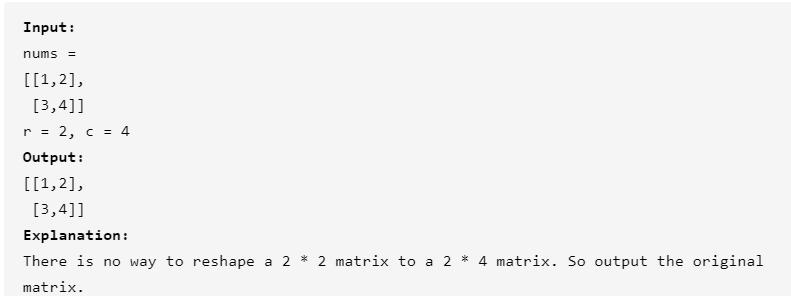
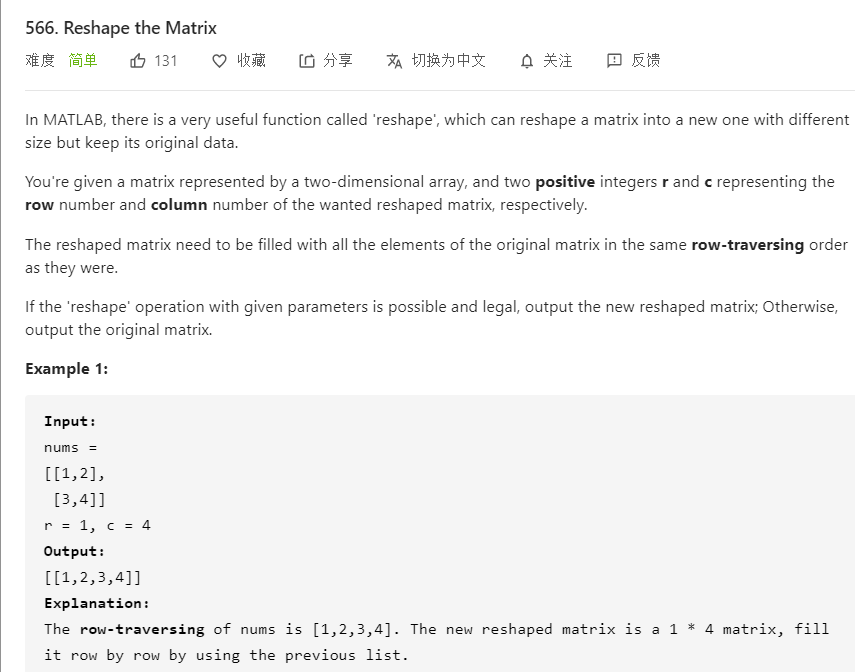
解法1：双指针 互换两个指针的元素如果第二个指针的元素是0，不然第二个指针往后移



