

1. 题目

LC21.合并两个有序链表

linked list, <https://leetcode.cn/problems/merge-two-sorted-lists/>

思路:

代码:

```
class Solution(object):
```

```
    def mergeTwoLists(self, list1, list2):
```

```
        """
```

```
        :type list1: Optional[ListNode]
```

```
        :type list2: Optional[ListNode]
```

```
        :rtype: Optional[ListNode]
```

```
        """
```

```
        ap=list1
```

```
        bp=list2
```

```
        h=pre=ListNode(-1)
```

```
        while ap and bp:
```

```
            if ap.val<=bp.val:
```

```
                pre.next=ap
```

```
                pre=ap
```

```
                ap=ap.next
```

```
            else:
```

```
                pre.next=bp
```

```
                pre=bp
```

```
bp=bp.next
```

```
if bp:
```

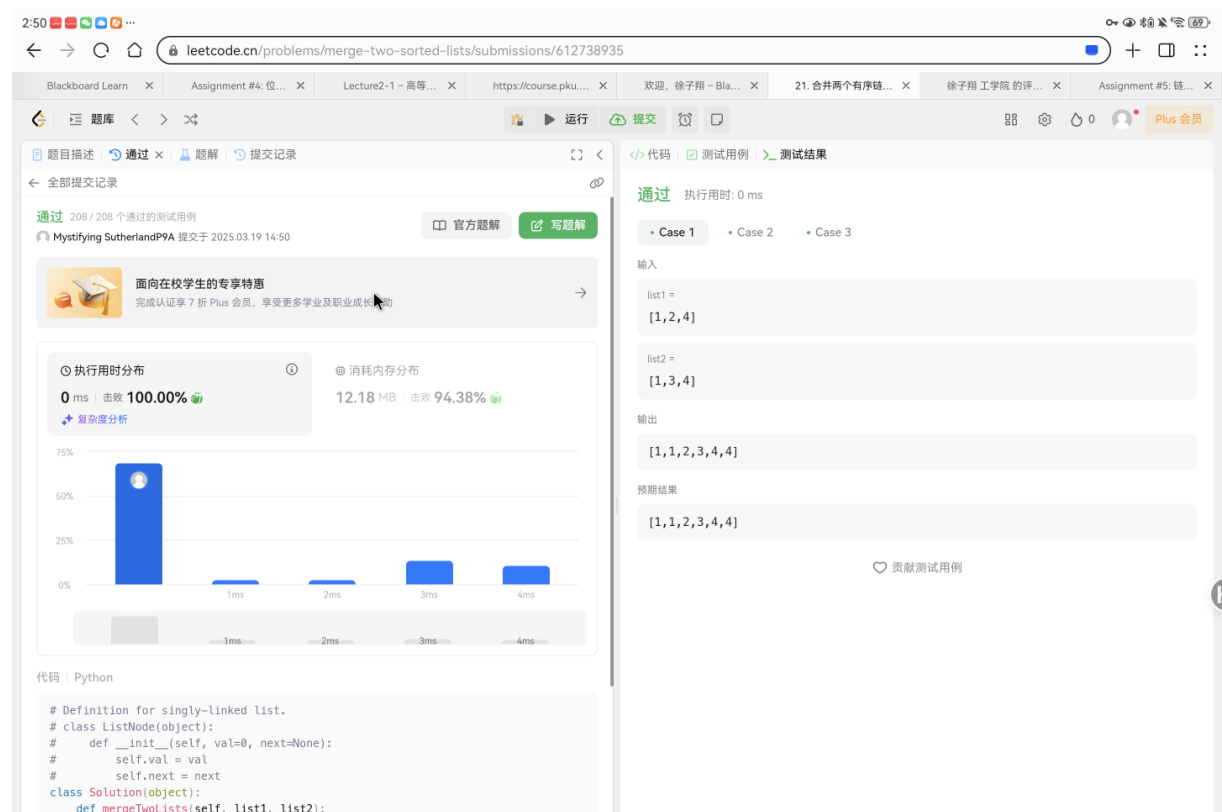
```
pre.next=bp
```

```
if ap:
```

```
pre.next=ap
```

```
return h.next
```

