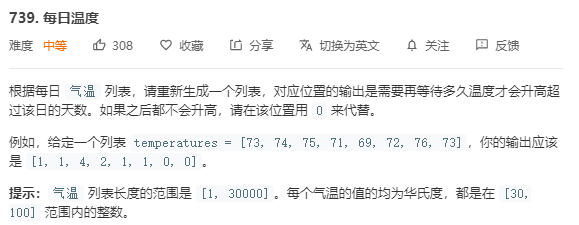
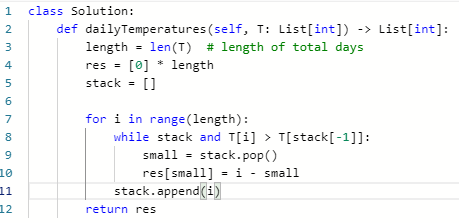




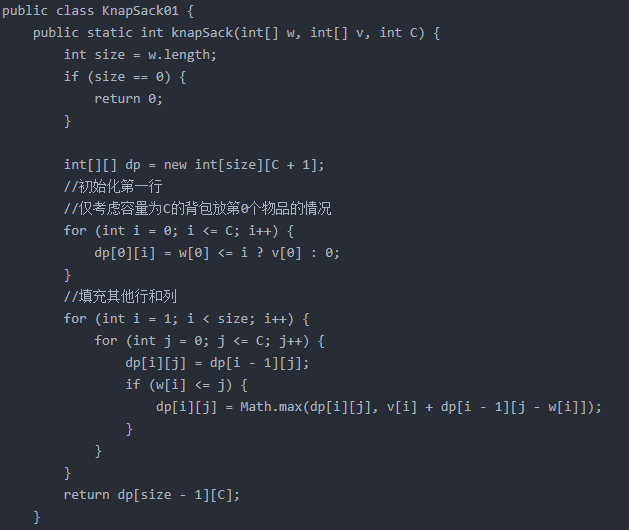
找出数组中每个数右边第一个比它大的元素--时间复杂度o(n)单调栈解法

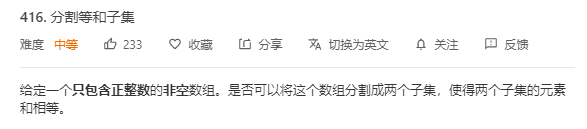
<https://blog.csdn.net/smileiam/article/details/88732245?depth_1-utm_source=distribute.pc_relevant.none-task&utm_source=distribute.pc_relevant.none-task>