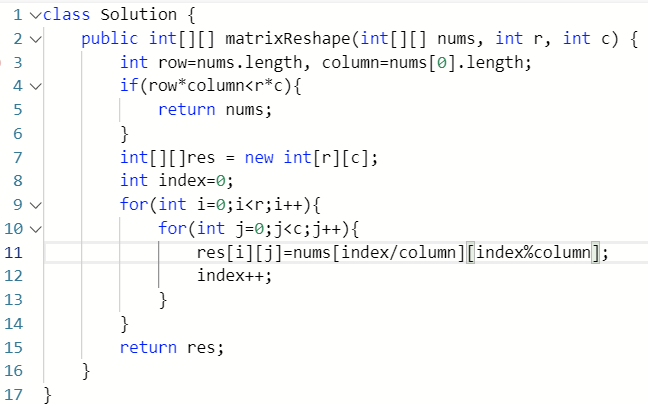
解法2：把数组里不是0的元素从index 0开始放，之后在再补满0



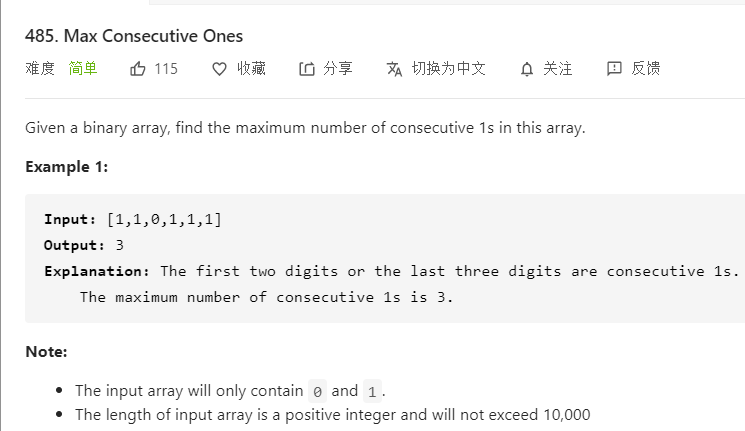
2.



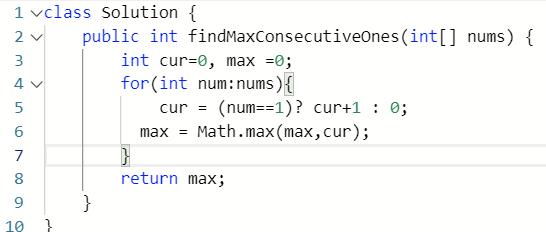
解法：先判断nums里的数字总数是不是不能组成r\*c的矩阵，返回nums



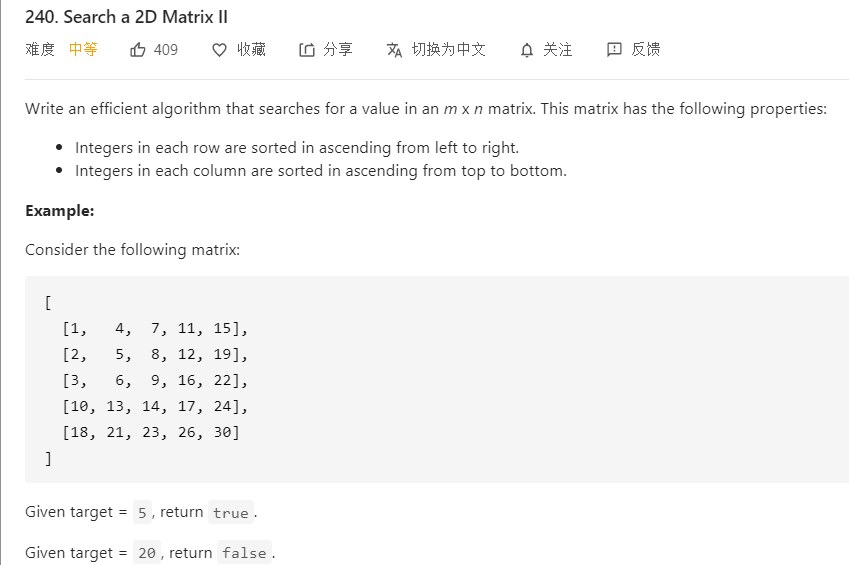
3.