代码运行截图（至少包含有"Accepted"）



LC234.回文链表

linked list, <https://leetcode.cn/problems/palindrome-linked-list/>

请用快慢指针实现。

代码:

```
class Solution(object):
```

```
def isPalindrome(self, head):
```

```
    """
```

```
:type head: Optional[ListNode]
```

```
:rtype: bool
```

```
"""
```

```
fast=slow=head
```

```
while fast and fast.next:
```

```
    slow=slow.next
```

```
    fast=fast.next.next
```

```
p=slow
```

```
pre=None
```

```
while p:
```

```
    temp=p.next
```

```
    p.next=pre
```

```
    pre=p
```

```
    p=temp
```

```
p1=head
```

```
p2=pre
```

```
while p1 and p2:
```

```
    if p1.val!=p2.val:
```

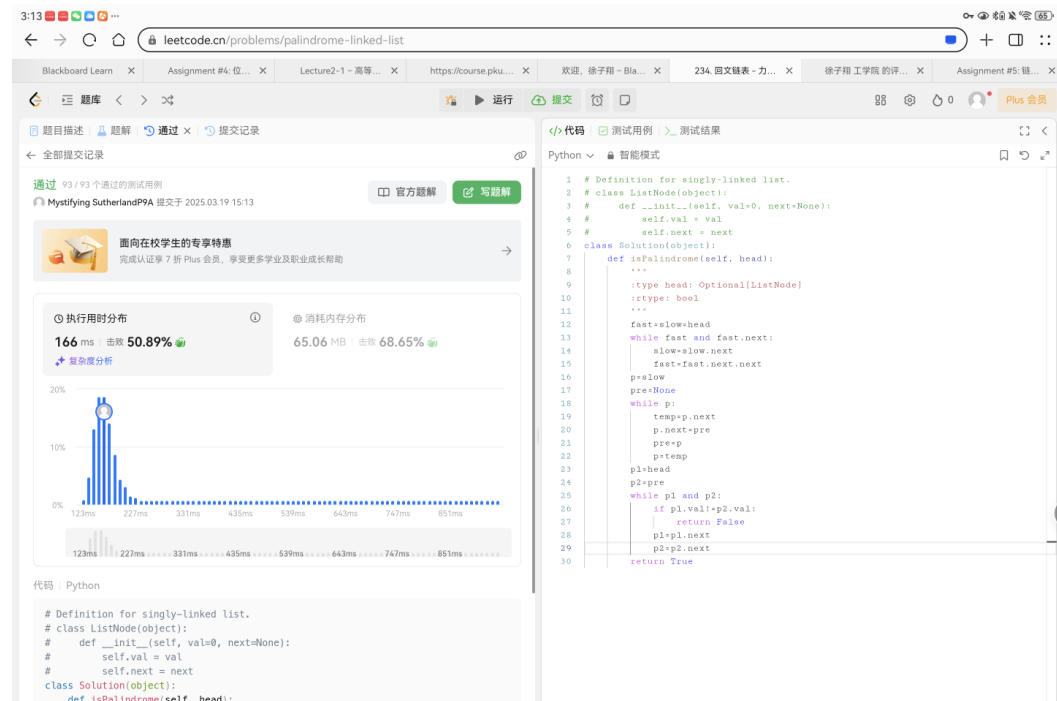
```
        return False
```

```
    p1=p1.next
```

```
    p2=p2.next
```

```
return True
```

代码运行截图 （至少包含有"Accepted"）



LC1472.设计浏览器历史记录

doubly-lined list, <https://leetcode.cn/problems/design-browser-history/>

请用双链表实现。

代码:

class link():

```
def __init__(self, url, left=None, right=None):
```

```
    self.left = left
```

```
    self.right = right
```

```
    self.url = url
```

class BrowserHistory:

```
def __init__(self, homepage):
```

```
    self.cur = link(url=homepage, right=None, left=None)
```

```
def visit(self, url):
```

```
    node = link(url)
```

```
    self.cur.right = node
```

```
    node.left = self.cur
```

```
    self.cur = node
```

```
def back(self, steps):
```

```
    cnt = 0
```

```
    while cnt < steps and self.cur.left:
```

```
        self.cur = self.cur.left
```

```
        cnt += 1
```

```
    return self.cur.url
```

```
def forward(self, steps):
```

```
    cnt = 0
```

