Assignment #5: 链表、栈、队列和归并排序

Updated 1348 GMT+8 Mar 17, 2025

2025 spring, Complied by 同学的姓名、院系

> **说明：**
>
> 1. **解题与记录：**
>
>   对于每一个题目，请提供其解题思路（可选），并附上使用Python或C++编写的源代码（确保已在OpenJudge， Codeforces，LeetCode等平台上获得Accepted）。请将这些信息连同显示“Accepted”的截图一起填写到下方的作业模板中。（推荐使用Typora https://typoraio.cn 进行编辑，当然你也可以选择Word。）无论题目是否已通过，请标明每个题目大致花费的时间。
>
> 2. **提交安排：**提交时，请首先上传PDF格式的文件，并将.md或.doc格式的文件作为附件上传至右侧的“作业评论”区。确保你的Canvas账户有一个清晰可见的头像，提交的文件为PDF格式，并且“作业评论”区包含上传的.md或.doc附件。
>
> 3. **延迟提交：**如果你预计无法在截止日期前提交作业，请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。
>
> 请按照上述指导认真准备和提交作业，以保证顺利完成课程要求。

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：

```
# Definition for singly-linked list.
class ListNode(object):
    def __init__(self,val=0,next=None):
        self.val=val
        self.next=None
class Solution(object):
    def mergeTwoLists(self, list1, list2):
        """
        :type list1: Optional[ListNode]
        :type list2: Optional[ListNode]
        :rtype: Optional[ListNode]
        """
        head=ListNode(-1)
        prev=head
        while list1 and list2:
            if list1.val<=list2.val:
```

```
                prev.next=list1
                list1=list1.next
            else:
                prev.next=list2
                list2=list2.next
            prev=prev.next
        prev.next=list1 if list1 is not None else list2
        return head.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：

```
# Definition for singly-linked list.
# class ListNode(object):
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution(object):
    def isPalindrome(self, head):
        """
        :type head: Optional[ListNode]
        :rtype: bool
        """
        if not head or not head.next:
            return True
        slow=fast=head
        while fast and fast.next:
            slow=slow.next
            fast=fast.next.next
        pre=None
        while slow:
            newnode=slow.next
```

```
            slow.next=pre
            pre=slow
            slow=newnode
        while pre:
            if head.val!=pre.val:
                return False
            head=head.next
            pre=pre.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 ListNode:
    def __init__(self,url):
        self.url=url
        self.next=None
        self.prev=None

class BrowserHistory(object):

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

```
        :type homepage: str
        """
        self.cur=ListNode(homepage)

    def visit(self, url):
        """
        :type url: str
        :rtype: None
        """
        newnode=ListNode(url)
        self.cur.next=newnode
        newnode.prev=self.cur
        self.cur=newnode

    def back(self, steps):
        """
        :type steps: int
        :rtype: str
        """
        while steps and self.cur.prev:
            steps-=1
            self.cur=self.cur.prev
        return self.cur.url

    def forward(self, steps):
        """
        :type steps: int
        :rtype: str
        """
        while steps and self.cur.next:
            steps-=1
            self.cur=self.cur.next
        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:

```
def trans(s):
    stack=[]
    record=[]
    num=''
    for i in s:
        if i in '0123456789' or i=='.':
            num+=i
        else:
            if num!='':
                stack.append(num)
                num=''
            if i=='(':
                record.append(i)
            elif i in '+-':
                while record!=[] and record[-1] in '/*+-':
                    stack.append(record.pop())
                record.append(i)
            elif i in '/*':
                while record!=[] and record[-1] in '/*':
                    stack.append(record.pop())
                record.append(i)
            elif i==')':
                while record!=[] and record[-1]!='(':
                    stack.append(record.pop())
                record.pop()
```

```
    if num:
        stack.append(num)
    while record:
        stack.append(record.pop())
    return stack