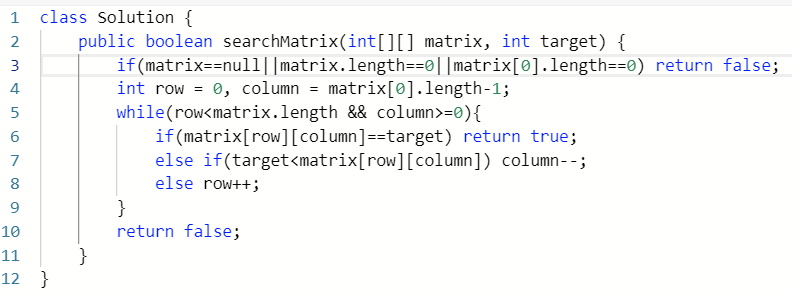
解法：一个指针，当没0就++，遇到0就变回0，return ++最长的数



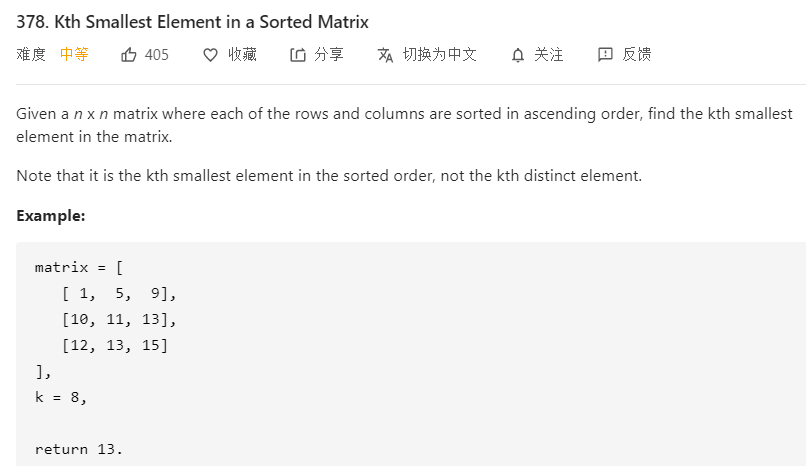
4.



解法：每次查找去找每行的最后一个元素，如果target比这个数小则往前一个搜索，否则直接跳到下一行



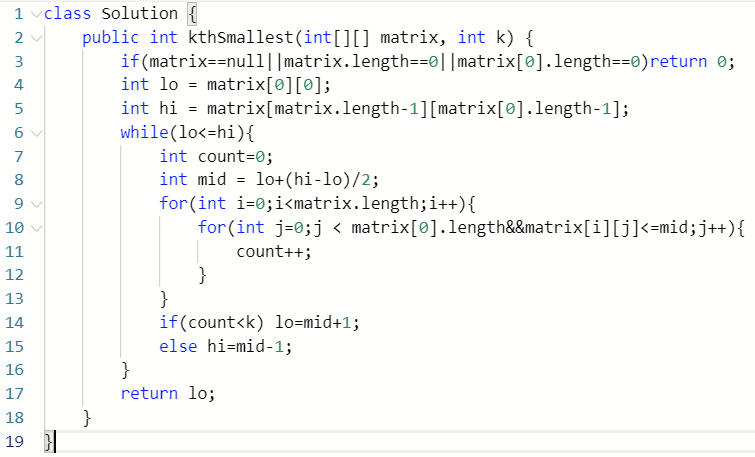
5.



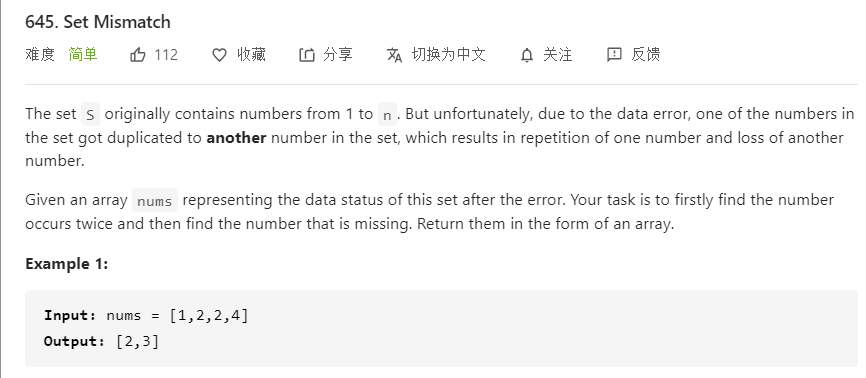
解法：用二分查找法 先确定low和high也就是数组里最小和最大的数，mid=low+(high-low)/2每次都从数组的第一个元素开始找，如果第一行遇到比mid大的数则从第二行找，每遇到一个小于or等于mid的数 count++，如果count小于k，说明mid要向右大个四分之一，

反之mid要向左小四分之一，当low>high时候停止

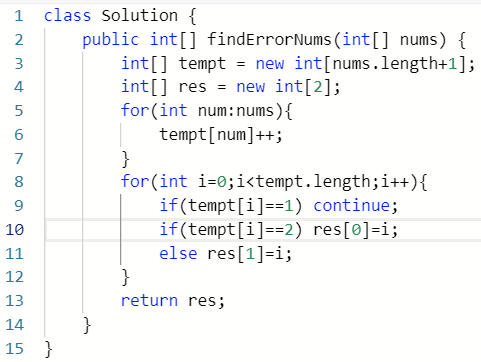
相当于把2D数组里的数从小到大排序，每次找count是不是等于k 如果等于则此时的lo就是第K的值



6.