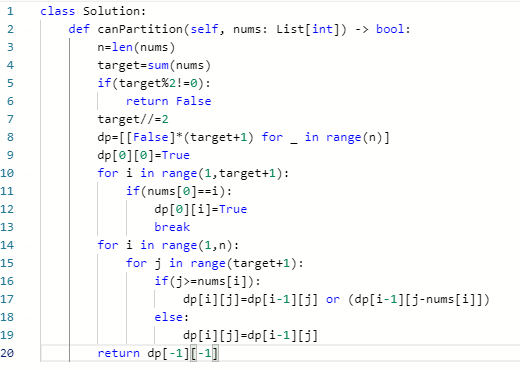


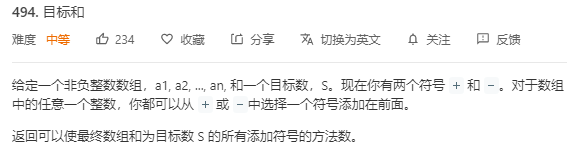


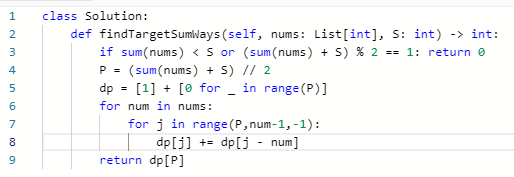
0-1背包问题

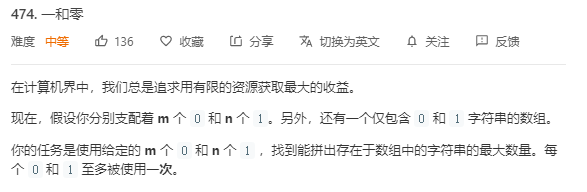


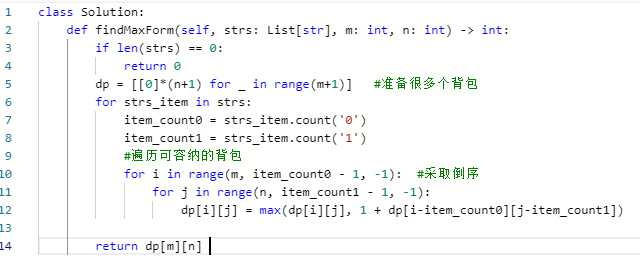




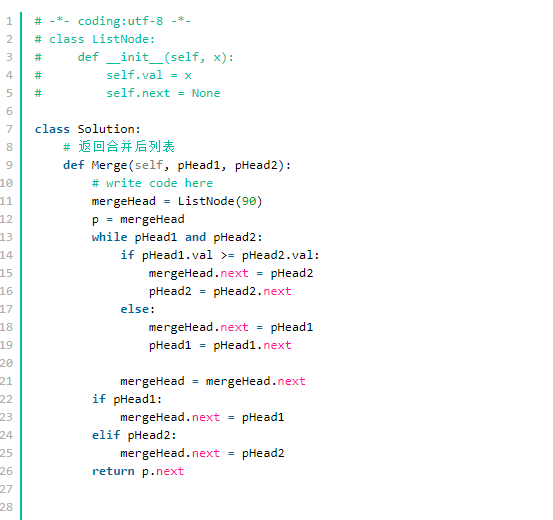




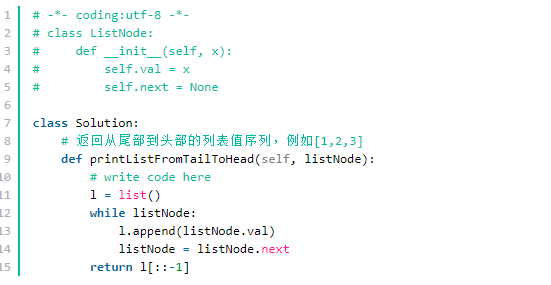




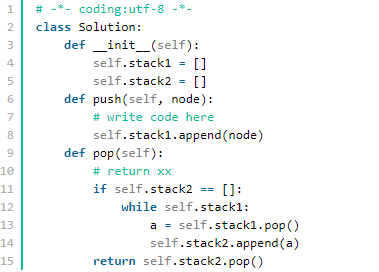
合并两个有序链表：



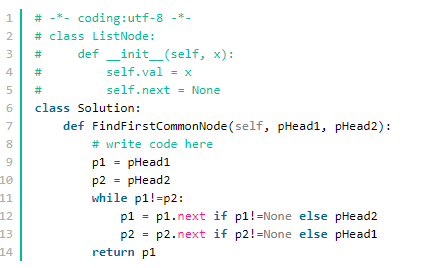
输入一个链表，按链表从尾到头的顺序返回一个ArrayList。



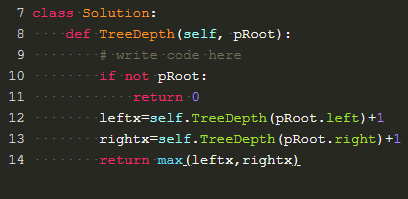
用两个栈来实现一个队列，完成队列的Push和Pop操作。



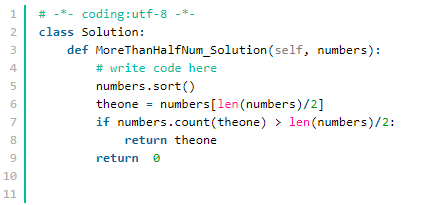
输入两个链表，找出它们的第一个公共结点。

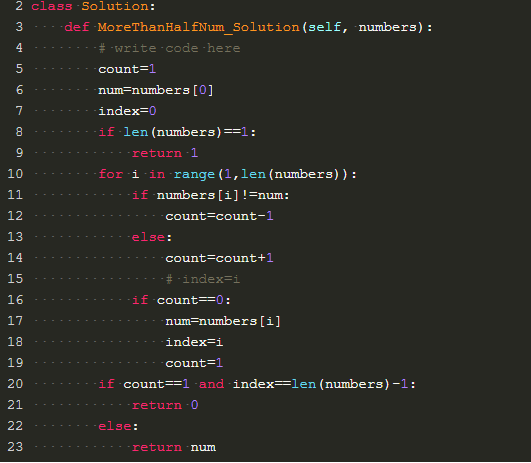


二叉树深度：

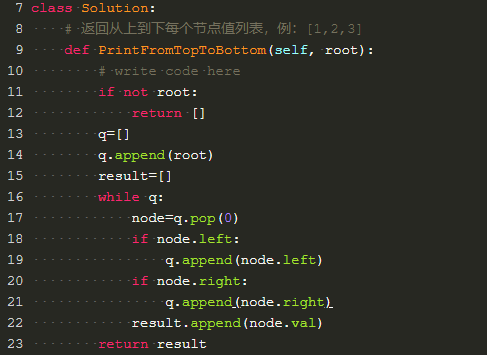


数组中有一个数字出现的次数超过数组长度的一半，请找出这个数字。

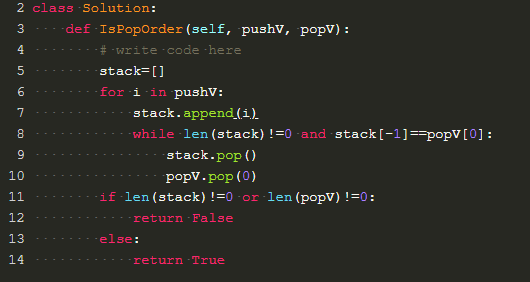




从上往下打印出二叉树的每个节点，同层节点从左至右打印。

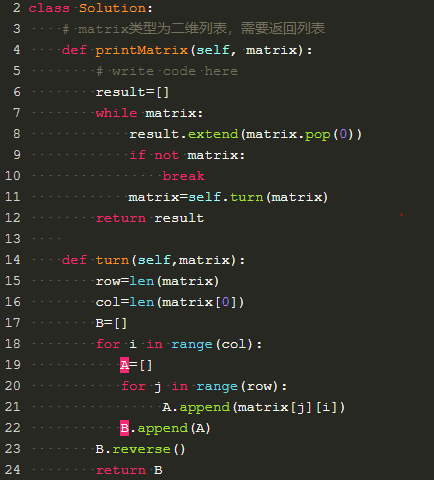


输入两个整数序列，第一个序列表示栈的压入顺序，请判断第二个序列是否可能为该栈的弹出顺序。



输入一个矩阵，按照从外向里以顺时针的顺序依次打印出每一个数字。

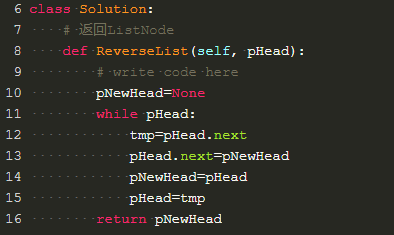
（List extend()方法没有返回值！！！）



操作给定的二叉树，将其变换为源二叉树的镜像。

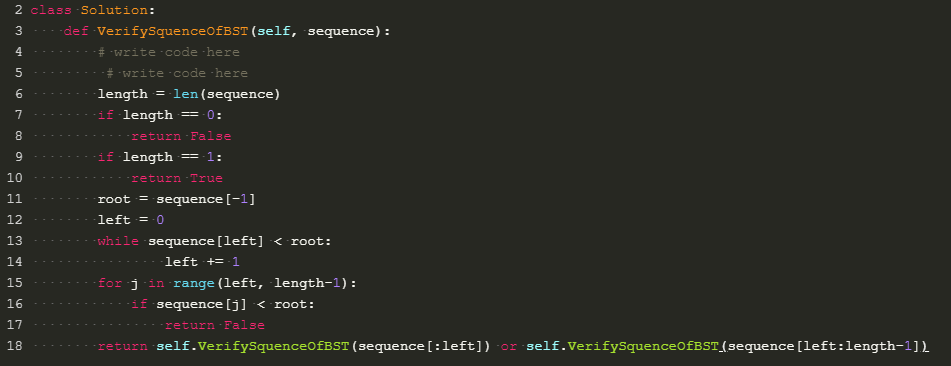


输入一个链表，反转链表后，输出新链表的表头。

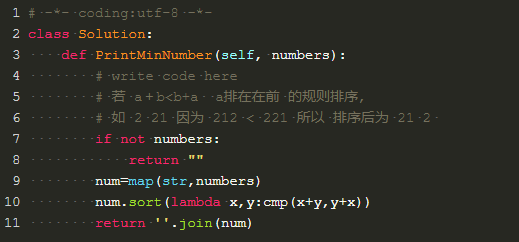


输入一个整数数组，判断该数组是不是某二叉搜索树的后序遍历的结果。如果是则输出Yes,否则输出No。假设输入的数组的任意两个数字都互不相同。

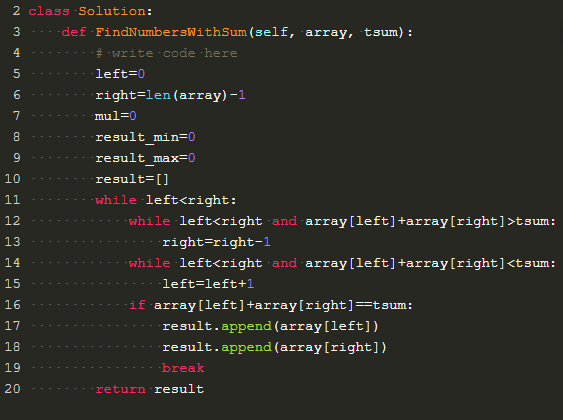
后序遍历、的序列中，最后一个数字是树的根节点 ，数组中前面的数字可以分为两部分：第一部分是左子树节点的值，都比根节点的值小；第二部分是右子树节点的值，都比根节点 的值大，后面用递归分别判断前后两部分是否符合以上原则。



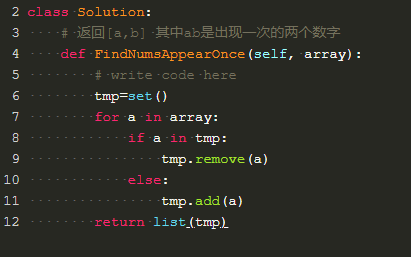
输入一个正整数数组，把数组里所有数字拼接起来排成一个数，打印能拼接出的所有数字中最小的一个。



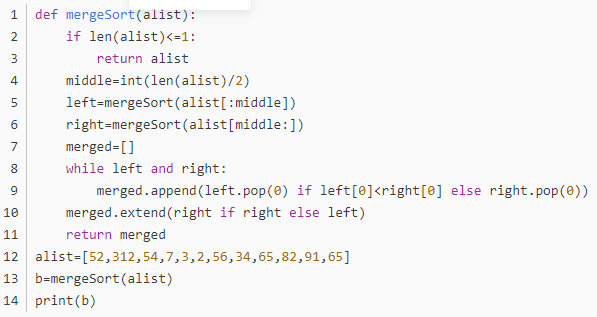
输入一个递增排序的数组和一个数字S，在数组中查找两个数，使得他们的和正好是S，如果有多对数字的和等于S，输出两个数的乘积最小的。



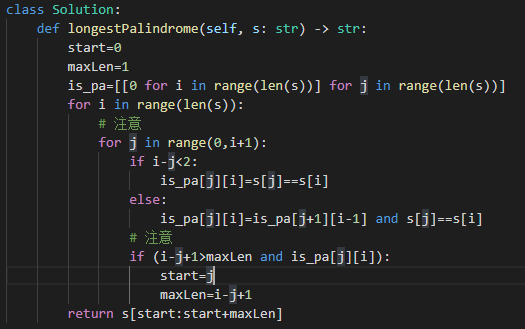
一个整型数组里除了两个数字之外，其他的数字都出现了两次。请写程序找出这两个只出现一次的数字。

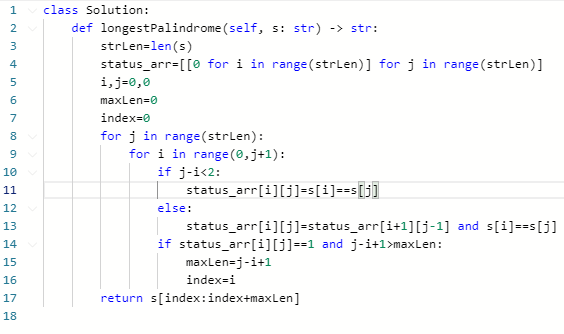


归并排序

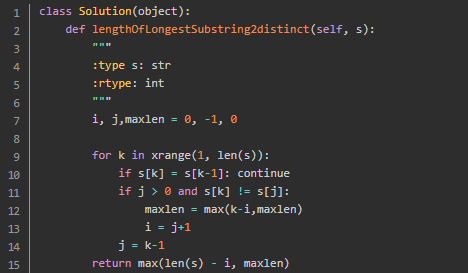


最长回文子串：

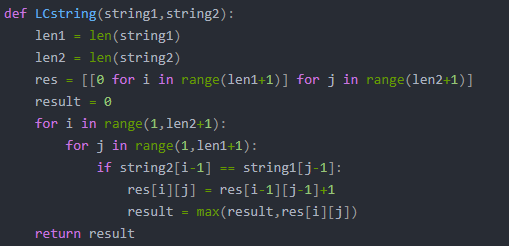




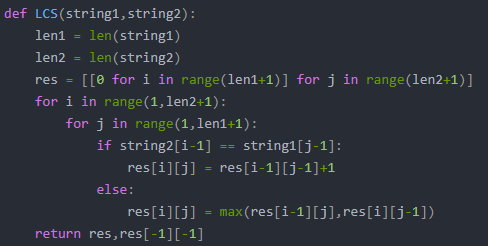
给定字符串，寻找最多包含两个重复字符的最长子字符串。



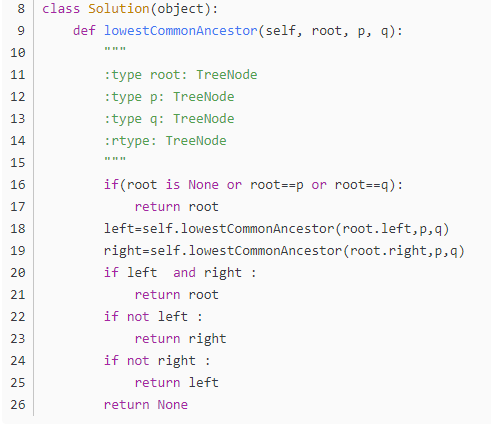
最长公共子串



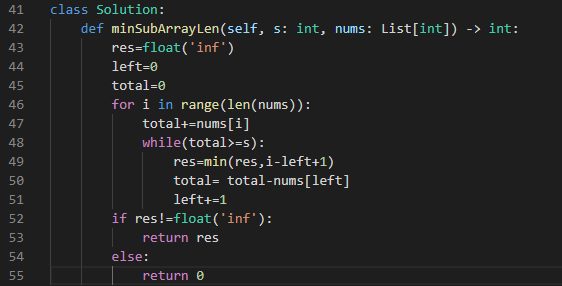
最长公共子序列



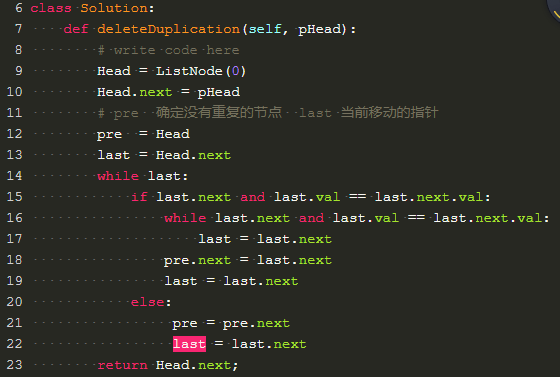
二叉树的最近公共祖先



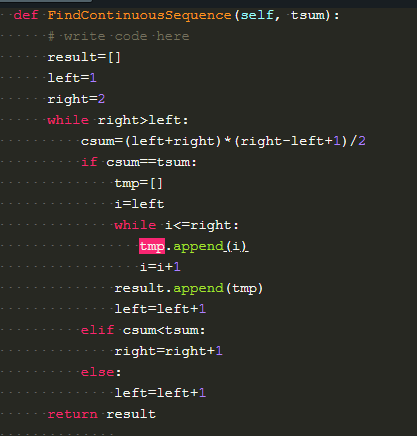
给定一个含有 n 个正整数的数组和一个正整数s ，找出该数组中满足其和 ≥ s 的长度最小的连续子数组。如果不存在符合条件的连续子数组，返回 0。



