1. 判断一个正整数是不是回文数。

回文数的定义是，将这个数反转之后，得到的数仍然是同一个数。

 注意事项

给的数一定保证是32位正整数，但是反转之后的数就未必了。

您在真实的面试中是否遇到过这个题？

Yes

**样例**

11, 121, 1, 12321 这些是回文数。

23, 32, 1232 这些不是回文数。

思路：把数字化成字符串，然后反转一下，在比较是否相同

class Solution:

# @param {int} num a positive number

# @return {boolean} true if it's a palindrome or false

def palindromeNumber(self, num):

# Write your code here

num=str(num)

length=len(num)

if length==1:

return True

for i in range(length):

if num[i]!=num[length-i-1]:

return False

return True

2. 分割一个整数数组，使得奇数在前偶数在后。

您在真实的面试中是否遇到过这个题？

Yes

**样例**

给定 [1, 2, 3, 4]，返回 [1, 3, 2, 4]。

思路：遍历数组遇到偶数就pop下，然后append到最后，计数应该保证为数组长度

class Solution:

# @param nums: a list of integers

# @return: nothing

def partitionArray(self, nums):

# write your code here

length=len(nums)

i=j=0

while j<length:

if nums[i]&1!=0:

j+=1

i+=1

else:

t=nums[i]

del(nums[i])

nums.append(t)

j+=1

return nums

3.

给定一个单链表和数值x，划分链表使得所有小于x的节点排在大于等于x的节点之前。

你应该保留两部分内链表节点原有的相对顺序。

您在真实的面试中是否遇到过这个题？

Yes

**样例**

给定链表 **1->4->3->2->5->2->null**，并且 x=3

返回**1->2->2->4->3->5->null**

**思路：大于小于的分别存在两个列表内，然后创建列表即可。**

"""

Definition of ListNode

class ListNode(object):

def \_\_init\_\_(self, val, next=None):

self.val = val

self.next = next

"""

class Solution:

"""

@param head: The first node of linked list.

@param x: an integer

@return: a ListNode

"""

def partition(self, head, x):

# write your code here

a=[]

b=[]

temp=head

while head!=None:

if head.val>=x:

a.append(head.val)

else:

b.append(head.val)

head=head.next

head=temp

for i in range(len(b)):

head.val=b[i]

head=head.next

for j in range(len(a)):

head.val=a[j]

head=head.next

return temp

4. 给出一个不含重复数字的排列，求这些数字的所有排列按字典序排序后该排列的编号。其中，编号从1开始。

您在真实的面试中是否遇到过这个题？

Yes

**样例**

例如，排列 [1,2,4] 是第 1 个排列。

思路就是给你一个数你需要判断他是排列中的第几个。思路一：可以直接把所有情况都做成一个字典，然后索引即可。但这样复杂度有点高，而且做出全排列的化也应该是交换。加入我可以做出本身按大小的排序，然后计算他们之前的排列有多少种即可。下面就是做了。先排序做字典在把原数组按照字典找到对应序号。然后每个位置都是全排除以长度乘以当前位置。这样一直循环即可，因为最后是一要加上1

class Solution:

# @param {int[]} A an integer array

# @return {long} a long intege

def permutationIndex(self, A):

# Write your code here

return jishuan(A)+1

def jishuan(A):

length=len(A)

if length==0:

return 0

dict=getdict(A)

temp=jiecheng(length)/length\*(dict[A[0]])

sum=jishuan(A[1:])

sum+=temp

return sum

def jiecheng(num):

factorial=1

for i in range(1,num + 1):

factorial = factorial\*i

return factorial

def getdict(A):

dict={}

b=sorted(A)

for i in range(len(b)):

dict[b[i]]=i

return dict

5. 给定一个非负数，表示一个数字数组，在该数的基础上+1，返回一个新的数组。

该数字按照大小进行排列，最大的数在列表的最前面。

您在真实的面试中是否遇到过这个题？

Yes

**样例**

给定 [1,2,3] 表示 123, 返回 [1,2,4].

给定 [9,9,9] 表示 999, 返回 [1,0,0,0].