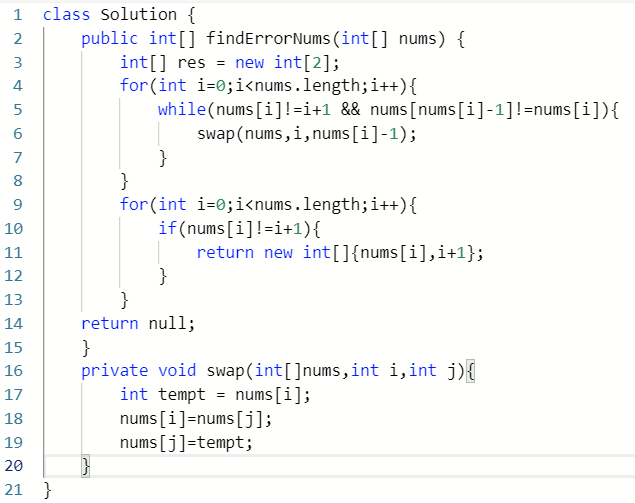
解法1：数组桶 在一个新的数组里放上给的数组的元素作为index，每放1次新数组对应位置+1 如果对应位置是2则放i到返回的数组 是1则继续 else把i放到第返回数组的第二位，因为是没出现过的



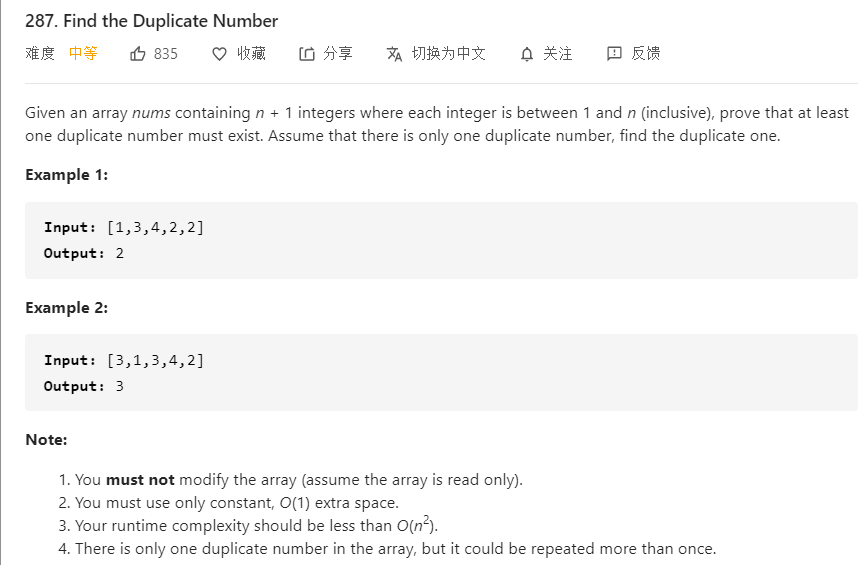
解法2：先交换数组元素使元素在正确的index上 比如322变成223

先检查index0是不是1再检查index0的数字所要在的index是不是是正确的数（index2应该是3）补满足其中一个就交换位置

返回的是第一个不等于i+1的元素和原本应该在这个位置的元素 也就是i+1

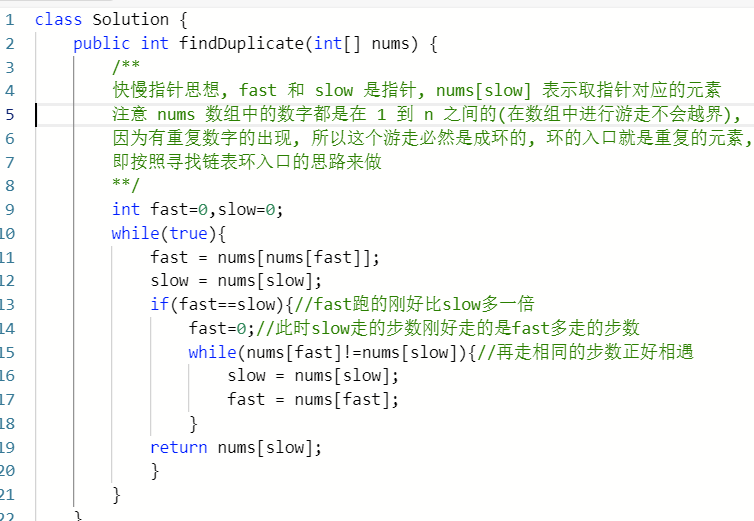


7.