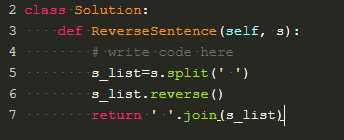
在一个排序的链表中，存在重复的结点，请删除该链表中重复的结点，重复的结点不保留，返回链表头指针。



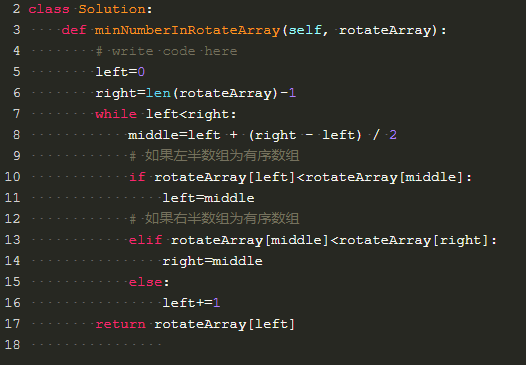
输出所有和为S的连续正数序列。序列内按照从小至大的顺序，序列间按照开始数字从小到大的顺序。

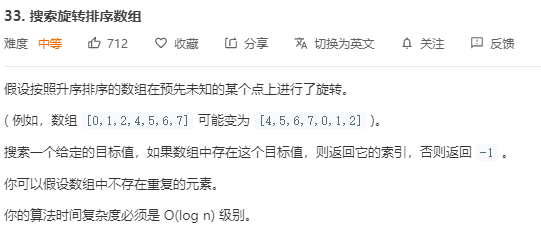


句子翻转：



把一个数组最开始的若干个元素搬到数组的末尾，我们称之为数组的旋转。  
输入一个非递减排序的数组的一个旋转，输出旋转数组的最小元素。  
例如数组{3,4,5,1,2}为{1,2,3,4,5}的一个旋转，该数组的最小值为1。



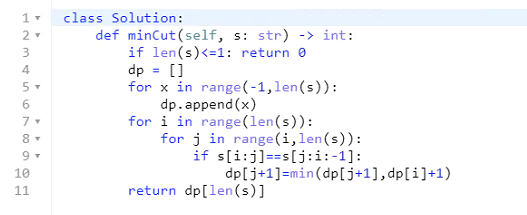




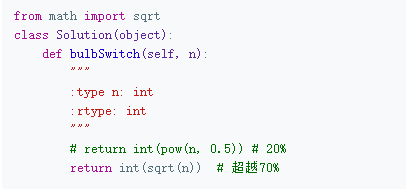
# 连续子数组的最大和

# 

使每个子串都是回文的最小分割次数。



初始时有 *n*个灯泡关闭。 第 1 轮，你打开所有的灯泡。 第 2 轮，每两个灯泡你关闭一次。 第 3 轮，每三个灯泡切换一次开关（如果关闭则开启，如果开启则关闭）。第 *i* 轮，每 *i*个灯泡切换一次开关。 对于第 *n*轮，你只切换最后一个灯泡的开关。 找出 *n*轮后有多少个亮着的灯泡。



<https://leetcode-cn.com/problems/sum-of-subarray-minimums/>

100层楼2个鸡蛋，如何得知鸡蛋能承受几层的撞击？

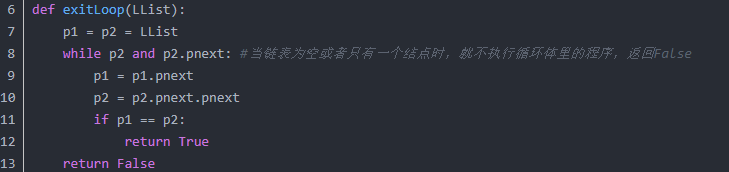
<https://www.cnblogs.com/yangai/p/5391533.html>

<https://blog.csdn.net/nirendao/article/details/80953827>

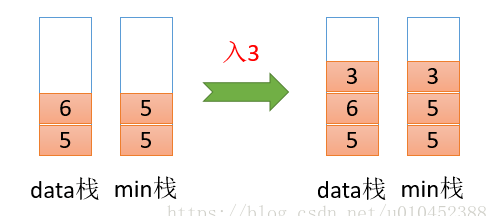
<https://www.bilibili.com/video/av96214853>

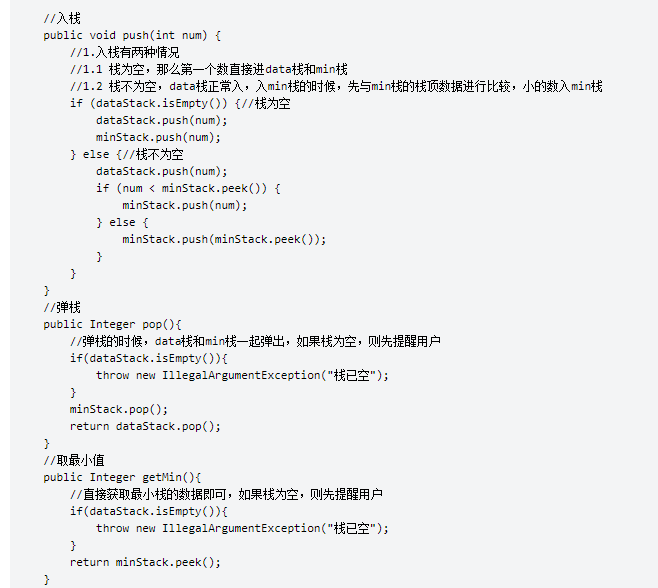
14

判断一个链表是否存在环？



实现一个特殊的栈，在实现栈的基本功能的基础上，再实现返回栈中最小元素的操作。





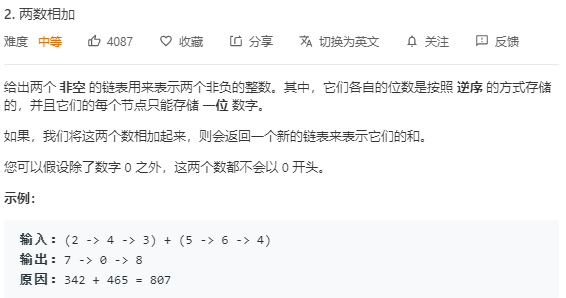
给定一个正整数 n，将其拆分为至少两个正整数的和，并使这些整数的乘积最大化。 返回你可以获得的最大乘积。

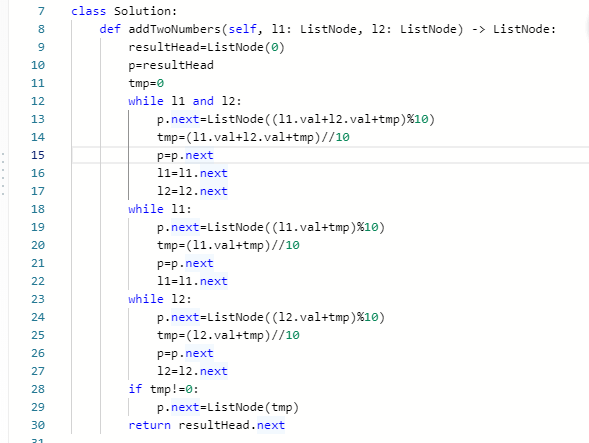
通过上面的例子你会发现这样的规律：想要乘积最大，那么一定要将3作为基底，并且<=4的余数不再进行拆分。

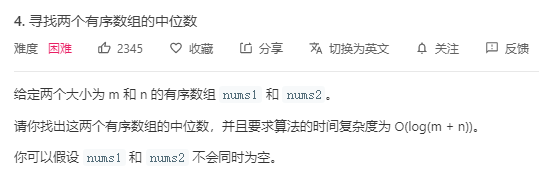
<https://blog.csdn.net/qq_17550379/article/details/82811007>

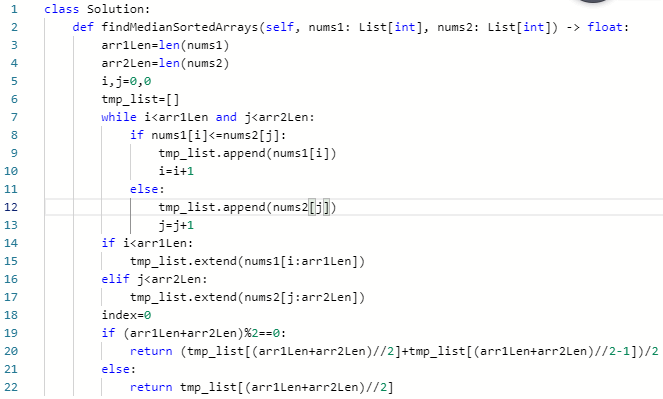
求比n小的一个数，使其各位数的乘积最大

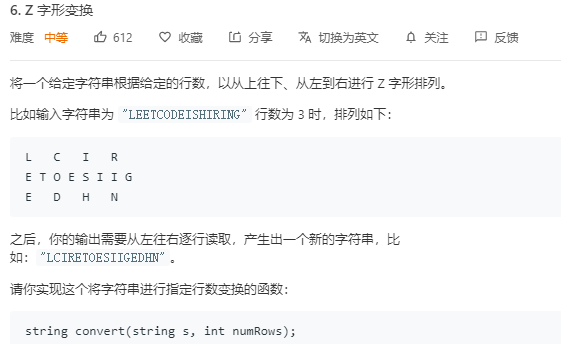
<https://blog.csdn.net/altoer/article/details/89074847>

