

# Assignment #4: 排序、栈、队列和树

---

Updated 0005 GMT+8 March 11, 2024

2024 spring, Compiled by 赵策 数学科学学院

## 编程环境

操作系统: Windows 11

Python编程环境: Visual Studio Code 1.86.2

## 1. 题目

---

### 05902: 双端队列

<http://cs101.openjudge.cn/practice/05902/>

思路:

代码

```
#
class Deque:
    def __init__(self):
        self.items = []
    def add(self, item):
        self.items.append(item)
    def remove_front(self):
        return self.items.pop(0)
    def remove_rear(self):
        return self.items.pop()

t=int(input())
for _ in range(t):
    n=int(input())
    deque=Deque()
    for _ in range(n):
        a,b=list(map(int, input().split()))
        if a==1:
            deque.add(b)
        else:
            if b==0:
                deque.remove_front()
            else:
                deque.remove_rear()
    res=deque.items
    if res:
        print(' '.join(map(str, res)))
```

```
else:
    print('NULL')
```

代码运行截图 (至少包含有"Accepted")

05902: 双端队列

Accepted

## 02694: 波兰表达式

<http://cs101.openjudge.cn/practice/02694/>

思路:

代码

```
#
s=input().split()[::-1]
stack=[]
for _ in s:
    if _ in '+-*/':
        a,b=stack.pop(),stack.pop()
        stack.append(str(eval(a+_+b)))
    else:
        stack.append(_)
print('%.6f'%float(stack[0]))
```

代码运行截图 (至少包含有"Accepted")

02694: 波兰表达式

Accepted

## 24591: 中序表达式转后序表达式

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

思路:

代码

```
#
def infixtopostfix(a):
    prec={'*':3,'/':3,'+':2,'-':2,'(':1}
```

```

infix=a.split()
stack=[]
postfix=[]
for i in infix:
    if i not in ['*', '/', '+', '-', '(', ')']:
        postfix.append(i)
    elif i=='(':
        stack.append(i)
    elif i==')':
        top=stack.pop()
        while top!='(':
            postfix.append(top)
            top=stack.pop()
    else:
        while stack!=[] and prec[stack[-1]]>=prec[i]:
            postfix.append(stack.pop())
        stack.append(i)
while len(stack)!=0:
    postfix.append(stack.pop())
return ' '.join(postfix)

n=int(input())
for _ in range(n):
    a=input()
    b=''
    for i in a:
        if i in ['*', '/', '+', '-', '(', ')']:
            b+=' '+i+' '
        else:
            b+=i
    print(infixtopostfix(b))

```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

24591: 中序表达式转后序表达式

Accepted

22068: 合法出栈序列

<http://cs101.openjudge.cn/practice/22068/>

思路:

代码

```

#
def is_valid_stack_sequence(x, s):
    l=len(x)
    idx={char:i+1 for i,char in enumerate(x)}

```

```

    for i in range(1-2):
        for j in range(i+1,1-1):
            for k in range(j+1,1):
                if idx[s[i]]>idx[s[k]]>idx[s[j]]:
                    return 'NO'
            return 'YES'

x=input().strip()
while True:
    try:
        s=input().strip()
        if set(s)==set(x):
            result=is_valid_stack_sequence(x,s)
            print(result)
        else:
            print('NO')
    except EOFError:
        break

```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

22068: 合法出栈序列

Accepted

## 06646: 二叉树的深度

<http://cs101.openjudge.cn/practice/06646/>

思路:

代码

```

#
n=int(input())
dp=[1]*n
for _ in range(n):
    a,b=map(int,input().split())
    for i in [a,b]:
        if i!=-1:
            dp[i-1]=dp[_]+1
print(max(dp))

```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

06646: 二叉树的深度

Accepted

## 02299: Ultra-QuickSort

<http://cs101.openjudge.cn/practice/02299/>

思路:

代码

```
#
def merge_sort(arr):
    if len(arr) <= 1:
        return arr, 0
    mid = len(arr) // 2
    left, inv_left = merge_sort(arr[:mid])
    right, inv_right = merge_sort(arr[mid:])
    merged, inv_merge = merge(left, right)
    return merged, inv_left + inv_right + inv_merge

def merge(left, right):
    merged = []
    inversions = 0
    i = j = 0
    while i < len(left) and j < len(right):
        if left[i] <= right[j]:
            merged.append(left[i])
            i += 1
        else:
            merged.append(right[j])
            inversions += len(left) - i
            j += 1
    merged.extend(left[i:])
    merged.extend(right[j:])
    return merged, inversions

def count_swaps(arr):
    _, swaps = merge_sort(arr)
    return swaps

while True:
    n = int(input())
    if n == 0:
        break
    arr = [int(input()) for _ in range(n)]
    print(count_swaps(arr))
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

02299: Ultra-QuickSort

Accepted

## 2. 学习总结和收获

---

如果作业题目简单，有否额外练习题目，比如：OJ“2024spring每日选做”、CF、LeetCode、洛谷等网站题目。

本次作业有一定难度，同时要注意题目中的细节处理，如中序转后序中在operator前后添空格；还有一些可能出现的小陷阱，如合法出栈中的待检测字符串可能含有标准字符串中没有的字符；最后一题Ultra-QuickSort本质上就是计算逆序对(Count Inversions)，其中主要用到分治算法进行合并排序，写完后又尝试用树状数组硬解减时但是空间太大内存炸了