

Assignment #4: 排序、栈、队列和树

Updated 0005 GMT+8 March 11, 2024

2024 spring, Compiled by 余汶青 生命科学学院

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

操作系统: 版本 Windows 11 家庭中文版

版本 22H2

安装日期 2023/7/18

操作系统版本 22621.2283

序列号 5CD323PJKL

体验 Windows Feature Experience Pack 1000.22662.1000.0

Python编程环境: * Spyder version: 5.4.3 (conda)

- Python version: 3.11.4 64-bit
- Qt version: 5.15.2
- PyQt5 version: 5.15.7
- Operating System: Windows 10

1. 题目

05902: 双端队列

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

思路:

代码

```
class two_tail:
    def __init__(self):
        self.items=[]
```

```

def push(self,n):
    self.items.append(n)
def popleft(self):
    self.items.pop(0)
def popright(self):
    self.items.pop(-1)
def qprint(self):
    if len(self.items)==0:
        print("NULL")
    else:
        print(*self.items,sep=' ')
t=int(input())
for _ in range(t):
    n=int(input())
    queue=two_tail()
    for _ in range(n):
        typ,a=map(int,input().split())
        if typ==1:
            queue.push(a)
        if typ==2 and a==0:
            queue.popleft()
        if typ==2 and a==1:
            queue.popright()
    queue.qprint()

```

代码运行截图

#44210554提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

class two_tail:
    def __init__(self):
        self.items=[]
    def push(self,n):
        self.items.append(n)
    def popleft(self):
        self.items.pop(0)
    def popright(self):

```

基本信息

#: 44210554
 题目: 05902
 提交人: 23n2300012265
 内存: 3660kB
 时间: 41ms
 语言: Python3
 提交时间: 2024-03-14 13:37:41

02694: 波兰表达式

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

思路:

代码

```

class stack:
    def __init__(self):
        self.items=[]

```

```

def push(self,n):
    self.items.append(n)
def pop(self):
    return self.items.pop(-1)

s=[i for i in input().split()]
n=len(s)
num=stack()
for i in range(n-1,-1,-1):
    if s[i]=='+':
        num.push(num.pop()+num.pop())
    elif s[i]=='-':
        num.push(num.pop()-num.pop())
    elif s[i]=='*':
        num.push(num.pop()*num.pop())
    elif s[i]=='/':
        num.push((num.pop())/num.pop())
    else:
        num.push(float(s[i]))
print("%f"%(num.pop()))

```

代码运行截图

#44210724提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

class stack:
    def __init__(self):
        self.items=[]
    def push(self,n):
        self.items.append(n)
    def pop(self):
        return self.items.pop(-1)

```

基本信息

#: 44210724
 题目: 02694
 提交人: 23n2300012265
 内存: 3536kB
 时间: 22ms
 语言: Python3
 提交时间: 2024-03-14 14:00:53

24591: 中序表达式转后序表达式

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

思路:

代码

```

class stack:
    def __init__(self):
        self.items=[]

```

```

def empty(self):
    return len(self.items)
def push(self,n):
    self.items.append(n)
def pop(self):
    return self.items.pop(-1)

n=int(input())
for _ in range(n):
    s=input()
    i=0
    ope=stack()
    num=stack()
    while i<len(s):
        if s[i]=='(':
            ope.push(s[i])
        elif s[i]=='*' or s[i]=='/':
            while ope.empty():
                a=ope.pop()
                if a=='*' or a=='/':
                    num.push(a)
                if a=='(' or a=='+' or a=='-':
                    ope.push(a)
                    break
            ope.push(s[i])
        elif s[i]=='+' or s[i]=='-':
            while ope.empty():
                a=ope.pop()
                if a=='*' or a=='/' or a=='+' or a=='-':
                    num.push(a)
                if a=='(':
                    ope.push(a)
                    break
            ope.push(s[i])
        elif s[i]==')':
            a=ope.pop()
            while a=='*' or a=='/' or a=='+' or a=='-':
                num.push(a)
                a=ope.pop()
        else:
            j=i
            while i<len(s) and s[i]!='(' and s[i]!=')' and s[i]!='*' and
s[i]!='/' and s[i]!='+' and s[i]!='-':
                i+=1
            if '.' in s[j:i]:
                num.push(float(s[j:i]))
            else:
                num.push(int(s[j:i]))
            i-=1
            i+=1
    while ope.empty():
        num.push(ope.pop())
    a=[]
    while num.empty():
        a.append(num.pop())
    for i in range(len(a)-1,-1,-1):

