

# Assignment #6: 链表、栈和排序

Updated 2143 GMT+8 Oct 13, 2025

2025 fall, Compiled 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本人头像，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

### E24588: 后序表达式求值

Stack, <http://cs101.openjudge.cn/practice/24588/>

思路：stack

代码：

```
n=int(input())
for _ in range(n):
    s=input().split()
    l=[]
    for x in s:
        if x[0].isnumeric():
            l.append(float(x))
        else:
            b=l.pop()
            a=l.pop()
            if x=='+' :
                c=a+b
```

```
        elif x=='-':
            c=a-b
        elif x=='*':
            c=a*b
        else:
            c=a/b
        l.append(c)
    print(f'{l[0]:.2f}')
```

代码运行截图 (至少包含有"Accepted")

#50352163提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
n=int(input())
for _ in range(n):
    s=input().split()
    l=[]
    for x in s:
        if x[0].isnumeric():
            l.append(float(x))
        else:
            b=l.pop()
            a=l.pop()
            if x=='+':
                c=a+b
            elif x=='-':
                c=a-b
            elif x=='*':
                c=a*b
            else:
                c=a/b
            l.append(c)
    print(f'{l[0]:.2f}')
```

基本信息

#: 50352163  
题目: 24588  
提交人: 24n2400011474  
内存: 3636kB  
时间: 31ms  
语言: Python3  
提交时间: 2025-10-14 12:08:47

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

## M234.回文链表

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

请用快慢指针实现  $O(1)$  空间复杂度。

思路: 翻转后一半的链表和前半一半比较

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:
        a,b=head,head
        while b and b.next:
            a=a.next
            b=b.next.next
        x=a
        y=None
        while x:
            x.next,x,y=y,x.next,x
        while y:
            if y.val!=head.val:
                return False
            y=y.next
            head=head.next
        return True

```

代码运行截图 (至少包含有"Accepted")

The screenshot shows a code execution environment with the following details:

- Submission Status:** 通过 (Passed) 93 / 93 个通过的测试用例 (93 / 93 test cases passed).
- User:** Ama2ing Tere... 提交于 2025.10.14 15:27 (Submitted on 2025.10.14 15:27).
- Execution Time:** 32 ms | 击败 71.86% (32 ms | Beat 71.86%).
- Memory Usage:** 34.25 MB | 击败 52.65% (34.25 MB | Beat 52.65%).
- Code:** The code defines a `ListNode` class and a `Solution` class with a `isPalindrome` method. It uses two pointers, `a` and `b`, to traverse the list and a third pointer `x` to reverse the first half.

## M27217: 有多少种合法的出栈顺序

<http://cs101.openjudge.cn/practice/27217/>

思路：感觉还是数学靠谱

代码：

```

from math import factorial
n=int(input())
print(factorial(2*n)//(factorial(n)**2*(n+1)))

```

代码运行截图 (至少包含有"Accepted")

状态: **Accepted**

源代码

```
from math import factorial
n=int(input())
print(factorial(2*n)//(factorial(n)**2*(n+1)))
```

基本信息

#: 50334014

题目: 27217

提交人: 24n2400011474

内存: 3580kB

时间: 19ms

语言: Python3

提交时间: 2025-10-13 09:23:28

## M24591:中序表达式转后序表达式

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

思路: ai说的, 感觉很有道理, 但最近的脑子已经思考不动了

代码

```
n=int(input())
def solve(s):
    d={'+':1, '-':1, '*':2, '/':2}
    a=[]
    b=[]
    i=0
    l=len(s)
    while i<l:
        if s[i].isdigit():
            num = ''
            while i<l and s[i].isdigit():
                num+=s[i]
                i+=1
            a.append(num)
        elif s[i] == '(':
            b.append('(')
            i+=1
        elif s[i] == ')':
            while b and b[-1] != '(':
                a.append(b.pop())
            b.pop()
            i+=1
        else:
            r=s[i]
            while (b and b[-1] != '(' and d.get(b[-1], 0)>=d.get(r, 0)):
                a.append(b.pop())
            b.append(r)
            i+=1
    while b:
        a.append(b.pop())
```

```
return ' '.join(a)
for _ in range(n):
    print(solve(input()))
```

(至少包含有"Accepted")

#50354795提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
n=int(input())
def solve(s):
    d={'+':1,'-':1,'*':2,'/':2}
    a=[]
    b=[]
    i=0
    l=len(s)
    while i<l:
        if s[i].isdigit():
            num = ''
            while i<l and s[i].isdigit():
                num+=s[i]
                i+=1
            a.append(num)
        elif s[i] == '(':
            b.append('(')
            i+=1
        elif s[i] == ')':
            while b and b[-1] != '(':
                a.append(b.pop())
            b.pop()
            i+=1
        else:
            r=s[i]
            while (b and b[-1] != '(' and d.get(b[-1], 0)>=d.get(r, 0)):
                a.append(b.pop())
            b.append(r)
            i+=1
    while b:
        a.append(b.pop())
    return ' '.join(a)
for _ in range(n):
    print(solve(input()))
```

基本信息

#: 50354795  
题目: 24591  
提交人: 24n2400011474  
内存: 3684kB  
时间: 23ms  
语言: Python3  
提交时间: 2025-10-14 14:54:32

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

## M02299:Ultra-QuickSort

merge sort, <http://cs101.openjudge.cn/practice/02299/>

思路：归并排序

代码