解法1：用快慢指针方法

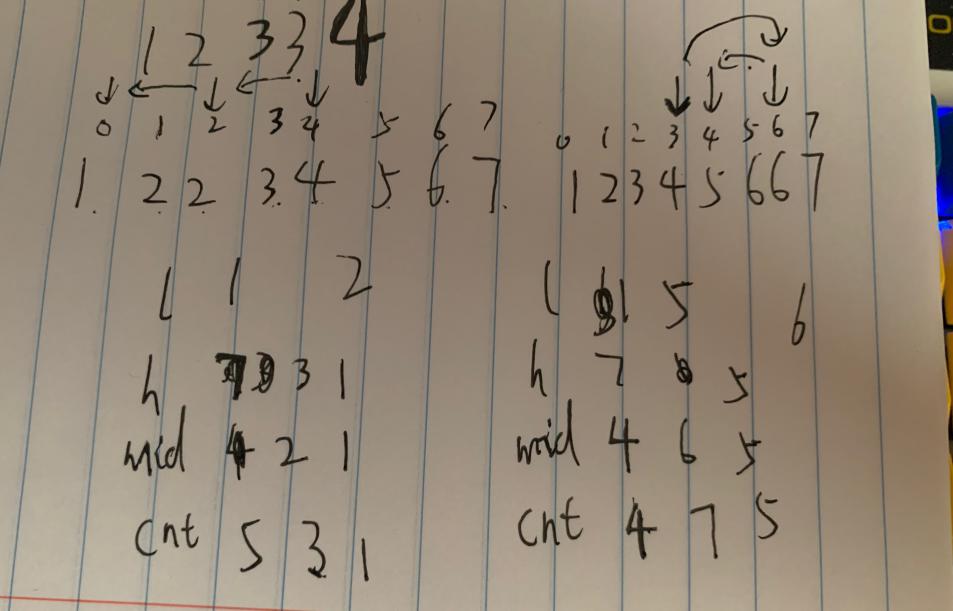
fast置0是因为

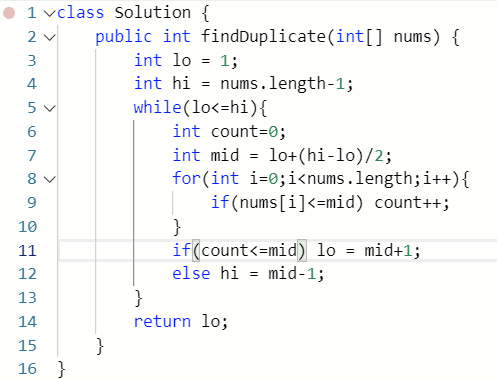


解法2：用二分查找法 因为数组是从小到大连续的数 比如12234 234556

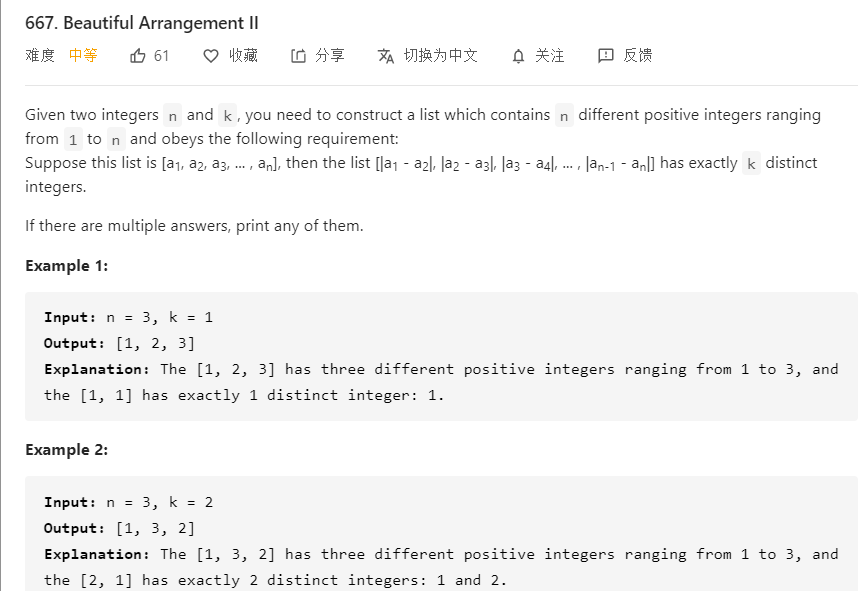
