

Assignment #4: 排序、栈、队列和树

Updated 0005 GMT+8 March 11, 2024

2024 spring, Compiled by 陈亚偲 工学院

说明:

1) The complete process to learn DSA from scratch can be broken into 4 parts:

Learn about Time complexities, learn the basics of individual Data Structures, learn the basics of Algorithms, and practice Problems.

2) 请把每个题目解题思路（可选），源码Python, 或者C++（已经在Codeforces/Openjudge上AC），截图（包含Accepted），填写到下面作业模版中（推荐使用 typora <https://typoraio.cn>，或者用 word）。AC 或者没有AC，都请标上每个题目大致花费时间。

3) 提交时候先提交pdf文件，再把md或者doc文件上传到右侧“作业评论”。Canvas需要有同学清晰头像、提交文件有pdf、“作业评论”区有上传的md或者doc附件。

4) 如果不能在截止前提交作业，请写明原因。

编程环境

==（请改为同学的操作系统、编程环境等）==

操作系统：Windows

Python编程环境：Spyder IDE 5.2.2

1. 题目

05902: 双端队列

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

思路:

直接按题意实现即可

代码

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 12 15:38:47 2024

@author: 陈亚偲2300011106
"""

from collections import deque
m=int(input())
for iiii in range(m):
    n=int(input())
```

```

ww=deque([])
for hhhh in range(n):
    p,q=map(int,input().split())
    if p==1:
        ww.append(q)
    else:
        if q:
            ww.pop()
        else:
            ww.popleft()
if len(ww)!=0:
    print(*ww)
else:
    print('NULL')

```

代码运行截图 == (至少包含有"Accepted") ==

#44182799提交状态

状态: Accepted

源代码

```

# -*- coding: utf-8 -*-
"""
Created on Tue Mar 12 15:38:47 2024

@author: 陈亚偲2300011106
"""

from collections import deque
m=int(input())
for iiii in range(m):
    n=int(input())
    ww=deque([])
    for hhhh in range(n):
        p,q=map(int,input().split())
        if p==1:
            ww.append(q)
        else:
            if q:
                ww.pop()
            else:
                ww.popleft()
    if len(ww)!=0:
        print(*ww)
    else:
        print('NULL')

```

02694: 波兰表达式

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

思路:

从后往前遍历 (相当于递归)

最后可以用format函数, 但当时不会, 用的字符串操作

代码

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 12 15:48:38 2024

@author: 陈亚偲 2300011106
"""
global boza
boza=['+', '-', '*', '/']
def dp(a):
    for i in range(len(a)-3, -1, -1):
        if a[i] in boza:
            if a[i]=='+' :
                temp=float(a[i+1])+float(a[i+2])
                del a[i]
                del a[i]
                a[i]=temp
            if a[i]=='-' :
                temp=float(a[i+1])-float(a[i+2])
                del a[i]
                del a[i]
                a[i]=temp
            if a[i]=='*' :
                temp=float(a[i+1])*float(a[i+2])
                del a[i]
                del a[i]
                a[i]=temp
            if a[i]=='/' :
                temp=float(a[i+1])/float(a[i+2])
                del a[i]
                del a[i]
                a[i]=temp
    return a[0]
a=input().split()
k=list(str(round(dp(a),6)))
w=len(k)
for i in range(w):
    if k[i]=='.' :
        for j in range(7-w+i):
            k.append('0')
print(''.join(k))
```

代码运行截图 == (至少包含有"Accepted") ==

#44182850提交状态

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 12 15:48:38 2024

@author: 陈亚恩 2300011106
"""
global boza
boza=['+', '-', '*', '/']
def dp(a):
    for i in range(len(a)-3,-1,-1):
        if a[i] in boza:
            if a[i]=='+' :
                temp=float(a[i+1])+float(a[i+2])
                del a[i]
                del a[i]
                a[i]=temp
            if a[i]=='-' :
                temp=float(a[i+1])-float(a[i+2])
                del a[i]
                del a[i]
                a[i]=temp
            if a[i]=='*' :
                temp=float(a[i+1])*float(a[i+2])
                del a[i]
                del a[i]
                a[i]=temp
            if a[i]=='/' :
                temp=float(a[i+1])/float(a[i+2])
                del a[i]
                del a[i]
                a[i]=temp
    return a[0]
a=input().split()
k=list(str(round(dp(a),6)))
w=len(k)
for i in range(w):
    if k[i]=='.' :
        for j in range(7-w+i):
            k.append('0')
print(''.join(k))
```

24591: 中序表达式转后序表达式

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

思路：

代码

```
#
```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

22068: 合法出栈序列

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

思路：

模拟出栈，依次判断（我试图找规律，找了很久还是WA，后来按gpt的招式直接ac了）

代码

```
#
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 19 00:33:34 2024

@author: 陈亚偲2300011106
"""

def c(w):
    s=[]
    for i in x:
        s.append(i)
        while s and w and s[-1] == w[0]:
            s.pop()
            w = w[1:]
    return len(s)+len(w)==0

x=input()
a=['NO','YES']
while True:
    try:
        w=input()
        print(a[c(w)])
```

```
except:
    break
```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 19 00:33:34 2024

@author: 陈亚恩2300011106
"""

def c(w):
    s=[]
    for i in x:
        s.append(i)
        while s and w and s[-1] == w[0]:
            s.pop()
            w = w[1:]
    return len(s)+len(w)==0

x=input()
a=['NO','YES']
while True:
    try:
        w=input()
        print(a[c(w)])
    except:
        break
```

©2002-2022 POJ 京ICP备20010980号-1

06646: 二叉树的深度

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

思路:

代码

```
#
```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

02299: Ultra-QuickSort

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

思路：

用bisect锁定index，其实是On2操作，但不知道为什么过了

代码

```
#
import bisect
while True:
    k=int(input())
    if k==0:
        break
    else:
        ct=0
        temp=[]
        for i in range(k):
            w=int(input())
            index=bisect.bisect_left(temp, w)
            ct+=i-index
            temp.insert(index,w)
        print(ct)
```

代码运行截图 == (AC代码截图，至少包含有"Accepted") ==

状态: Accepted

源代码

```
import bisect
while True:
    k=int(input())
    if k==0:
        break
    else:
        ct=0
        temp=[]
        for i in range(k):
            w=int(input())
            index=bisect.bisect_left(temp, w)
            ct+=i-index
            temp.insert(index,w)
        print(ct)
```

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。==

二叉树还没太懂，题还写不了，争取这周搞懂

链表解决02299应该会很快（之前的On2稀里糊涂地过了，可能是biosect极限卡了On2的常数）

链表感觉解决排序类问题非常好用，因为它的插入是O1

合法出栈序列找规律，找了很长时间，来来回回耗了两周，心态有点崩，后来gpt给出了即为简洁的模拟方法（虽然我的规律自己已经举不出反例了，但WA说明还是有问题，这个问题的规律到头来还是没能搞清楚[可能也搞不清楚了]）