

Assignment #5: 链表、栈、队列和归并排序

Updated 1348 GMT+8 Mar 17, 2025

2025 spring, Compiled by 郑涵予 物理学院

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:

思路和合并两个有序数组是类似的 (用时约5min)

代码:

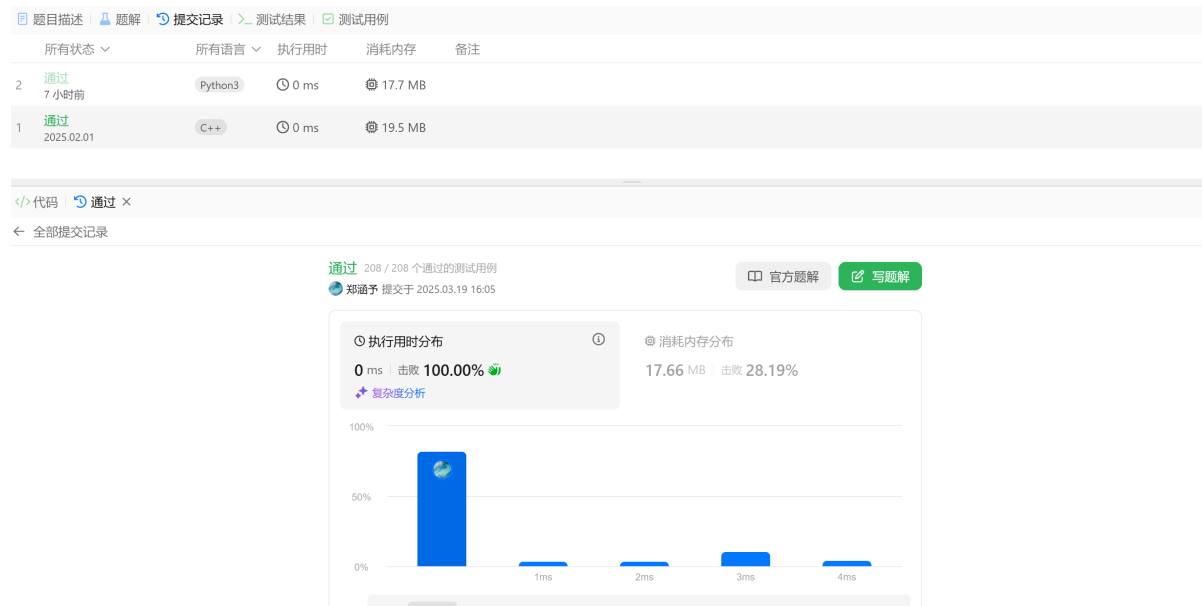
```
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution:
```

```

def mergeTwoLists(self, list1: Optional[ListNode], list2:
Optional[ListNode]) -> Optional[ListNode]:
    h=ListNode()
    cur=h
    while list1 and list2:
        if list1.val<list2.val:
            cur.next=list1
            cur=cur.next
            list1=list1.next
        else:
            cur.next=list2
            cur=cur.next
            list2=list2.next
    while list1:
        cur.next=list1
        cur=cur.next
        list1=list1.next
    while list2:
        cur.next=list2
        cur=cur.next
        list2=list2.next
    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。

如果只是简单的把题目做出来还是很容易的，直接拉到一个数组里比较就行。这里要求用快慢指针，且空间复杂度 $O(1)$ ，实际上是整合了“寻找链表中间节点”和“反转链表”两个操作，光靠自己想还是不容易想到。（用时约15min）

代码：

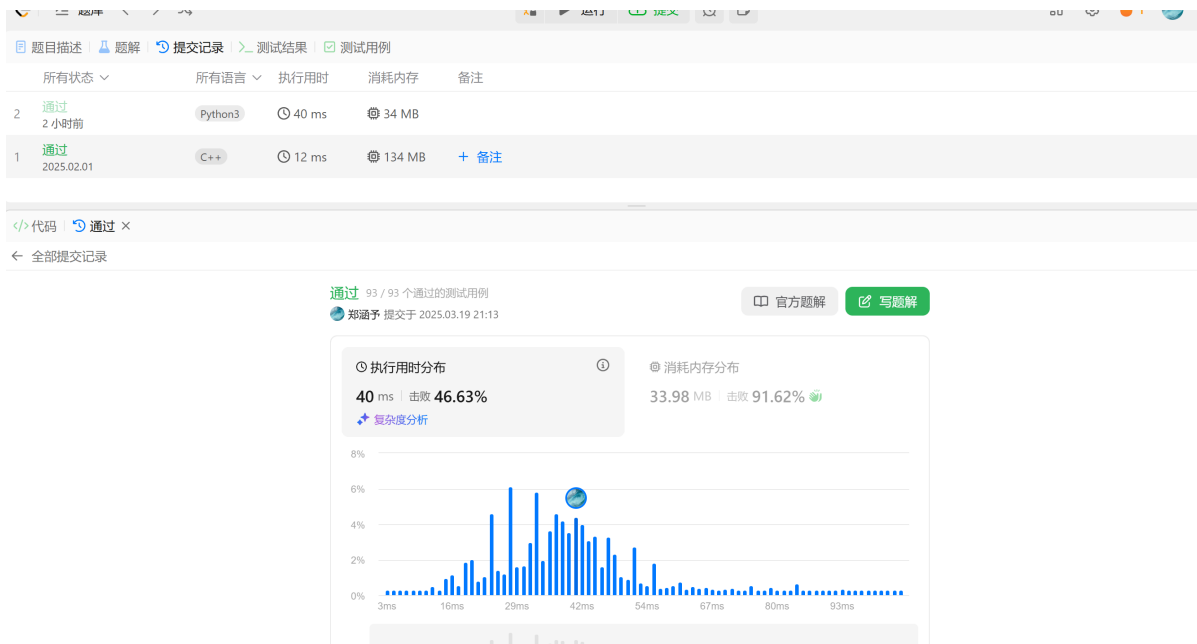
```
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution:
    def isPalindrome(self, head: Optional[ListNode]) -> bool:

        def find_mid():
            slow=fast=head
            while fast and fast.next:
                slow=slow.next
                fast=fast.next.next
            return slow

        def reverselist(head):
            pre,cur=None,head
            while cur:
                nxt=cur.next
                cur.next=pre
                pre=cur
                cur=nxt
            return pre

        mid=find_mid()
        head2=reverselist(mid)
        while head2:
            if head.val!=head2.val:
                return False
            head=head.next
            head2=head2.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。

这题在每日一题里一出现时我就做了，当时用的不是双链表而是数组，在删除前进的记录时要调用切片，会导致时间复杂度变大。而用链表就只需要更改下一个指向的节点就好了。（用时约10min）

代码：

```
class Node:
    def __init__(self, val="", next=None, pre=None):
        self.val=val
        self.next=next
        self.pre=pre
class BrowserHistory:

    def __init__(self, homepage: str):
        self.cur=Node()
        self.cur.val=homepage

    def visit(self, url: str) -> None:
        nd=Node()
        nd.val=url
        self.cur.next=nd
        nd.pre=self.cur
        self.cur=self.cur.next

    def back(self, steps: int) -> str:
```

```

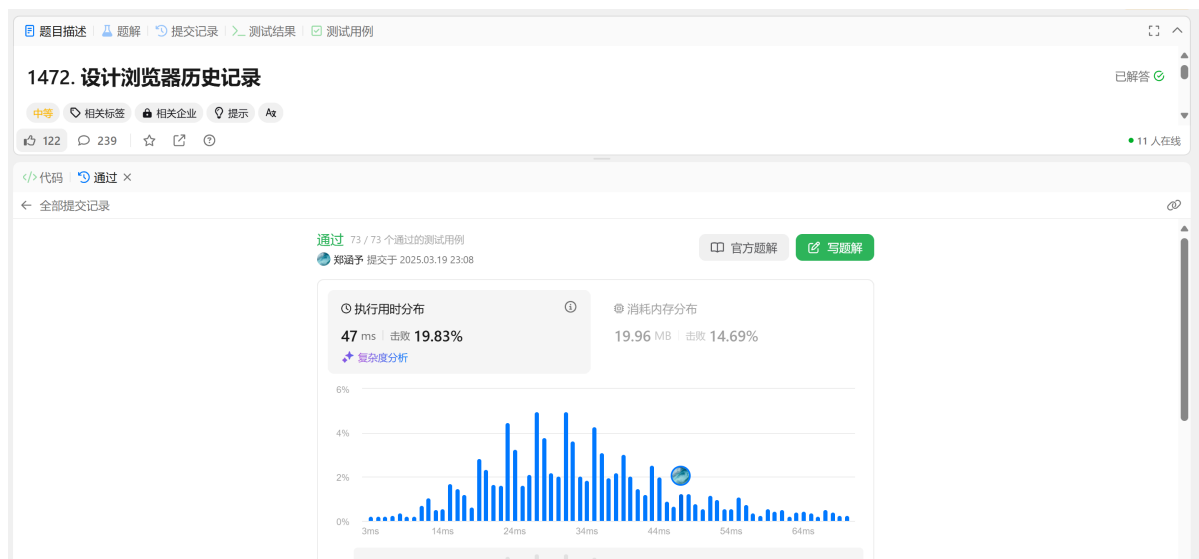
t=0
while self.cur.pre and t<steps:
    t+=1
    self.cur=self.cur.pre
return self.cur.val

def forward(self, steps: int) -> str:
    t=0
    while self.cur.next and t<steps:
        t+=1
        self.cur=self.cur.next
    return self.cur.val

# Your BrowserHistory object will be instantiated and called as such:
# obj = BrowserHistory(homepage)
# obj.visit(url)
# param_2 = obj.back(steps)
# param_3 = obj.forward(steps)

```

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:

经典题目，需要注意的是对数字的处理，因为在输入的时候并不能把一个数字一次性完整读入，所以要专门设置一个变量来存数字。另外，转化后的数字原先如果是整数是不能在后面加小数点的，所以还要额外加上一个判断。(用时约15min)

代码:

```
def change(s):
    mp={'+':1,'-':1,'*':2,'/':2}
    stack=[]
    postfix=[]
    number=""
    for x in s:
        if x.isnumeric() or x=='.':number+=x
        else:
            if number:
                postfix.append(int(number) if '.' not in number else
float(number))
                number=""
            if x in mp:
                while stack and stack[-1]in mp and mp[stack[-1]]>=mp[x]:
                    postfix.append(stack.pop())
                stack.append(x)
            elif x=='(':stack.append(x)
            elif x==')':
                while stack and stack[-1]!='(':postfix.append(stack.pop())
                stack.pop()
            if number:postfix.append(int(number) if '.' not in number else
float(number))
            while stack:
                postfix.append(stack.pop())
            return ' '.join(map(str,postfix))
n=int(input())
for _ in range(n):
    print(change(input()))
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
def change(s):
    mp={'+':1,'-':1,'*':2,'/':2}
    stack=[]
    postfix=[]
    number=""
    for x in s:
        if x.isnumeric() or x=='.':number+=x
        else:
            if number:
                postfix.append(int(number) if '.' not in number else float(number))
                number=""
            if x in mp:
                while stack and stack[-1] in mp and mp[stack[-1]]>=mp[x]:
                    postfix.append(stack.pop())
                stack.append(x)
            elif x=='(':stack.append(x)
            elif x==')':
                while stack and stack[-1]!='(':postfix.append(stack.pop())
                stack.pop()
            if number:postfix.append(int(number) if '.' not in number else float(number))
    while stack:
        postfix.append(stack.pop())
    return ' '.join(map(str,postfix))
n=int(input())
for _ in range(n):
    print(change(input()))
```