比较简单就是list化成整数，然后加一存进去，代码写的太麻烦，直接转化成字符串，字符串变成整数加一在变成字符串然后变成list

class Solution:

# @param {int[]} digits a number represented as an array of digits

# @return {int[]} the result

def plusOne(self, digits):

# Write your code here

length=len(digits)

sum=0

for i in range(length):

sum=sum+digits[i]\*pow(10,length-i-1)

sum=sum+1

a=[]

while sum>=1:

h=sum

a.append(h)

sum=sum//10

a=a[::-1]

return a

6.

给定一个整数数组A。

定义B[i] = A[0] \* ... \* A[i-1] \* A[i+1] \* ... \* A[n-1]， 计算B的时候请不要使用除法。

您在真实的面试中是否遇到过这个题？

Yes

**样例**

给出A=**[1, 2, 3]**，返回 B为**[6, 3, 2]**

**思路：把A中的元素相乘就缺少第i个元素。既然除法不能用就用乘法吧，每次都从A终取值唯独不去i==j的，这样做乘法。最后补进去。代码写的比较麻烦，其实就是做个length然后循环分内循环和外循环，当内循环的值等于外循环的值时就跳过，否则就连乘。**

class Solution:

"""

@param A: Given an integers array A

@return: An integer array B and B[i]= A[0] \* ... \* A[i-1] \* A[i+1] \* ... \* A[n-1]

"""

def productExcludeItself(self, A):

# write your code here

import copy

b=[]

if len(A)==0:

return None

for i in range(len(A)):

jishuan(A,i,b)

return b

def jishuan(A,i,b):

temp=1

if len(A)==1:

b.append(1)

return b

for j in range(len(A)):

if j==i:

temp\*=1

else:

temp\*=A[j]

return b.append(temp)

7. 给定一个**旋转**排序数组，在原地恢复其排序。

您在真实的面试中是否遇到过这个题？

Yes

**说明**

什么是旋转数组？

* 比如，原始数组为[1,2,3,4], 则其旋转数组可以是[1,2,3,4], [2,3,4,1], [3,4,1,2], [4,1,2,3]

**样例**

[4, 5, 1, 2, 3] -> [1, 2, 3, 4, 5]

出现改变的点就要切分，如果没有就一直入。就是比较大小出现拐点时切分。

class Solution:

"""

@param nums: The rotated sorted array

@return: nothing

"""

def recoverRotatedSortedArray(self, nums):

# write your code here

nums.sort()

return nums

8. 给定一个排序数组，在原数组中删除重复出现的数字，使得每个元素只出现一次，并且返回新的数组的长度。

不要使用额外的数组空间，必须在原地没有额外空间的条件下完成。

您在真实的面试中是否遇到过这个题？

Yes

**样例**

给出数组A =**[1,1,2]**，你的函数应该返回长度2，此时A=**[1,2]**。

思路直接做set然后在变成list就行。但这样需要另外开辟空间直接比较吧不相同就加一相同就删除然后什么也不做，因为len（A）减少了。

class Solution:

"""

@param A: a list of integers

@return an integer

"""

def removeDuplicates(self, A):

# write your code here

i=0

if len(A)==0:

return 0

while i<= len(A)-2:

if A[i]==A[i+1]:

del A[i+1]

else:

i=i+1

9.

跟进“删除重复数字”：

如果可以允许出现两次重复将如何处理？

思路就是允许两次重复

class Solution:

"""

@param A: a list of integers

@return an integer

"""

def removeDuplicates(self, A):

# write your code here

i=0

if len(A)==0:

return 0

while i<= len(A)-3:

if A[i]==A[i+2]:

del A[i+2]

else:

i=i+1

10. 给定一个排序链表，删除所有重复的元素每个元素只留下一个。

您在真实的面试中是否遇到过这个题？

Yes

**样例**

给出 1->1->2->null，返回 1->2->null

给出 1->1->2->3->3->null，返回 1->2->3->null

思路要是直接在上面做的话，就是找到相同的然后删除就行。

"""

Definition of ListNode

class ListNode(object):

def \_\_init\_\_(self, val, next=None):

self.val = val

self.next = next

"""

class Solution:

"""

@param head: A ListNode

@return: A ListNode

"""

def deleteDuplicates(self, head):

# write your code here

if head==None:

return None

pre=head

cur=head.next

while cur!=None :

if pre.val==cur.val:

pre.next=cur.next

cur=cur.next

else:

pre=cur

cur=cur.next

return head