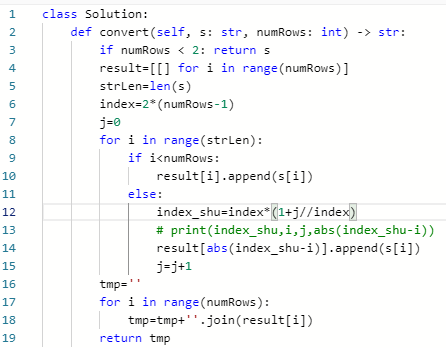




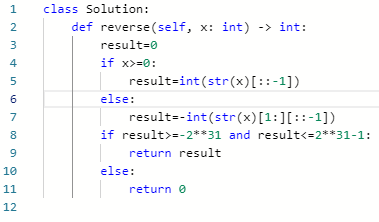






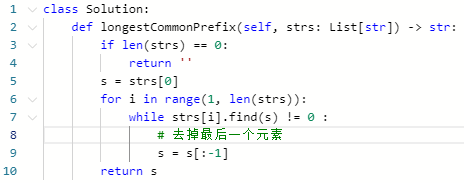


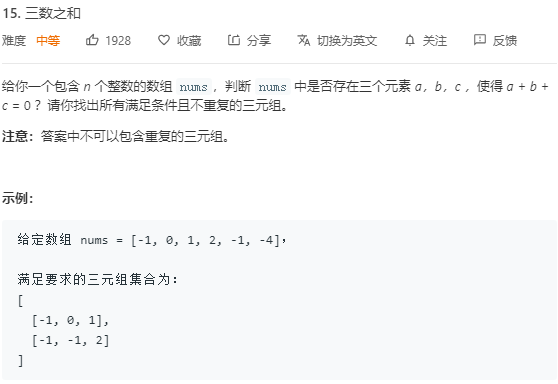


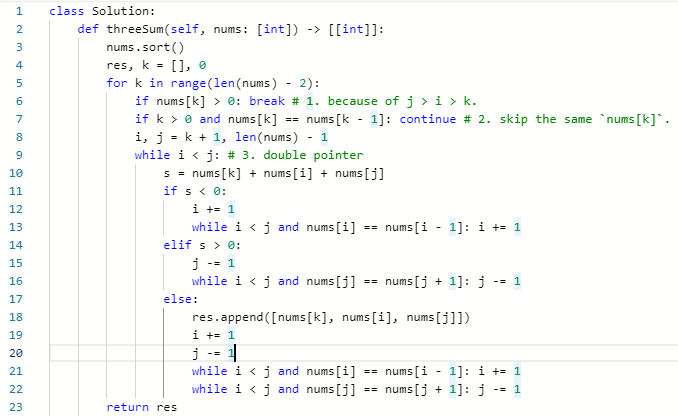


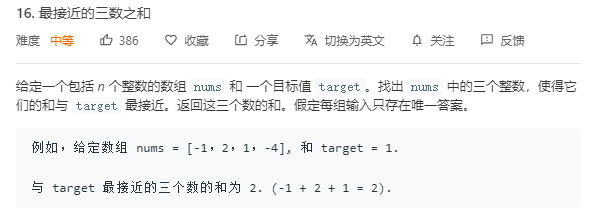


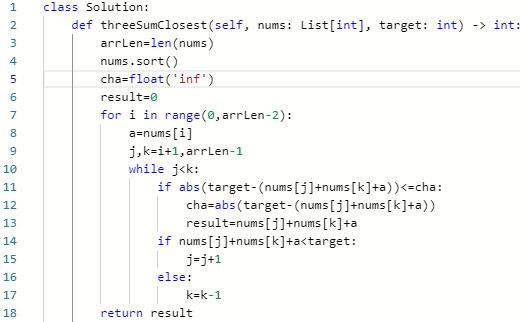
<https://www.runoob.com/python/att-string-find.html>