如果重复的数在中间数的左边则开始往左边找 否则往右

low和high指针因为有重复元素所以肯定会low到high右边

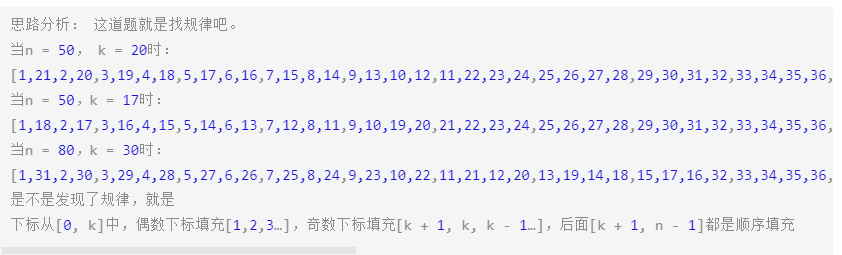


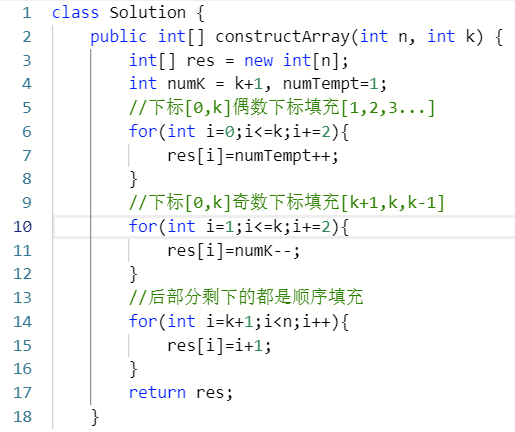


8.