n=int(input())
for _ in range(n):
    s=input()
    ans=trans(s)
    print(*ans)
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
def trans(s):
    stack=[]
    record=[]
    num=''
    for i in s:
        if i in '0123456789' or i=='.':
            num+=i
        else:
            if num!='':
                stack.append(num)
                num=''
            if i=='(':
                record.append(i)
            elif i in '+-':
                while record!=[] and record[-1] in '/*+-':
                    stack.append(record.pop())
                record.append(i)
            elif i in '/*':
                while record!=[] and record[-1] in '/*':
                    stack.append(record.pop())
                record.append(i)
            elif i==')':
                while record!=[] and record[-1]!='(':
                    stack.append(record.pop())
```

03253: 约瑟夫问题No.2

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

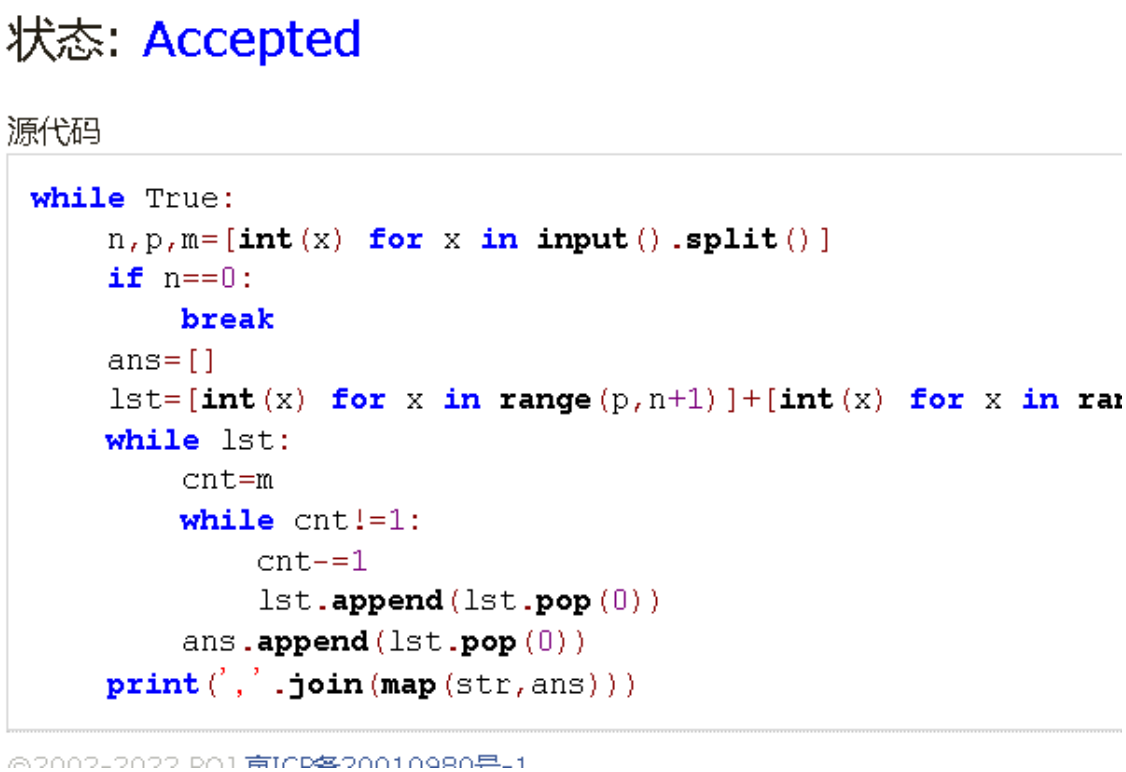
请用队列实现。

感觉这道题是作业中最简单的一题了，是因为以前写过了还有点印象么

代码：

```
while True:
    n,p,m=[int(x) for x in input().split()]
    if n==0:
        break
    ans=[]
    lst=[int(x) for x in range(p,n+1)]+[int(x) for x in range(1,p)]
    while lst:
        cnt=m
        while cnt!=1:
            cnt-=1
            lst.append(lst.pop(0))
        ans.append(lst.pop(0))
    print(','.join(map(str,ans)))
```

代码运行截图 （至少包含有"Accepted"）



20018: 蚂蚁王国的越野跑

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

思路:

不看答案自己写不出来.....

代码:

```
def merge(arr,temp,left,right):
    if left>=right:
        return 0
    mid=(left+right)//2
    cnt=merge(arr,temp,left,mid)+merge(arr,temp,mid+1,right)
```

```
    i,j,k=left,mid+1,left
    while i<=mid and j<=right:
        if arr[i]>=arr[j]:
            temp[k]=arr[i]
            i+=1
        else:
            temp[k]=arr[j]
            cnt+=(mid-i+1)
            j+=1
        k+=1
    while i<=mid:
        temp[k]=arr[i]
        i+=1
        k+=1
    while j<=right:
        temp[k]=arr[j]
        j+=1
        k+=1
    for i in range(left,right+1):
        arr[i]=temp[i]
    return cnt

n=int(input())
arr=[int(input()) for _ in range(n)]
temp=[0]*n
print(merge(arr,temp,0,n-1))
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
def merge(arr,temp,left,right):
    if left>=right:
        return 0
    mid=(left+right)//2
    cnt=merge(arr,temp,left,mid)+merge(arr,temp,mid+1,right)
    i,j,k=left,mid+1,left
    while i<=mid and j<=right:
        if arr[i]>=arr[j]:
            temp[k]=arr[i]
            i+=1
        else:
            temp[k]=arr[j]
            cnt+=(mid-i+1)
            j+=1
        k+=1
    while i<=mid:
        temp[k]=arr[i]
        i+=1
        k+=1
    while j<=right:
        temp[k]=arr[j]
        j+=1
```

2. 学习总结和收获

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。

还是感觉链表有点难懂，有的题不看答案根本做不出来……

leetcode上链表题的代码该怎么在本地运行啊……