```
def solve(x):
    c=len(x)
    if c==1:
```

```

        return x,0
a=x[:c//2]
b=x[c//2:]
i,j=0,0
p,u=solve(a)
q,t=solve(b)
ans=[]
num=0
while i<c//2 and j<c-c//2:
    if p[i]<=q[j]:
        ans.append(p[i])
        i+=1
    else:
        ans.append(q[j])
        j+=1
        num+=c//2-i
ans+=p[i:c//2]+q[j:c-c//2]
return ans,num+u+t
while True:
    n=int(input())
    if n==0:
        break
    l=[]
    for _ in range(n):
        l.append(int(input()))
    _,w=solve(l)
    print(w)

```

(至少包含有"Accepted")

状态: Accepted

## 基本信息

#: 50354370

题目: 02299

提交人: 24n2400011474

内存: 34932kB

时间: 3908ms

语言: Python3

提交时间: 2025-10-14 14:37:18

## 源代码

```
def solve(x):
    c=len(x)
    if c==1:
        return x,0
    a=x[:c//2]
    b=x[c//2:]
    i,j=0,0
    p,u=solve(a)
    q,t=solve(b)
    ans=[]
    num=0
    while i<c//2 and j<c-c//2:
        if p[i]<=q[j]:
            ans.append(p[i])
            i+=1
        else:
            ans.append(q[j])
            j+=1
        num+=c//2-i
    ans+=p[i:c//2]+q[j:c-c//2]
    return ans,num+u+t
while True:
    n=int(input())
    if n==0:
        break
    l=[]
    for _ in range(n):
        l.append(int(input()))
    _,w=solve(l)
    print(w)
```

## M146.LRU缓存

hash table, doubly-linked list, <https://leetcode.cn/problems/lru-cache/>

思路: 双向链表

代码:

```
class ListNode:
    def __init__(self, key=None, val=None):
        self.key = key
        self.val = val
        self.prev = None
        self.next = None

class LRUCache:

    def __init__(self, capacity: int):
        self.a={}
        self.b=capacity
```

```

self.head=ListNode()
self.tail=ListNode()
self.head.next=self.tail
self.tail.prev=self.head

def move(self,k):
    k.next.prev,k.prev.next=k.prev,k.next
    k.next,k.prev=self.tail,self.tail.prev
    self.tail.prev.next,self.tail.prev=k,k

def get(self, key: int) -> int:
    k=self.a.get(key,-1)
    if key not in self.a:
        return -1
    self.move(k)
    return k.val

def put(self, key: int, value: int) -> None:
    k=self.a.get(key)
    if k:
        k.val=value
        self.move(k)
    else:
        if len(self.a)==self.b:
            k=self.head.next
            self.head.next=k.next
            k.next.prev=self.head
            del self.a[k.key]
        k=ListNode(key,value)
        k.next,k.prev=self.tail,self.tail.prev
        self.tail.prev.next,self.tail.prev=k,k
        self.a[key]=k

```

```

# Your LRUCache object will be instantiated and called as such:
# obj = LRUCache(capacity)
# param_1 = obj.get(key)
# obj.put(key,value)

```

代码运行截图 (至少包含有"Accepted")



题目描述通过 × 题解提交记录

← 全部提交记录

通过 24 / 24 个通过的测试用例  
Amazing Tere... 提交于 2025.10.14 15:26 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... →

⌚ 执行用时分布  
143 ms | 击败 54.62% 🌱  
✦ 复杂度分析

📊 消耗内存分布  
76.24 MB | 击败 69.25% 🌱

</> 代码 Python3 智能模式

```
1 class ListNode:
2     def __init__(self, key=None, val=None):
3         self.key = key
4         self.val = val
5         self.prev = None
6         self.next = None
7
8 class LRUCache:
9
10    def __init__(self, capacity: int):
11        self.a={}
12        self.b=capacity
13        self.head=ListNode()
14        self.tail=ListNode()
15        self.head.next=self.tail
16        self.tail.prev=self.head
17
18    def move(self, k):
```

已存储 行 13, 列 29

## 2. 学习总结和个人收获

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2025fall每日选做”、LeetCode、Codeforces、洛谷等网站上的题目。

最近精神状态不佳，各种作业，面试，本研，考试都压上来了，希望还能跟上每日选做