```

```
print(a[i],end=' ')\nprint()
```

代码运行截图

#44224803提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
class stack:\n    def __init__(self):\n        self.items=[]\n    def empty(self):\n        return len(self.items)\n    def push(self,n):\n        self.items.append(n)\n    def pop(self):
```

基本信息

#: 44224803
题目: 24591
提交人: 23n2300012265
内存: 3752kB
时间: 37ms
语言: Python3
提交时间: 2024-03-15 12:53:39

22068: 合法出栈序列

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

思路:

什么烂题，题目啥都没说清楚，害我找bug找了两小时，然后发现就因为所有字符都要输出，题目上就说明白不行吗!!!

代码

```
class stack:\n    def __init__(self):\n        self.items=[]\n    def empty(self):\n        return len(self.items)\n    def push(self,n):\n        self.items.append(n)\n    def pop(self):\n        return self.items.pop(-1)\ns=input()\nwhile True:\n    try:\n        a=input()\n        sta=stack()\n        j=0\n        v=1\n        if len(a)!=len(s):\n            print("NO")\n            v=0\n            continue\n        for i in range(len(a)):\n            index=s.find(a[i])\n            if index== -1:
```

```

        print("NO")
        v=0
        break
    if index<j:
        if sta.empty()==0 or sta.pop()!=a[i]:
            print("NO")
            v=0
            break
        else:
            for k in range(j,index):
                sta.push(s[k])
            j=index+1
    if v:
        print("YES")
except:
    break

```

代码运行截图

#44225716提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

class stack:
    def __init__(self):
        self.items=[]
    def empty(self):
        return len(self.items)
    def push(self,n):
        self.items.append(n)
    def pop(self):

```

基本信息

#: 44225716
 题目: 22068
 提交人: 23n2300012265
 内存: 3672kB
 时间: 27ms
 语言: Python3
 提交时间: 2024-03-15 14:18:07

06646: 二叉树的深度

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

思路:

代码

```

class tree:
    def __init__(self):
        self.left=None
        self.right=None

def tree_height(a):
    if a is None:
        return 0
    return max(tree_height(a.left),tree_height(a.right))+1

```

```

n=int(input())
node=[tree() for i in range(n)]
for i in range(n):
    l,r=map(int,input().split())
    if l!=-1:
        node[i].left=node[l-1]
    if r!=-1:
        node[i].right=node[r-1]
print(tree_height(node[0]))

```

代码运行截图

#44227652提交状态

状态: Accepted

源代码

```

class tree:
    def __init__(self):
        self.left=None
        self.right=None

def tree_height(a):
    if a is None:
        return 0
    return max(tree_height(a.left),tree_height(a.right))+1

```

查看	提交	账户设置	提问
基本信息		头像设置	
		联系信息	
		偏好设置	
		登出	
#: 44227652			
题目: 06646			
提交人: 23n2300012265			
内存: 3632kB			
时间: 21ms			
语言: Python3			
提交时间: 2024-03-15 15:42:14			

02299: Ultra-QuickSort

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

思路:

代码

```

ans=0
def merge_sort(alist):
    if len(alist)<=1:
        return alist
    num=len(alist)//2
    left=merge_sort(alist[:num])
    right=merge_sort(alist[num:])
    lp=0
    rp=0
    result=[]
    global ans
    while lp<len(left) and rp<len(right):
        if left[lp]<right[rp]:
            result.append(left[lp])
            lp+=1
        else:
            result.append(right[rp])
            rp+=1
    result+=left[lp:]
    result+=right[rp:]
    return result

```

```
        else:
            result.append(right[rp])
            rp+=1
            ans+=len(left)-lp
        result+=left[lp:]
        result+=right[rp:]
        return result

n=int(input())
while n!=0:
    ans=0
    s=[]
    for _ in range(n):
        s.append(int(input()))
    merge_sort(s)
    print(ans)
    n=int(input())
```

代码运行截图

#44229418提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
ans=0
def merge_sort(alist):
    if len(alist)<=1:
        return alist
    num=len(alist)//2
    left=merge_sort(alist[:num])
    right=merge_sort(alist[num:])
    lp=0
```

基本信息

#: 44229418
题目: 02299
提交人: 23n2300012265
内存: 38424kB
时间: 4261ms
语言: Python3
提交时间: 2024-03-15 17:26:35

2. 学习总结和收获

收获很大的一次作业

学习了类的用法、栈和树的写法

复习了归并排序