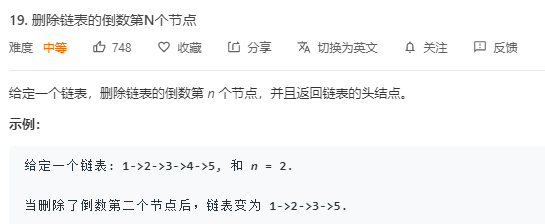


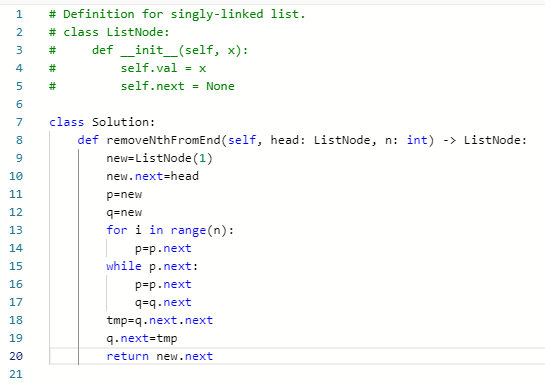


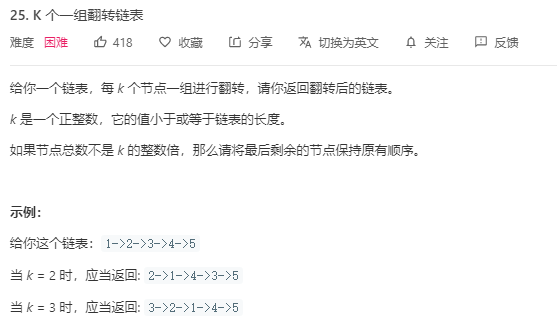




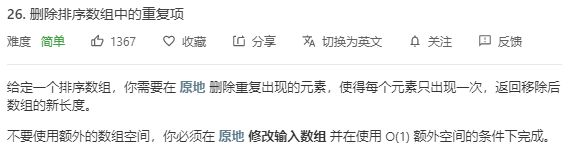


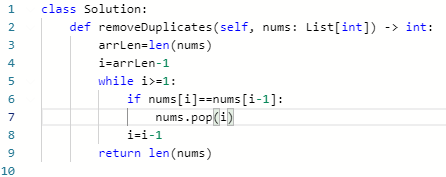


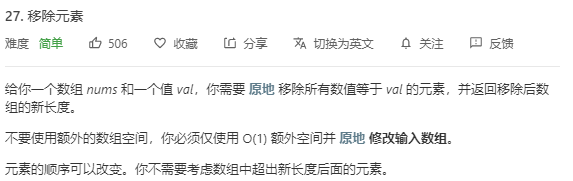


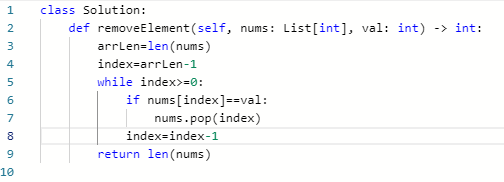




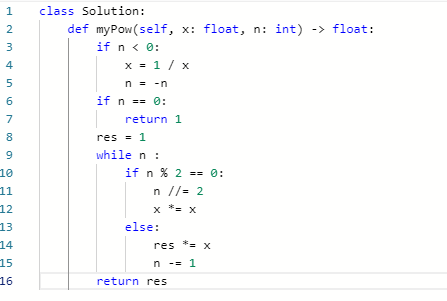


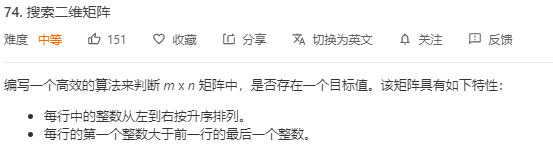


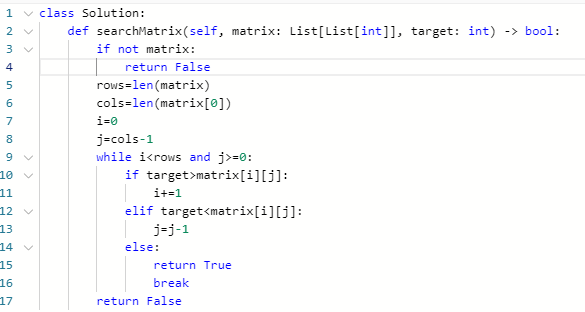


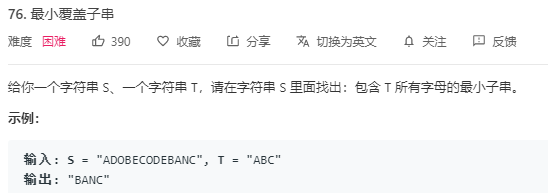


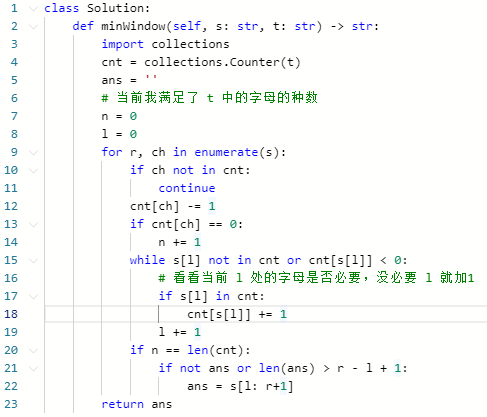


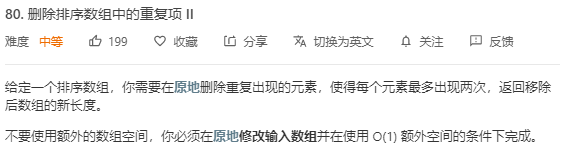


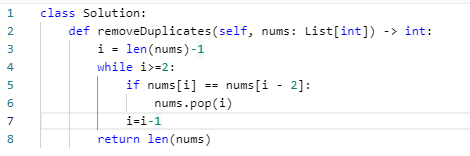




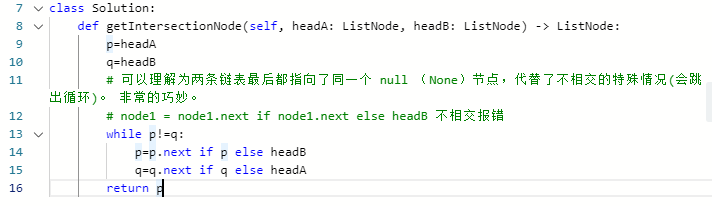


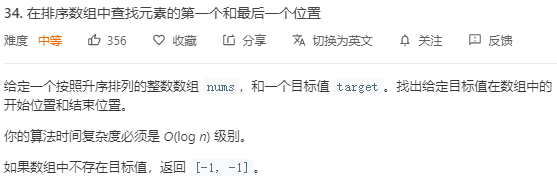


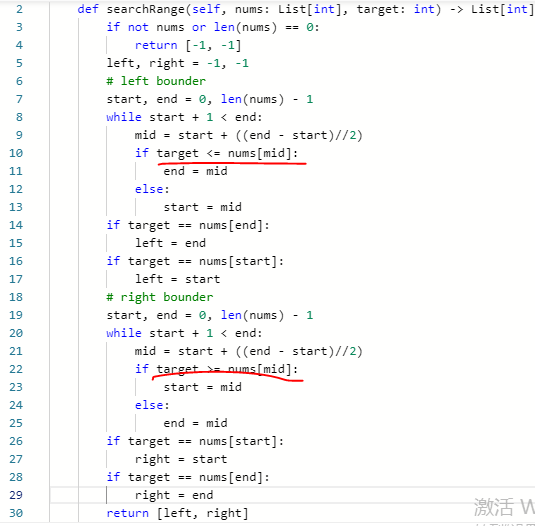


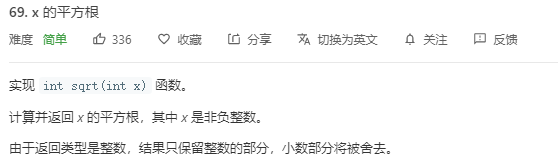


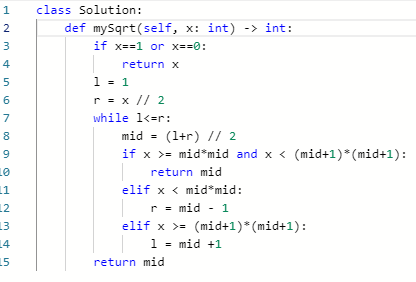


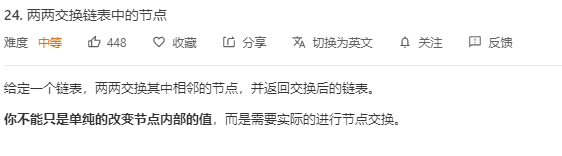


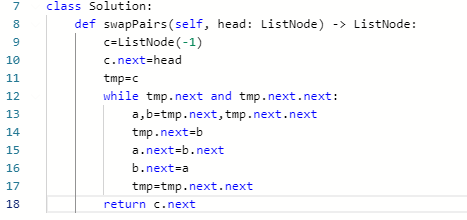


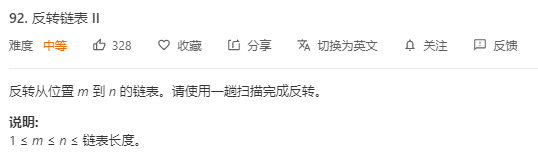


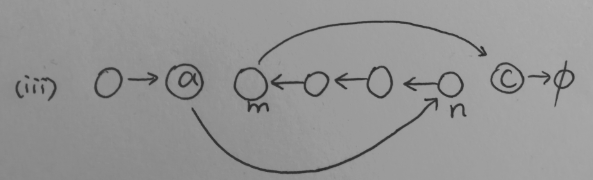


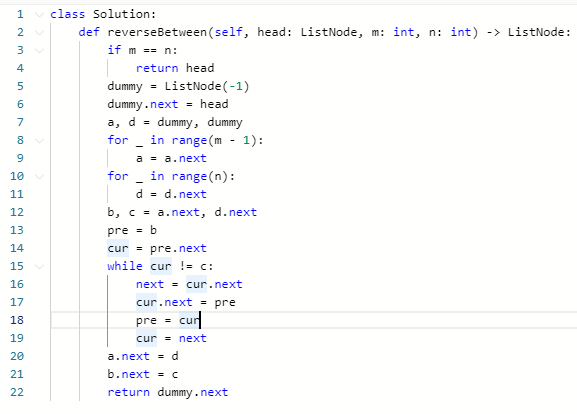


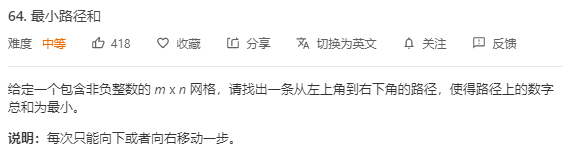


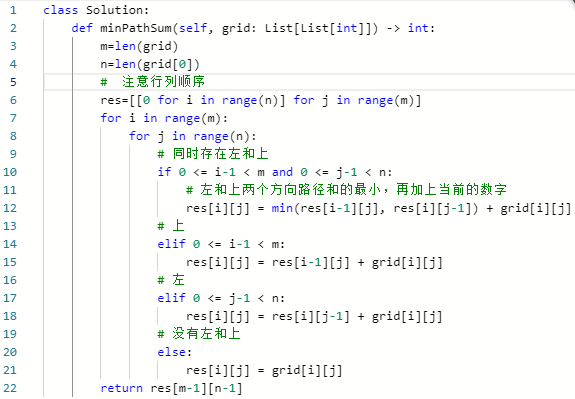


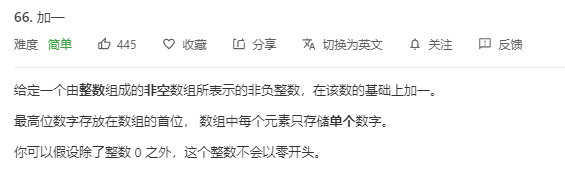


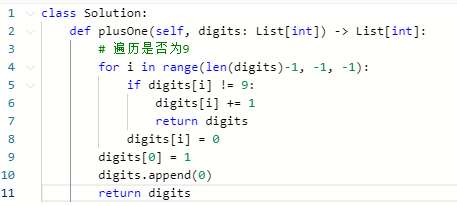


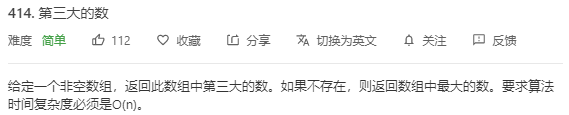


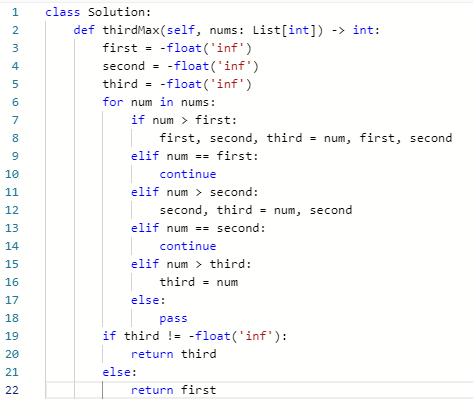


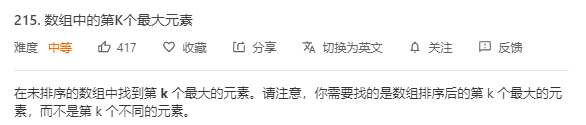


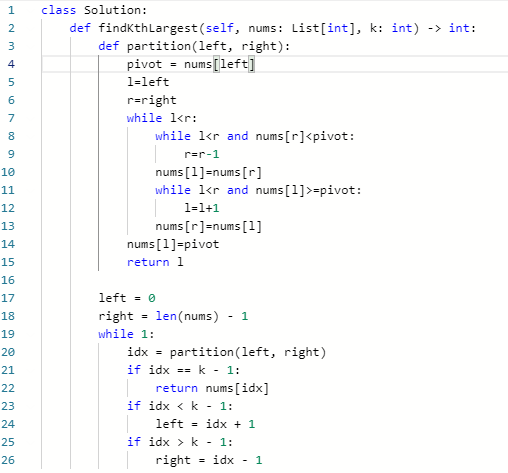




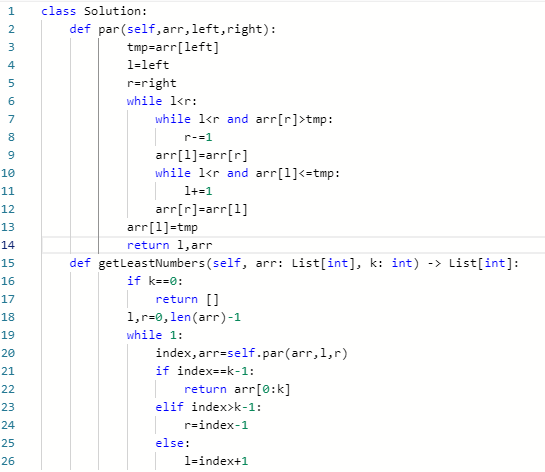