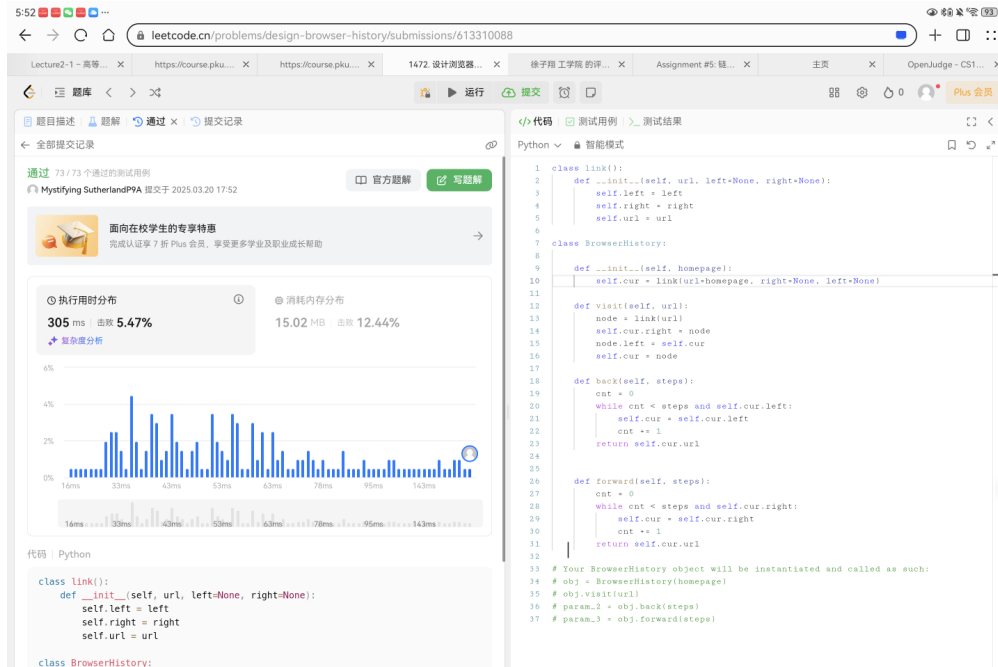
```
    while cnt < steps and self.cur.right:
```

```
        self.cur = self.cur.right
```

```
        cnt += 1
```

```
    return self.cur.url
```

代码运行截图 （至少包含有"Accepted"）



24591: 中序表达式转后序表达式
stack, <http://cs101.openjudge.cn/practice/24591/>

思路:

代码:

```
n=int(input())
def bolan(s):
    s='('+s+')'
    num=""
    numst,fhst=[],[]
    nump,fhp=-1,-1
    dic={'+':1,"-":1,"*":2,"/":2,"(":0,')':0}
    for k in s:
        numst.append("")
        fhst.append("")
        if '0'<=k<='9' or k=='.':
            num+=k
        elif num!=":
            nump+=1
            numst[nump]=num
            num=""
        if k in dic:
            if k=='(':
                fhp+=1
                fhst[fhp]=k
            elif k==')':
```

```

while nump>0 and fhp>-1 and fhst[fhp]!='(':
    numst[nump-1]+=' '+numst[nump]+' '+fhst[fhp]
    fhp-=1
    nump-=1
fhp-=1
else:
    while fhp>-1 and nump>0 and dic[fhst[fhp]]>=dic[k]:
        numst[nump-1]+=' '+numst[nump]+' '+fhst[fhp]
        fhp-=1
        nump-=1

    fhp+=1
    fhst[fhp]=k
return numst[0]

```

for i in range(n):

s=input()

print(bolan(s))

代码运行截图 （至少包含有"Accepted"）

8:25

#48647100提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

n=int(input())
def bolan(s):
    s+=' '+s+' '
    numst=''
    numst,fhst={},{}
    nump,fhp=-1,-1
    dic={'(':1,")":1,"*":2,"/":2,"(":0,")":0}
    for k in s:
        numst.append(k)
        fhst.append(k)
        if '0'<=k<='9' or k=='.':
            num+=k
        elif num!='':
            nump+=1
            numst[nump]=num
            num=''
        if k in dic:
            if k=='(':
                fhp+=1
                fhst[fhp]=k
            elif k==')':
                while nump>0 and fhp>-1 and fhst[fhp]!='(':
                    numst[nump-1]+=' '+numst[nump]+' '+fhst[fhp]
                    fhp-=1
                    nump-=1
                fhp-=1
            else:
                while fhp>-1 and nump>0 and dic[fhst[fhp]]>=dic[k]:
                    numst[nump-1]+=' '+numst[nump]+' '+fhst[fhp]
                    fhp-=1
                    nump-=1

                fhp+=1
                fhst[fhp]=k
    return numst[0]

for i in range(n):
    s=input()
    print(bolan(s))

```

基本信息

48647100
题目: 24591
提交人: 2400011041
内存: 3752kB
时间: 36ms
语言: Python3
提交时间: 2025-03-20 20:20:35

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

03253: 约瑟夫问题 No.2

queue, <http://cs101.openjudge.cn/practice/03253/>

请用队列实现。

代码:

while True:

```

n,p,m=map(int,input().split())
if n==0:
    break
op=""
l=[i for i in range(p,n+1)]+[j for j in range(1,p)]
for i in range(n):
    for j in range(m-1):
        l.append(l[0])
        del l[0]
    op+=str(l[0])+' '
    del l[0]
print(op[:-1])

```

代码运行截图 （至少包含有"Accepted"）

The screenshot shows a web browser window with the URL `cs101.openjudge.cn/practice/solution/48647332`. The page displays the submission status for problem 48647332 as "Accepted". The source code is shown in a code editor, and the submission details are listed on the right.

源代码

```

while True:
    n,p,m=map(int,input().split())
    if n==0:
        break
    op=""
    l=[i for i in range(p,n+1)]+[j for j in range(1,p)]
    for i in range(n):
        for j in range(m-1):
            l.append(l[0])
            del l[0]
        op+=str(l[0])+' '
        del l[0]
    print(op[:-1])

```

基本信息

- #: 48647332
- 题目: 03253
- 提交人: 2400011041
- 内存: 3632kB
- 时间: 29ms
- 语言: Python3
- 提交时间: 2025-03-20 20:36:29

©2002-2022 POJ 京ICP备20010980号-1 English 帮助 关于

20018: 蚂蚁王国的越野跑

merge sort, <http://cs101.openjudge.cn/practice/20018/>

思路:

代码:

```

n=int(input())
ct=0
l=[]
for i in range(n):
    l+=[int(input())]

```



```

def mergesort(l):
    if len(l)==1:
        return l
    mid=(len(l)+1)//2
    left=mergesort(l[:mid])
    right=mergesort(l[mid:])
    return merge(left,right)
def merge(left,right):
    global ct
    l=[]
    p,q=0,0
    while p<len(left) and q<len(right):
        if left[p]>=right[q]:
            l.append(left[p])
            p+=1
        else:
            ct+=len(left)-p
            l.append(right[q])
            q+=1
    l+=left[p:]+right[q:]
    return l

```

mergesort(l)

print(ct)

代码运行截图 （至少包含有"Accepted"）

The screenshot shows a web browser window with the OpenJudge website. The page title is "CS101 / 题库 (包括计概、数算题目)". The submission status for problem #48651972 is "Accepted". The source code is displayed in a text area, and the basic information on the right shows the submission was made on 2025-03-21 at 14:01:23.

状态: Accepted

源代码

```

n=int(input())
ct=0
l=[]
for i in range(n):
    l+=int(input())
def mergesort(l):
    if len(l)==1:
        return l
    mid=(len(l)+1)//2
    left=mergesort(l[:mid])
    right=mergesort(l[mid:])
    return merge(left,right)
def merge(left,right):
    global ct
    l=[]
    p,q=0,0
    while p<len(left) and q<len(right):
        if left[p]>=right[q]:
            l.append(left[p])
            p+=1
        else:
            ct+=len(left)-p
            l.append(right[q])
            q+=1
    l+=left[p:]+right[q:]
    return l
mergesort(l)
print(ct)

```

基本信息

#: 48651972
 题目: 20018
 提交人: 2400011041
 内存: 10816kB
 时间: 715ms
 语言: Python3
 提交时间: 2025-03-21 14:01:23

© 2002-2022 POJ 京ICP备20010980号-1 English 帮助 关于

2. 学习总结和收获

做 leetcode 上的题目对 class 有了更深入了解，更加熟悉了操作。

逆波兰表达式对思维要求高，需要把数字加运算符组成一个新的‘数’进行后续运算，需要抛弃固有数学运算思维。

顺带复习归并排序（注意，蚂蚁那道题如果把新元素加在列表头会超时，因为需要移动所有元素索引导致时间太多）

如果发现作业题目相对简单，有否寻找额外的练习题目，如“数算 2025spring 每日选做”、LeetCode、Codeforces、洛谷等网站上的题目。

日常跟进 OJ 每日选做