

Assignment #5: 链表、栈、队列和归并排序

Updated 1348 GMT+8 Mar 17, 2025

2025 spring, Compiled by 胡新璞, 工学院

1. 题目

LC21.合并两个有序链表

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

思路：当两个链表都还有的时候比较大小即可，其中一个空了之后合并另一个的尾部。


代码：

```
class Solution(object):
    def mergeTwoLists(self, list1, list2):
        head = ListNode(0)
        cur = head
        while list1 and list2:
            if list1.val <= list2.val:
                cur.next = list1
                list1 = list1.next
            else:
                cur.next = list2
                list2 = list2.next
            cur = cur.next
        if list1:
            cur.next = list1
        else:
            cur.next = list2
        return head.next
```

代码运行截图（至少包含有"Accepted"）

通过 208 / 208 个通过的测试用例

 ? 。 提交于 2025.03.25 13:49

 官方题解

 写题解

🕒 执行用时分布



0 ms | 击败 100.00% 🏆

🌟 复杂度分析

LC234.回文链表

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

请用快慢指针实现。这个解法参考了题解就不复制到这里了，学了一下快慢指针。自己写的不熟练，写了个粗暴转列表判断是否回文的方法，空间复杂度较高。


代码：

```
class Solution(object):
    def isPalindrome(self, head):
        lst = []
        cur = head
        while cur:
            lst.append(cur.val)
            cur = cur.next
        return True if lst == lst[::-1] else False
```

代码运行截图 （至少包含有"Accepted"）

通过 93 / 93 个通过的测试用例

 ? 。 提交于 2025.03.25 14:17

 官方题解

 写题解



面向在校学生的专享特惠

完成认证享 7 折 Plus 会员，享受更多学业及职业成长帮助



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.pre = None
        self.next = None

class BrowserHistory:
    def __init__(self, homepage):
        self.cur = ListNode(homepage)

    def visit(self, url):
        tmp = ListNode(url)
        tmp.pre = self.cur
        self.cur.next = tmp
        self.cur = tmp


    def back(self, steps):
        while steps > 0 and self.cur.pre:
            self.cur = self.cur.pre
            steps -= 1
        return self.cur.url

    def forward(self, steps):
        while steps > 0 and self.cur.next:
            self.cur = self.cur.next
            steps -= 1
        return self.cur.url
```

代码运行截图 （至少包含有"Accepted"）

通过 73 / 73 个通过的测试用例

 ? 。 提交于 2025.03.25 21:45

 官方题解

 写题解

🕒 执行用时分布



315 ms | 击败 **5.55%**

💡 复杂度分析

💾 消耗内存分布

15.07 MB | 击败 **11.57%**

24591: 中序表达式转后序表达式

stack, <http://cs101.openjudge.cn/practice/24591/>

代码:

```
n = int(input())
for _ in range(n):
    stack1 = []
    stack2 = []
    is_num = ""
    s = list(input())
    for i in range(len(s)):
        if s[i] in "1234567890.":
            is_num += s[i]
        else:
            if is_num:
                stack1.append(is_num)
                is_num = ""
            if s[i] == "(":
                stack2.append(s[i])
            elif s[i] == ")":
                while stack2:
                    if stack2[-1] != "(":
                        stack1.append(stack2[-1])
                        stack2.pop()
                    else:
                        break
                stack2.pop()
            elif s[i] in "+-*/":
                while stack2 and stack2[-1] in "+-*/":
                    if not (stack2[-1] in "+-" and s[i] in "*/"):
                        stack1.append(stack2[-1])
                        stack2.pop()
                    else:
                        break
                stack2.append(s[i])
    if is_num:
        stack1.append(is_num)
    while stack2:
        stack1.append(stack2[-1])
        stack2.pop()
    print(" ".join(map(str, stack1)))
```

代码运行截图 （至少包含有"Accepted"）

状态: **Accepted**

源代码

```
n = int(input())
for _ in range(n):
    stack1 = []
    stack2 = []
    is_num = ""
    s = list(input())
    for i in range(len(s)):
```

基本信息

#: 48709422
题目: 24591
提交人: 2400011037
内存: 7592kB
时间: 44ms
语言: Python3
提交时间: 2025-03-25 19:19:45

03253: 约瑟夫问题 No.2

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

请用队列实现。

代码：

```
from collections import deque
while True:
    n,p,m = map(int,input().split())
    if n == 0 and p == 0 and m == 0:
        break
    else:
        ans = []
        lst = []
        for i in range(p,n+1):
            lst.append(i)
        for j in range(1,p):
            lst.append(j)
        q = deque(lst)
        while q:
            for i in range(m - 1):
                q.append(q.popleft())
            ans.append(q.popleft())
        print(*ans,sep = ",")
```

这是不用队列的做法

```
while True:
    n,p,m = map(int,input().split())
    if n == 0 and p == 0 and m == 0:
        break
    else:
        lst = []
        ans = ""
        for i in range(1,n+1):
            lst.append(i)
        index = p - 1
        while len(lst) > 1:
            index = (index + m - 1) % len(lst)
            ans = ans + str(lst[index]) + ","
            lst.pop(index)
        ans = ans + str(lst[0])
        print(ans)
```

代码运行截图 （至少包含有"Accepted"）

状态: **Accepted**

源代码

```
from collections import deque
while True:
    n,p,m = map(int,input().split())
    if n == 0 and p == 0 and m == 0:
        break
    else:
        ans = []
```

基本信息

#: 48709880
题目: 03253
提交人: 2400011037
内存: 3640kB
时间: 33ms
语言: Python3
提交时间: 2025-03-25 19:54:51

20018: 蚂蚁王国的越野跑

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

思路：merge sort 还没学太明白，对着讲义和题解看了一会，基本理解，还需要多做几个类似题，再熟悉一下模板。

代码：

代码运行截图 （至少包含有"Accepted"）

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。

对我来说还是太有挑战性了，Leetcode 写法和链表的写法都不习惯（但是题解确实很有启发性），思路基本能想到，实现还是不太容易，常需要请教 ds，作业题里要求的方法也都不是我自己写的时候用的方法（而且要求的方法感觉有些挺麻烦不太会，要参照题解，不过确实有启发性）。那个中序表达式转后序表达式的题目还是太吃操作了（因为对后序表达式比较陌生，处理起来有点困难）。要争取再多花时间巩固讲义和补每日选做。