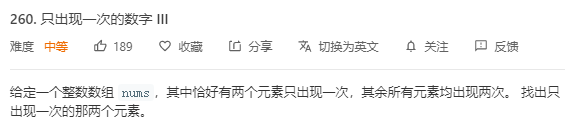


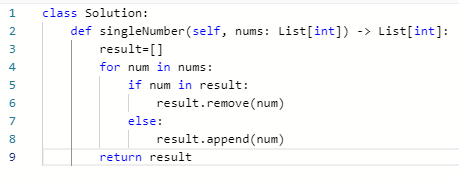


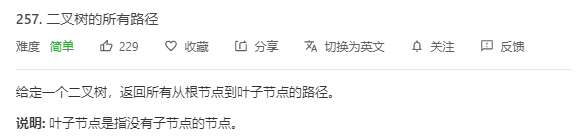


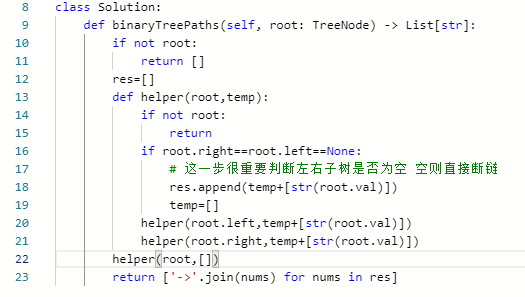


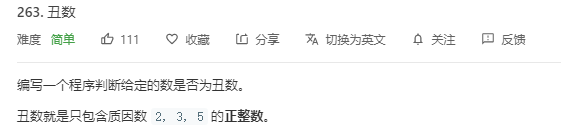


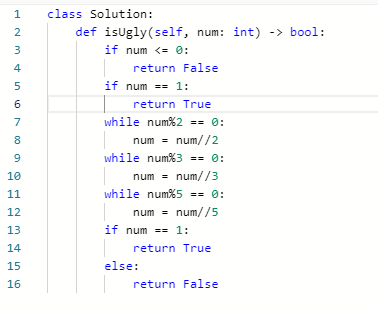


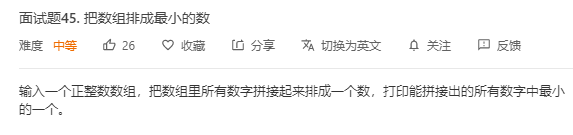


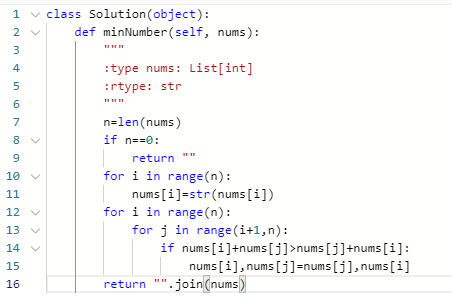




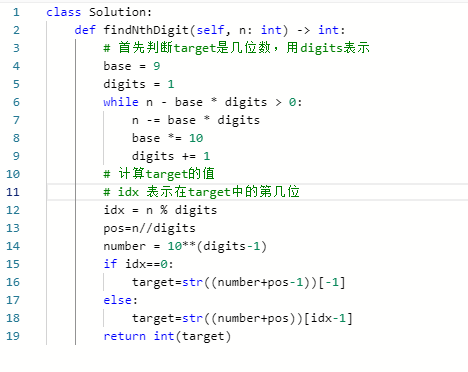




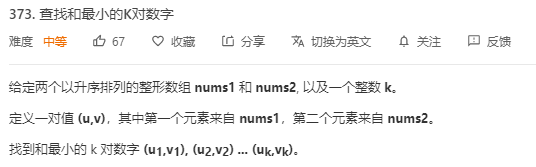


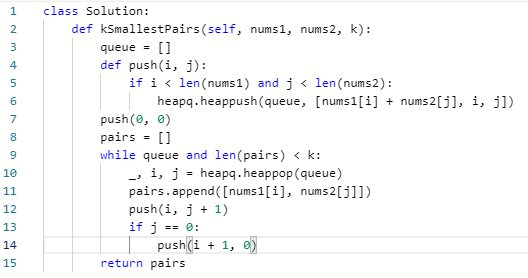


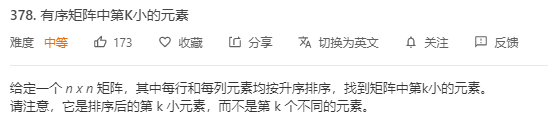




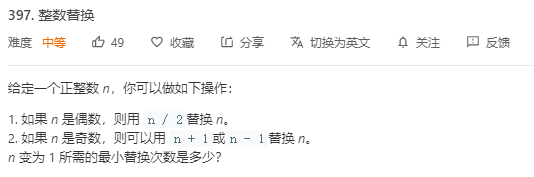
<https://blog.csdn.net/u013206202/article/details/78968438?depth_1-utm_source=distribute.pc_relevant.none-task&utm_source=distribute.pc_relevant.none-task>

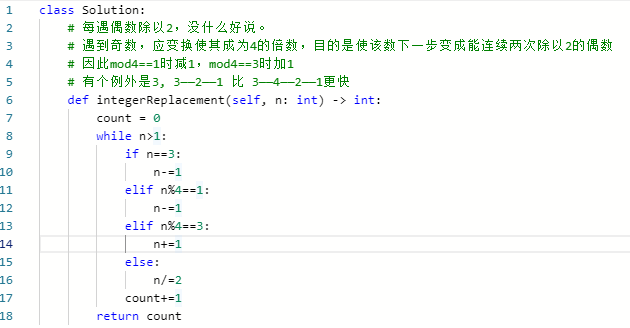




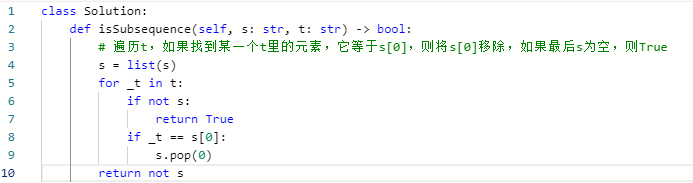


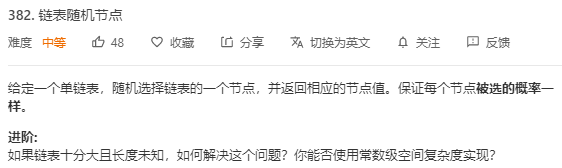


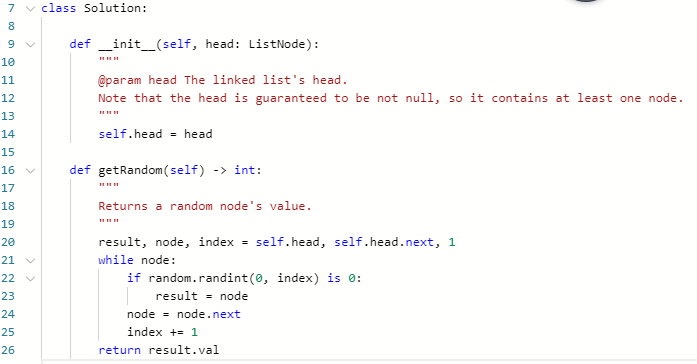


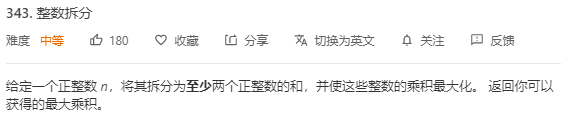


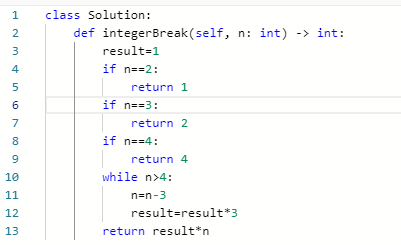


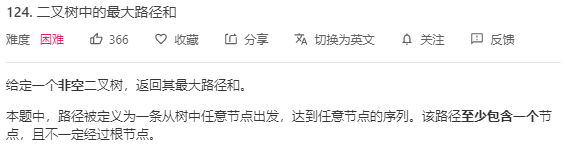


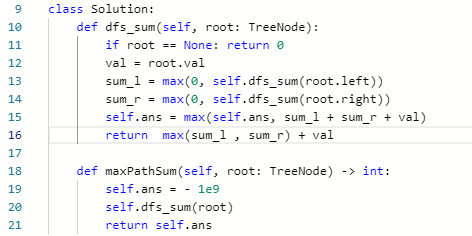




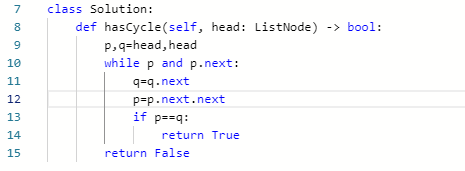






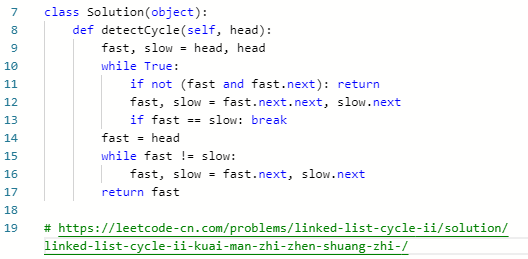


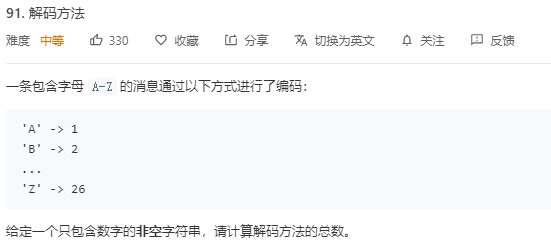


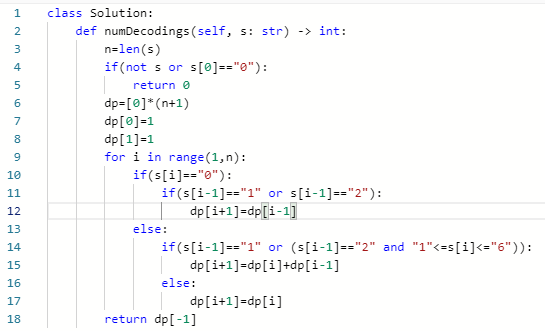


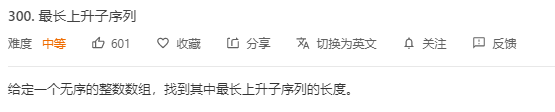
<https://leetcode-cn.com/problems/linked-list-cycle-ii/solution/linked-list-cycle-ii-kuai-man-zhi-zhen-shuang-zhi-/>