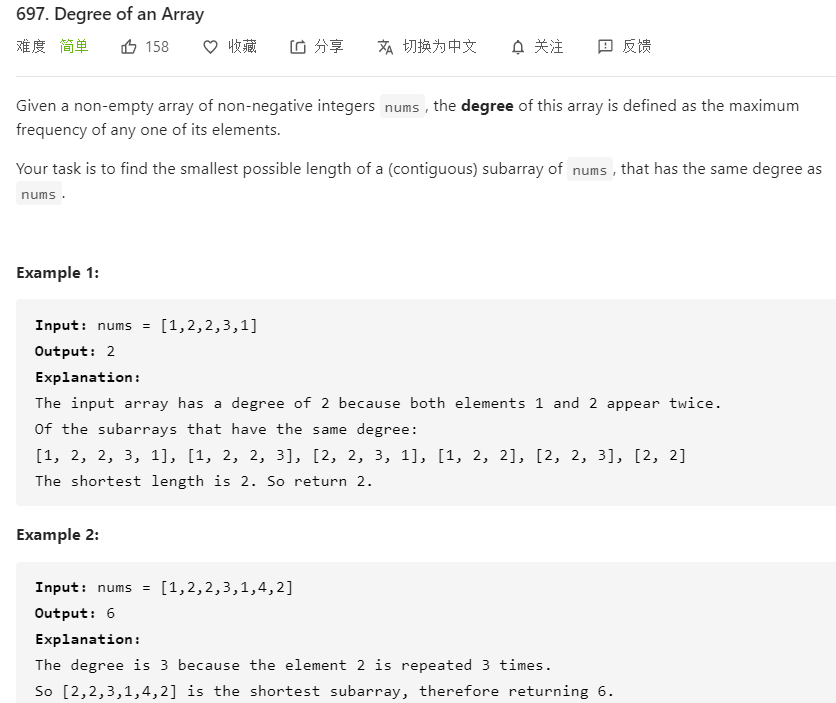


解法：





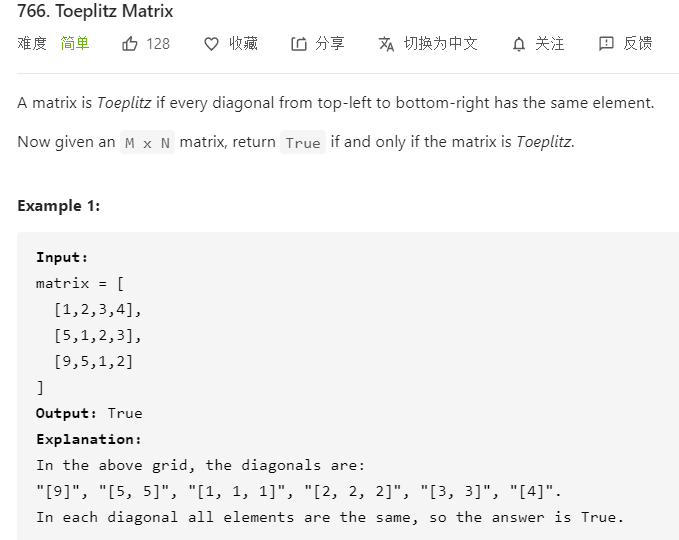
9.



解法：用三个哈希map，两个记录元素开始的位置，一个记录每个元素出现的次数

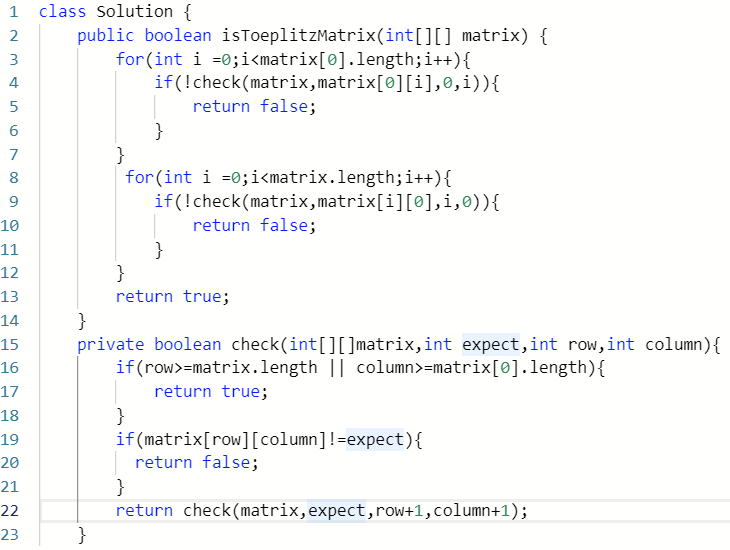


10.

