Assignment #4: 排序、栈、队列和树

Updated 0005 GMT+8 March 11, 2024

2024 spring, Complied by ==同学的姓名、院系==

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：macOS Ventura 13.4.1 (c)

Python编程环境：Spyder IDE 5.2.2, PyCharm 2023.1.4 (Professional Edition)

C/C++编程环境：Mac terminal vi (version 9.0.1424), g++/gcc (Apple clang version 14.0.3, clang-1403.0.22.14.1)

1. 题目

05902: 双端队列

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

思路：

代码

```
#
m=int(input())
for i in range(m):
    n=int(input())
    x=[]
    for j in range(n):
```

```
        a,b=[int(i) for i in input().split()]
        if(a==1):
            x.append(str(b))
        else:
            if(b==0):
                x.pop(0)
            else:
                x.pop()
    if(x==[]):
        print('NULL')
    else:
        print(' '.join(x))
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
m=int(input())
for i in range(m):
    n=int(input())
    x=[]
    for j in range(n):
        a,b=[int(i) for i in input().split()]
        if(a==1):
            x.append(str(b))
        else:
            if(b==0):
                x.pop(0)
            else:
                x.pop()
    if(x==[]):
        print('NULL')
    else:
        print(' '.join(x))
```

02694: 波兰表达式

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

思路:

代码

```
 a=[str(i) for i in input().split()]
flag=0

def sr():
    s=a.pop(0)
```

```
    if(s=='+'):
        return sr()+sr()
    elif(s=='-'):
        return sr()-sr()
    elif(s=='*'):
        return sr()*sr()
    elif(s=='/'):
        return sr()/sr()
    else:
        return float(s)
print(format(sr(),'.6f'))
```

状态: Accepted

源代码

```
a=[str(i) for i in input().split()]
flag=0

def sr():
    s=a.pop(0)
    if(s=='+'):
        return sr()+sr()
    elif(s=='-'):
        return sr()-sr()
    elif(s=='*'):
        return sr()*sr()
    elif(s=='/'):
        return sr()/sr()
    else:
        return float(s)
print(format(sr(),'.6f'))
```


24591: 中序表达式转后序表达式

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

思路:

代码

```
#
def infix_to_postfix(expression):
    precedence = {'+':1, '-':1, '*':2, '/':2}
    stack = []
```

```
    postfix = []
    for char in expression:
        if char.isdigit() or char == '.':
            postfix.append(char)
        elif char in '*/+-':
            if postfix and postfix[-1] not in '*/+- ':
                postfix.append(' ')
            while stack and stack[-1] in '*/+-' and precedence[stack[-1]] >=
precedence[char]:
                postfix.append(stack.pop())
                postfix.append(' ')
            stack.append(char)
        elif char == '(':
            stack.append(char)
        elif char == ')':
            while stack and stack[-1] != '(':
                postfix.append(' ')
                postfix.append(stack.pop())
            stack.pop()
        else:
            continue
    if postfix and postfix[-1] not in '*/+- ':
        postfix.append(' ')

    while stack:
        postfix.append(' ')
        postfix.append(stack.pop())

    return ''.join(postfix).strip()
n=int(input())
expressions=[]
for i in range(n):
    expressions.append(input())
for expr in expressions:
    s=infix_to_postfix(expr)
    for i in range(len(s)):
        if s[i]==' ' and s[i-1]==' ':
            s=list(s)
            s.pop(i)
            s=''.join(s)
    print(s)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

#44292520提交状态 查看 提交

状态: Runtime Error

源代码

```
def infix_to_postfix(expression):
    precedence = {'+':1, '-':1, '*':2, '/':2}
    stack = []
    postfix = []
    for char in expression:
        if char.isdigit() or char == '.':
            postfix.append(char)
        elif char in '*/+-':
            if postfix and postfix[-1] not in '*/+- ':
                postfix.append(' ')
            while stack and stack[-1] in '*/+-' and precedence[stack[-1]
                postfix.append(stack.pop())
                postfix.append(' ')
            stack.append(char)
        elif char == '(':
            stack.append(char)
        elif char == ')':
            while stack and stack[-1] != '(':
                postfix.append(' ')
                postfix.append(stack.pop())
            stack.pop()
        else:
            continue

    # Finish the last number
    if postfix and postfix[-1] not in '*/+- ':
```

基本信息

#: 44292520
题目: 24591
提交人: 23n23000
内存: 3724kB
时间: 23ms
语言: Python3
提交时间: 2024-03-1

22068: 合法出栈序列

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

思路:

代码

