

# assignment6

## Assignment #6: 链表、栈和排序

Updated 2143 GMT+8 Oct 13, 2025

2025 fall, Complied by 杨知进 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：

栈+递归。一开始没能注意到运算优先级可能导致a c b中a、b都不是数字，都是需要递归的东西，后修正。

代码：

```
n = int(input())
def cul(s):
    if not s:
        return 0.00
    sig = s.pop()
```

```

if sig in ['+', '-', '*', '/']:
    right = cul(s)
    left = cul(s)
    if sig == '+':
        return left + right
    elif sig == '-':
        return left - right
    elif sig == '*':
        return left * right
    elif sig == '/':
        return left / right
else:
    return float(sig)
for i in range(n):
    s = input().split()
    out = cul(s)
    out = f"{out:.2f}"
    print(out)

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

n = int(input())
def cul(s):
    if not s:
        return 0.00
    sig = s.pop()
    if sig in ['+', '-', '*', '/']:
        right = cul(s)
        left = cul(s)
        if sig == '+':
            return left + right
        elif sig == '-':
            return left - right
        elif sig == '*':
            return left * right
        elif sig == '/':
            return left / right
    else:
        return float(sig)
for i in range(n):
    s = input().split()
    out = cul(s)
    out = f"{out:.2f}"
    print(out)

```

基本信息

#: 50453391  
 题目: 24588  
 提交人: 25n2400011575  
 内存: 3616kB  
 时间: 24ms  
 语言: Python3  
 提交时间: 2025-10-19 17:11:54

用时: 22min左右

## 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：

快慢指针找到中点，反转后半段。注意实际动手数数，后半段会比前半段长，while循环比较的循环条件应该用first and first.next，否则会有7个样例不通过。

代码：

```
class Solution:
    def isPalindrome(self, head: Optional[ListNode]) -> bool:
        def reverse_link(x):
            pre = None
            cur = x
            while cur:
                nex = cur.next
                cur.next = pre
                pre = cur
                cur = nex
            return pre
        fast, slow = head, head
        while fast and fast.next:
            fast=fast.next.next
            slow=slow.next
        first = head
        second = reverse_link(slow)
        out = True
        while first and first.next:
            if first.val != second.val:
                out = False
                break
            first=first.next
            second=second.next
        return out
```

代码运行截图 (至少包含有"Accepted")

Python3 智能模式

```
6 class Solution:
7     def isPalindrome(self, head: Optional[ListNode]) -> bool:
8         def reverse_link(x):
9             pre = None
10            cur = x
11            while cur:
12                nex = cur.next
13                cur.next = pre
14                pre = cur
15                cur = nex
16            return pre
17         fast,slow = head,head
18         while fast and fast.next:
19             fast=fast.next.next
20             slow=slow.next
21         first = head
22         second = reverse_link(slow)
23         out = True
24         while first and first.next:
25             if first.val != second.val:
26                 out = False
27                 break
28             first=first.next
29             second=second.next
30         return out
```

已存储

题目描述 | 通过 × | 题解 | 提交记录

← 全部提交记录

通过 93 / 93 个通过的测试用例 用时: 6 m 20 s

Strange I2itchiePAA 提交于 2025.10.20 22:25

官方题解

写题解

25fall Algo/week6/image-2.png

用时：ai辅助学习快慢指针，然后独立用6min写完

## M27217: 有多少种合法的出栈顺序

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

思路：

代码：

```
import math
n = int(input())
out = math.comb(2*n, n) // (n + 1)
print(out)
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
import math
n = int(input())
out = math.comb(2*n, n) // (n + 1)
print(out)
```

基本信息

#: 50474777  
题目: 27217  
提交人: 25n2400011575  
内存: 3580kB  
时间: 22ms  
语言: Python3  
提交时间: 2025-10-20 23:13:42

25fall Algo/week6/image-3.png

用时: 15min左右

## M24591:中序表达式转后序表达式

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

思路:

代码

(至少包含有"Accepted")

## M02299:Ultra-QuickSort

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

思路:

代码

(至少包含有"Accepted")

## M146.LRU缓存

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

思路:

代码:

```
class ListNode:
    def __init__(self, key=None, value=None):
```

```
    self.key = key
    self.value = value
    self.prev = None
    self.next = None

class LRUCache:
    def __init__(self, capacity: int):
        self.capacity = capacity
        self.hashmap = {}
        self.head = ListNode()
        self.tail = ListNode()
        self.head.next = self.tail
        self.tail.prev = self.head

    def move_to_tail(self, key):
        x = self.hashmap[key]
        x.prev.next = x.next
        x.next.prev = x.prev
        x.prev = self.tail.prev
        x.next = self.tail
        self.tail.prev.next = x
        self.tail.prev = x

    def get(self, key):
        if key in self.hashmap:
            self.move_to_tail(key)
        res = self.hashmap.get(key, -1)
        if res == -1:
            return res
        else:
            return res.value

    def put(self, key, value):
        if key in self.hashmap:
            self.hashmap[key].value = value
            self.move_to_tail(key)
        else:
            if len(self.hashmap) == self.capacity:
                self.hashmap.pop(self.head.next.key)
                self.head.next = self.head.next.next
                self.head.next.prev = self.head
            new = ListNode(key, value)
            self.hashmap[key] = new
            new.prev = self.tail.prev
            new.next = self.tail
```

```
self.tail.prev.next = new
self.tail.prev = new
```

代码运行截图 (至少包含有"Accepted")

```
26     def get(self, key):
27         if key in self.hashmap:
28             self.move_to_tail(key)
29             res = self.hashmap.get(key, -1)
30             if res == -1:
31                 return res
32             else:
33                 return res.value
34
35     def put(self, key, value):
36         if key in self.hashmap:
37             self.move_to_tail(key)
38             self.hashmap[key] = Node(key, value)
39         else:
40             if len(self.hashmap) >= self.capacity:
41                 lru = self.tail.prev
42                 self.remove(lru)
43                 del self.hashmap[lru.key]
44             self.hashmap[key] = Node(key, value)
45             self.add_to_head(key, value)
46
47     def move_to_tail(self, key):
48         node = self.hashmap[key]
49         self.remove(node)
50         self.add_to_head(key, node.value)
```

已存储

题目描述 | 题解 | 通过 X | 提交记录

全部提交记录

通过 24 / 24 个通过的测试用例

官方题解 写题解

image-4.png

用时：阅读题解+解题30min左右

## 2. 学习总结和个人收获

忽然看到群里同学说的，多看答案远比自己闷头做有用，感觉有一定道理.....或许要更加调整策略，高效利用时间，得到更多收获。