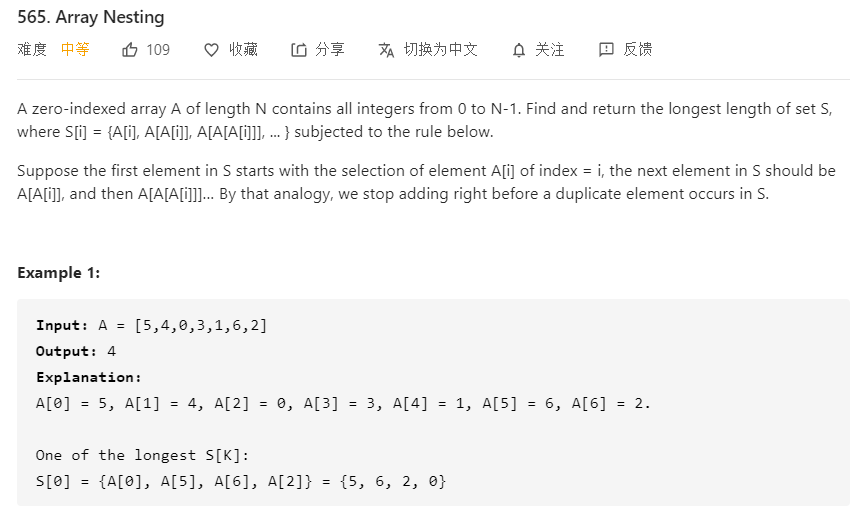


解法：先竖着检查第一行每个元素和他row+1和column+1的值一不一样

再横着检查第一列每个元素和他row+1和column+1的值一不一样

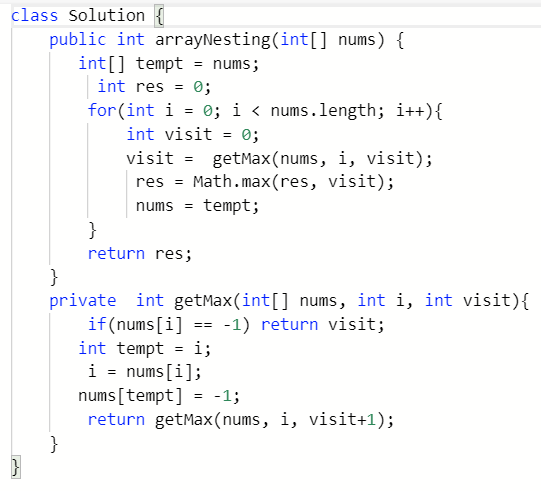


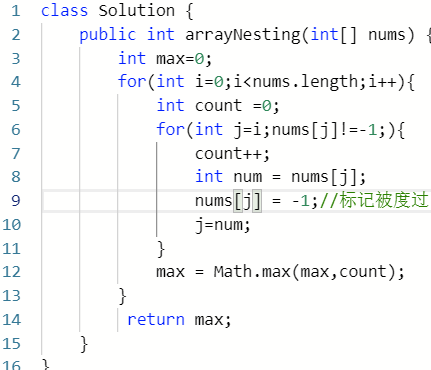
11.



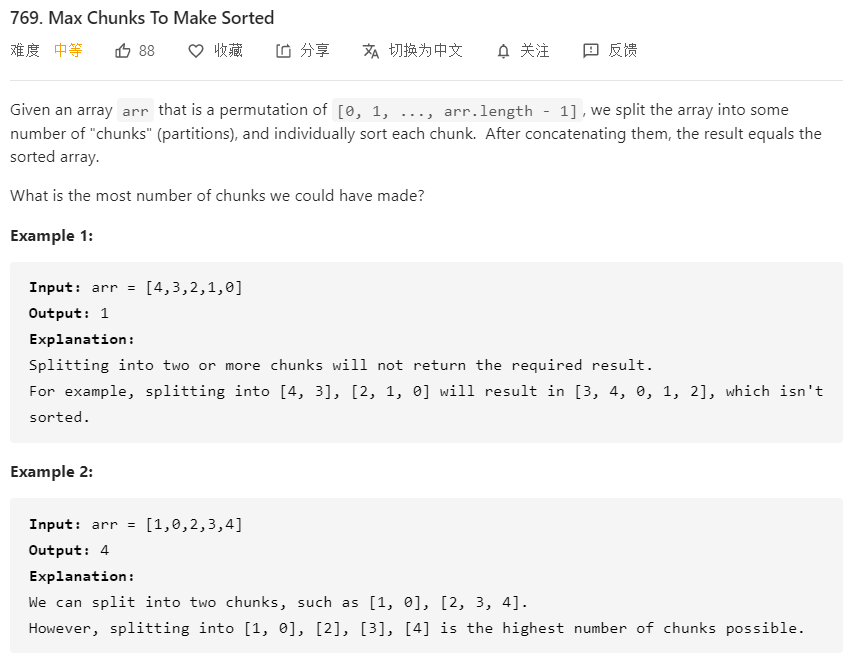
解法：对于每个元素，都去按照题目里的方法去做 看看哪个元素能得到最长的数组

每次读过用-1来标记

or



12.



解法：当遍历到第i个元素的时候，如果可以分块，则前i个位置的最大值一定等于i

否则，一定有比i小的数在后面，后面的快如果排序后一定不是升序