基本信息

#: 48299297
题目: 24591
提交人: 24n2400011325
内存: 3700kB
时间: 26ms
语言: Python3
提交时间: 2025-02-17 15:19:04

03253: 约瑟夫问题No.2

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

请用队列实现。

用队列直接模拟即可。（用时约5min）

代码：

```
from collections import deque
while 1:
    n,p,m=map(int,input().split())
    if not n:break
    q=deque()
    for i in range(n):
        q.append((i+p-1)%n+1)
    num=0
    res=[]
    while q:
        num+=1
        if num%m==0:
            res.append(q.popleft())
        else:
            q.append(q.popleft())
    print(' '.join(map(str,res)))
```

代码运行截图 (至少包含有"Accepted")

状态: **Accepted**

源代码

```
from collections import deque
while 1:
    n,p,m=map(int,input().split())
    if not n:break
    q=deque()
    for i in range(n):
        q.append((i+p-1)%n+1)
    num=0
    res=[]
    while q:
        num+=1
        if num%m==0:
            res.append(q.popleft())
        else:
            q.append(q.popleft())
    print(' '.join(map(str,res)))
```

基本信息

#: 48628408
题目: 03253
提交人: 24n2400011325
内存: 3612kB
时间: 30ms
语言: Python3
提交时间: 2025-03-19 16:14:28

20018: 蚂蚁王国的越野跑

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

思路:

这题是经典的利用归并排序计算逆序数，和寒假时做的Ultra-QuickSort几乎一模一样。
(用时约10min)

代码:

```
num=0
def mergesort(a):
    global num
    if len(a)>1:
        mid=len(a)//2
        L=a[:mid]
        R=a[mid:]
        mergesort(L)
        mergesort(R)
        i=j=k=0
        while i<len(L) and j<len(R):
            if L[i]>=R[j]:
                a[k]=L[i]
                i+=1
            else:
                a[k]=R[j]
                num+=len(L)-i
                j+=1
            k+=1
        while i<len(L):
            a[k]=L[i]
            i+=1
```



```

        k+=1
    while j<len(R):
        a[k]=R[j]
        j+=1
        k+=1

while 1:
    try:
        n=int(input())
        a=list(int(input()) for _ in range(n))
        num=0
        mergesort(a)
        print(num)
        print()
        input()
    except EOFError:break

```

代码运行截图 (至少包含有"Accepted")

#48606738提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

num=0
def mergesort(a):
    global num
    if len(a)>1:
        mid=len(a)//2
        L=a[:mid]
        R=a[mid:]
        mergesort(L)
        mergesort(R)
        i=j=k=0
        while i<len(L) and j<len(R):
            if L[i]>=R[j]:
                a[k]=L[i]
                i+=1
            else:
                a[k]=R[j]
                num+=len(L)-i
                j+=1
            k+=1
        while i<len(L):
            a[k]=L[i]
            i+=1
            k+=1
        while j<len(R):
            a[k]=R[j]
            j+=1
            k+=1

```

基本信息

#: 48606738
 题目: 20018
 提交人: 24n2400011325
 内存: 8964kB
 时间: 699ms
 语言: Python3
 提交时间: 2025-03-17 17:26:22

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。

这周参加了力扣的周赛，可能是因为题目比较顺手40min左右就AC了三个（然而剩下的时间对着第四题疯狂TLE）。这场周赛题目风格还是更偏向计概一点，比如第三题其实就是一个背包问题的变式，并没有涉及什么数算的知识。感觉自己做题状态还是不太稳定，状态好轻松就AC三个，状态不好可能挣扎半天才勉强AC2,希望在月考中能再多练练。