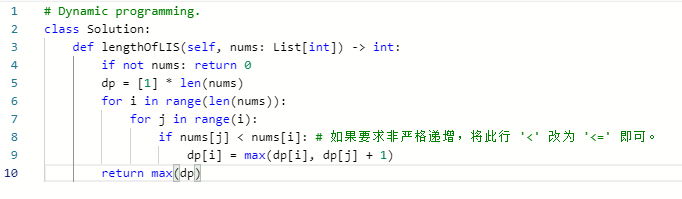


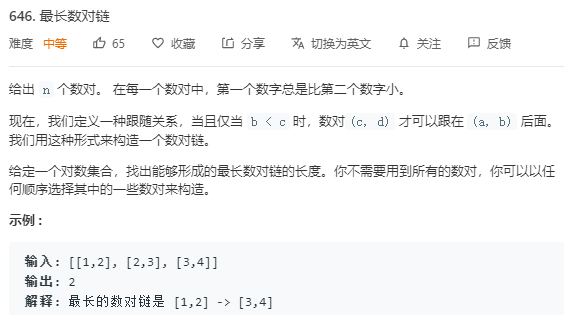


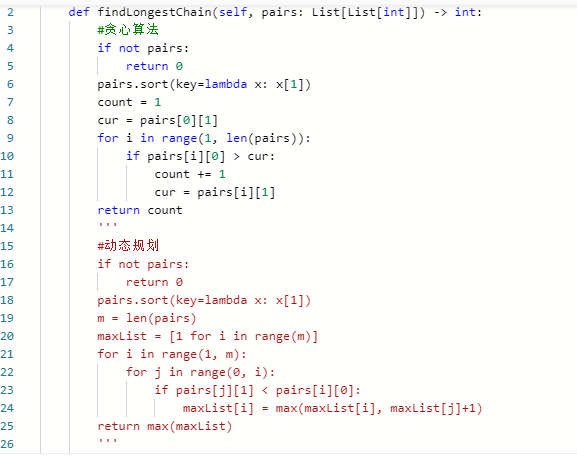


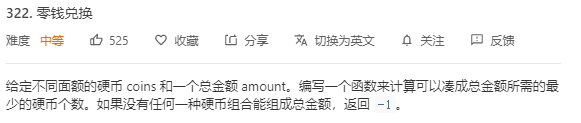


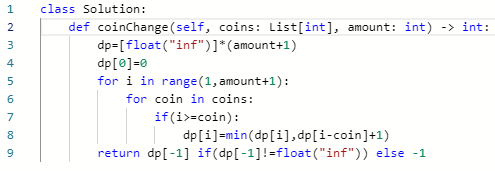


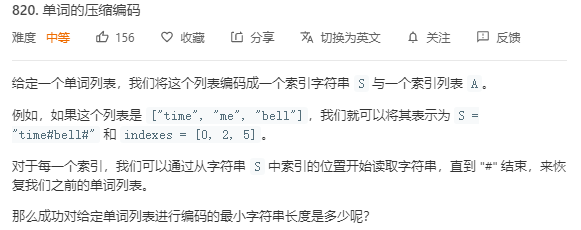


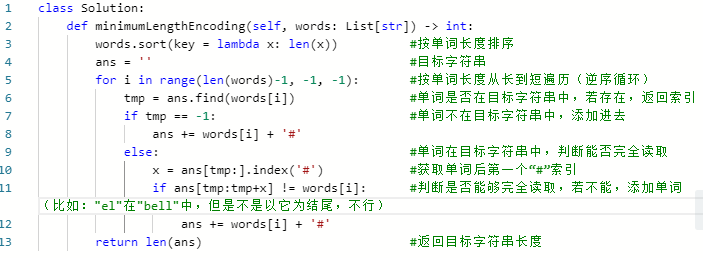


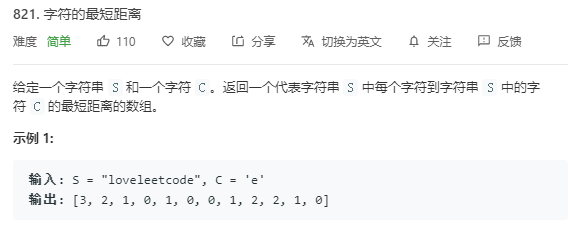


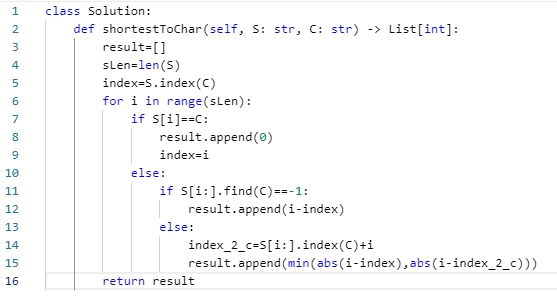


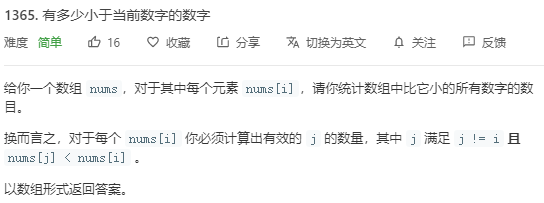


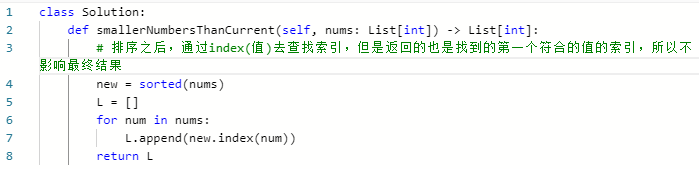


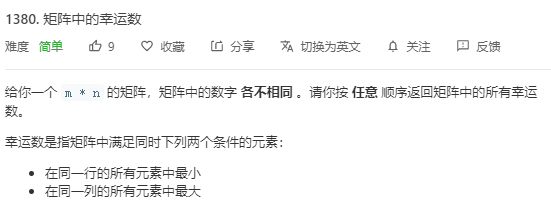


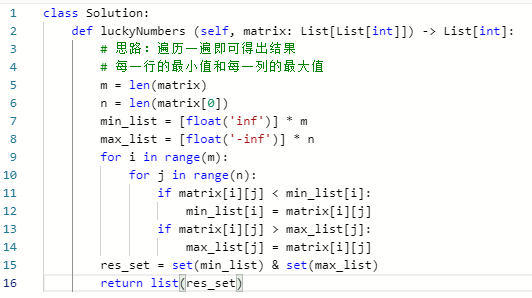


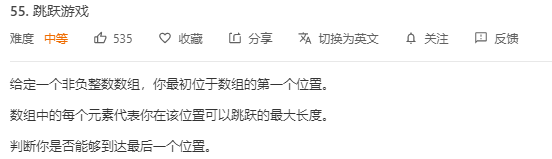


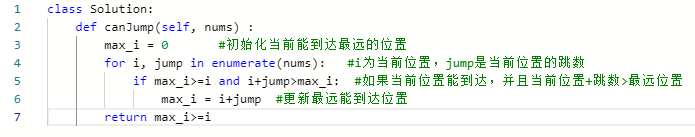


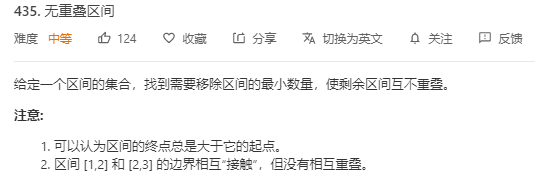


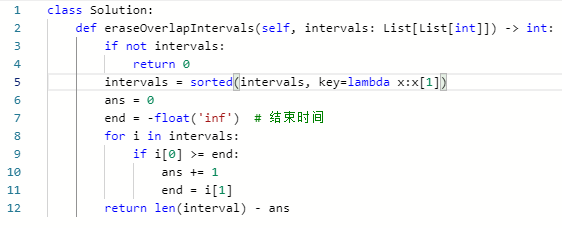


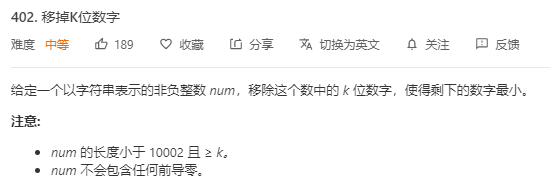




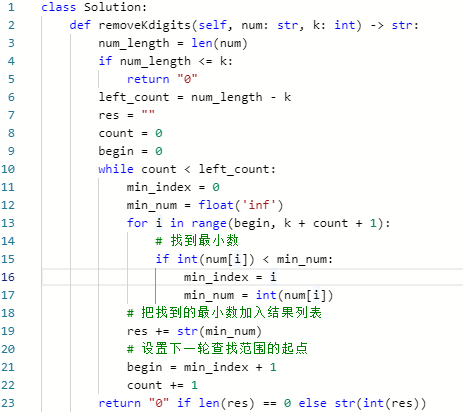


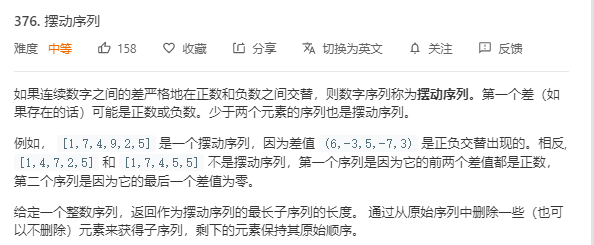


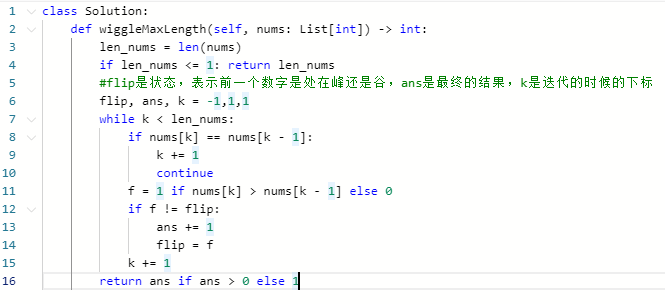




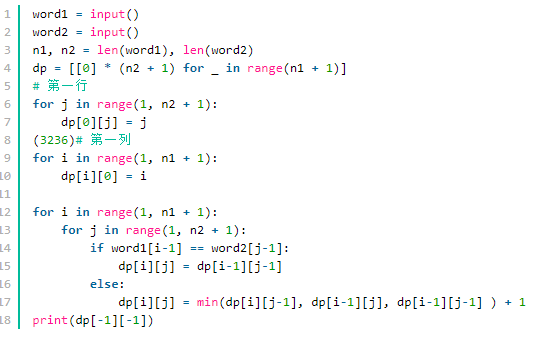
<https://leetcode-cn.com/problems/remove-k-digits/solution/cong-gao-wei-kai-shi-zai-mei-wei-shu-zi-de-you-xia/>