```
#
def is_valid_sequence(original, sequence):
    stack = []
    j = 0
    for char in sequence:
        if char not in original:
            return "NO"
        while j < len(original) and (not stack or stack[-1] != char):
            if original[j] not in sequence:
                return "NO"
            stack.append(original[j])
            j += 1
        if not stack or stack[-1] != char:
            return "NO"
        stack.pop()
    return "YES"

x = input()
sequences=[]
try:
    while True:
```

```
        sequence = input()
        if sequence:
            sequences.append(sequence)
        else:
            break
except EOFError:
    pass
for seq in sequences:
    print(is_valid_sequence(x, seq))
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态: Accepted

源代码

```
a=[str(i) for i in input().split()]
flag=0

def sr():
    s=a.pop(0)
    if(s=='+'):
        return sr()+sr()
    elif(s=='-'):
        return sr()-sr()
    elif(s=='*'):
        return sr()*sr()
    elif(s=='/'):
        return sr()/sr()
    else:
        return float(s)
print(format(sr(),'.6f'))
```


06646: 二叉树的深度

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

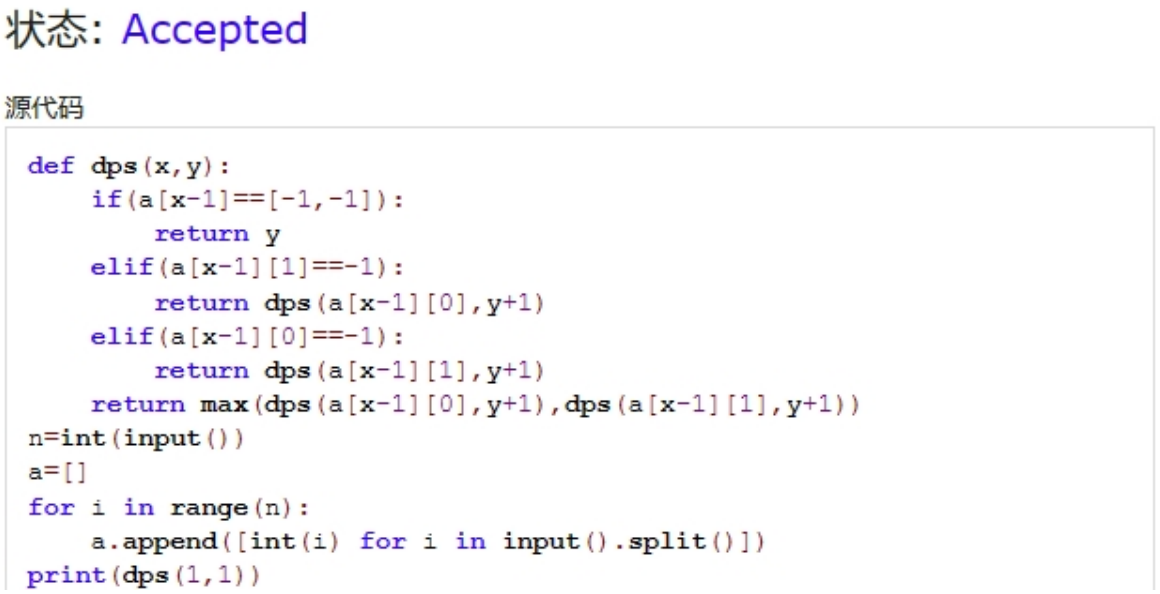
思路:

代码

```
#
def dps(x,y):
    if(a[x-1]==[-1,-1]):
        return y
    elif(a[x-1][1]==-1):
```

```
        return dps(a[x-1][0],y+1)
    elif(a[x-1][0]==-1):
        return dps(a[x-1][1],y+1)
    return max(dps(a[x-1][0],y+1),dps(a[x-1][1],y+1))
n=int(input())
a=[]
for i in range(n):
    a.append([int(i) for i in input().split()])
print(dps(1,1))
```

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:

代码

```
#
while True:
    n=int(input())
    if(n==0):
        break
    a=[]
    e=[i for i in range(n+1)]
    for i in range(n):
        a.append(int(input()))
    b=sorted(a)
    summ=0
    d={}
```

```
    for i in range(n):
                d[a[i]]=i
    for i in range(n):
        c=e[d[b[i]]]
        summ+=c-i
        if(c==i):
                continue
        elif(c>0):
            for j in range(c):
                e[j]+=1
    print(summ
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态: Runtime Error

源代码

```
while True:
    n=int(input())
    if(n==0):
        break
    a=[]
    e=[i for i in range(n+1)]
    for i in range(n):
        a.append(int(input()))
    b=sorted(a)
    summ=0
    d={}
    for i in range(n):
                d[a[i]]=i
    for i in range(n):
        c=e[d[b[i]]]
        summ+=c-i
        if(c==i):
                continue
        elif(c>0):
            for j in range(c):
                e[j]+=1
    print(summ)
```

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。==

这周有点小忙，花在数算上的时间真的不多，十分抱歉www...

下周一定把时间补回来！