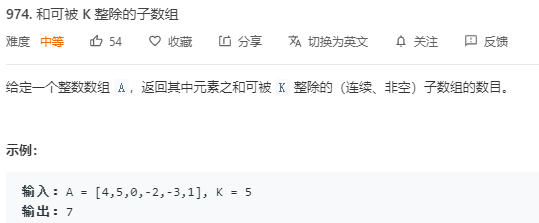


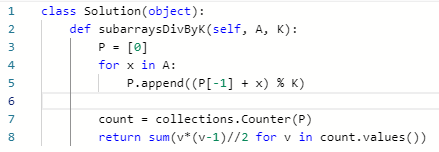


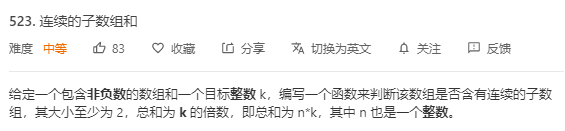


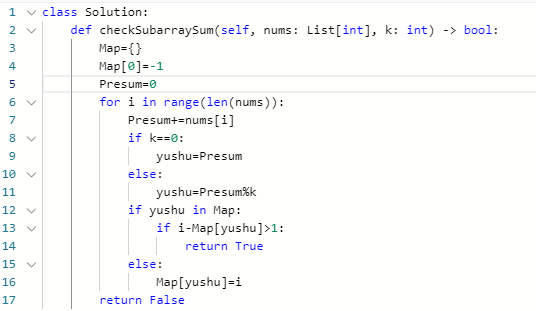
最小编辑距离：

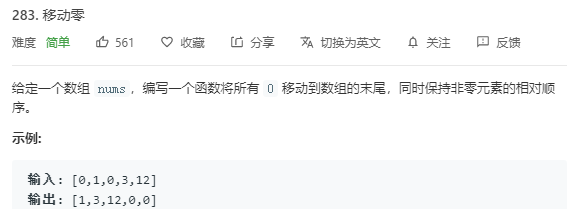


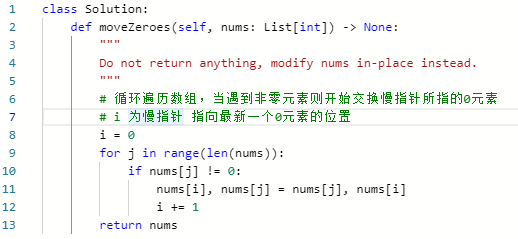


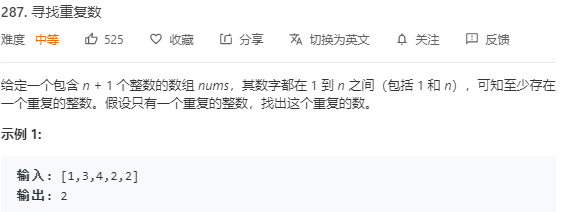


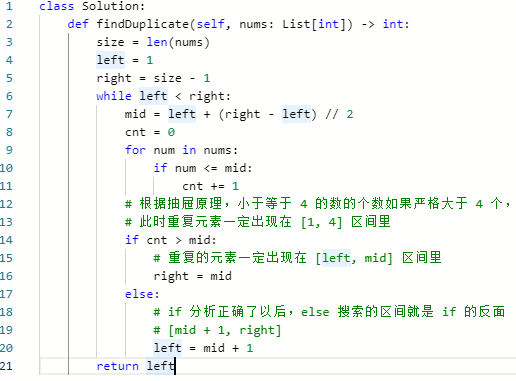


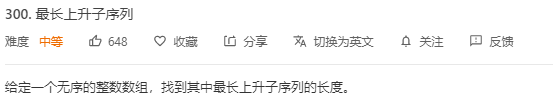


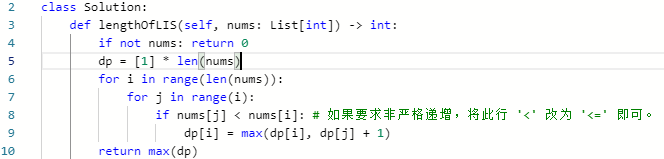




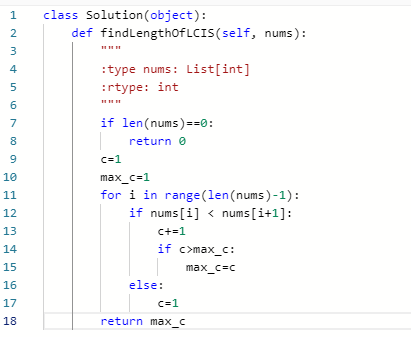


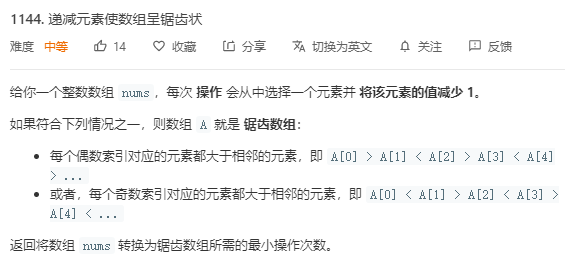


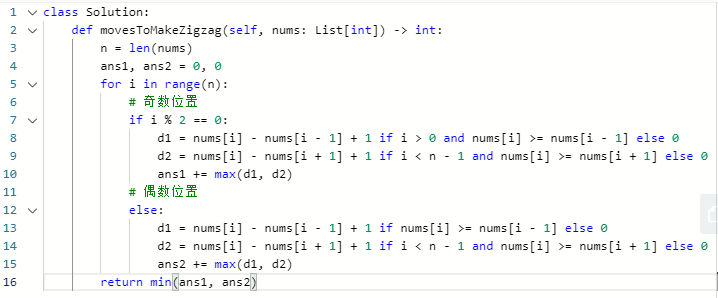


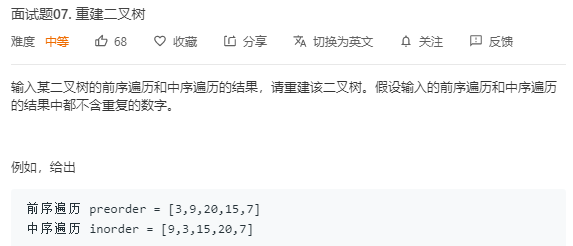


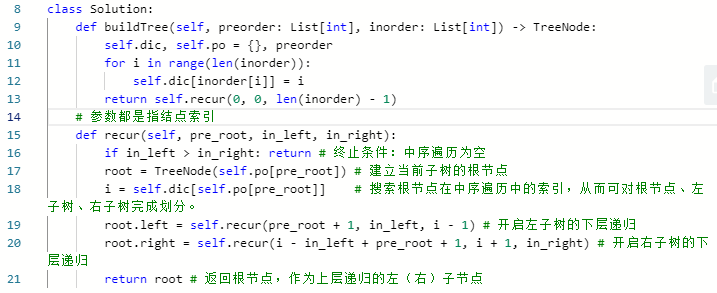


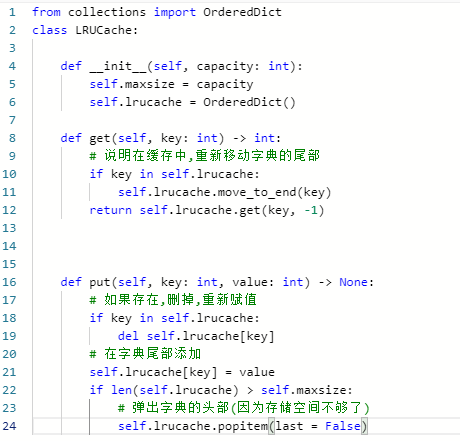




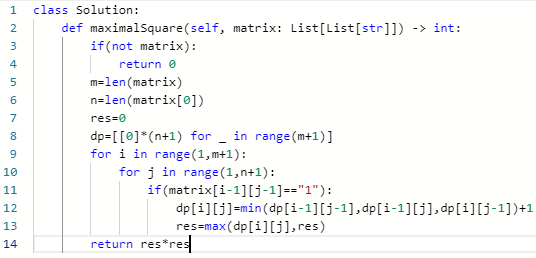


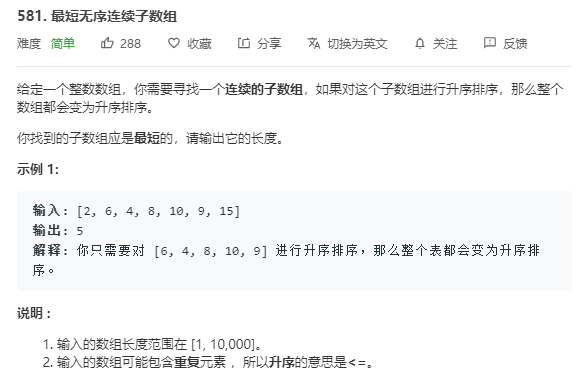


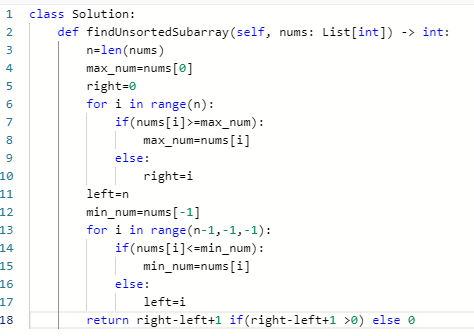




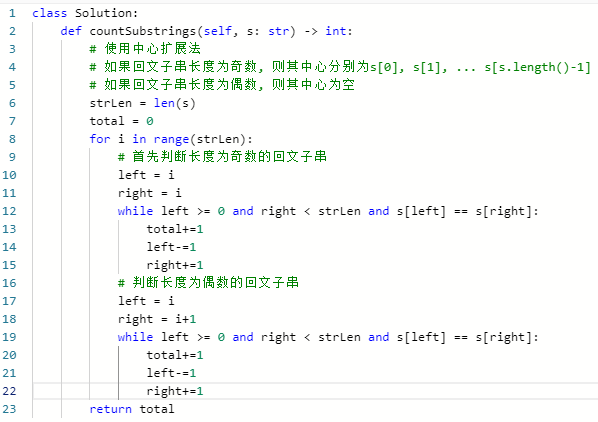


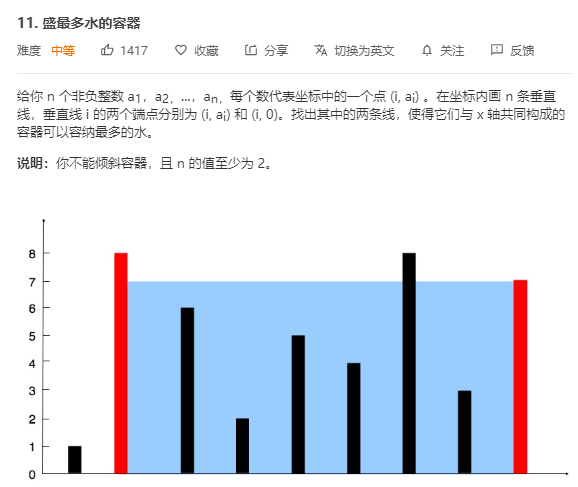


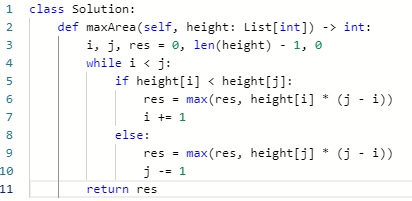




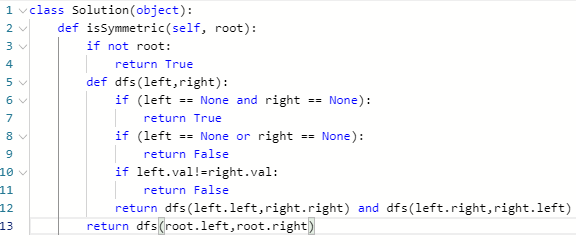






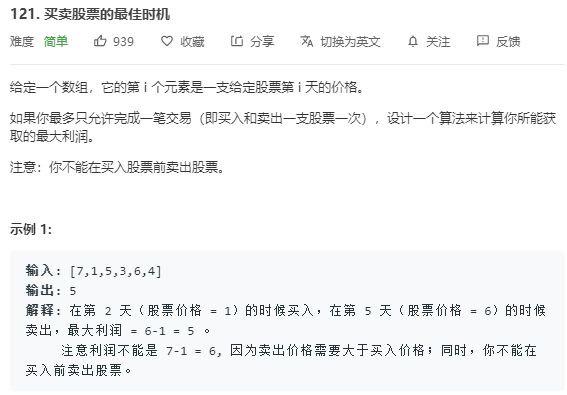


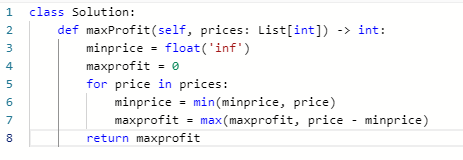












给定一个 没有重复 数字的序列，返回其所有可能的全排列。



