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==

赵云天 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

==（请改为同学的操作系统、编程环境等）==

操作系统：win10

Python编程环境：Spyder IDE 5.2.2

C/C++编程环境：

1. 题目

05902: 双端队列

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

思路：用deque即可解决

代码

```
import collections
n = int(input())
for _ in range(n):
    m = int(input())
    de = collections.deque()
    for _ in range(m):
        a,b = input().split()
```

```
        if a == '1':
            de.append(b)
        if a == '2':
            if b == '0':
                de.popleft()
            else:
                de.pop()
    if len(de) == 0:
        print('NULL')
    else:
        print(' '.join(de))
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
import collections
n = int(input())
for _ in range(n):
    m = int(input())
    de = collections.deque()
    for _ in range(m):
        a,b = input().split()
        if a == '1':
            de.append(b)
        if a == '2':
            if b == '0':
                de.popleft()
            else:
                de.pop()
    if len(de) == 0:
        print('NULL')
    else:
        print(' '.join(de))
```

基本信息

#:	44257193
题目:	05902
提交人:	23n2300012140(zyt)
内存:	3652kB
时间:	37ms
语言:	Python3
提交时间:	2024-03-16 21:42:28

02694: 波兰表达式

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

思路：不用函数也可以完成

代码

```
import collections
a = collections.deque(input().split())
b = []
if len(a) == 1:
    b.append(a[0])
else:
    c = ['+','-','*','/']
    b.append(a.popleft())
    b.append(a.popleft())
    while len(a) != 0:
        b.append(a.popleft())
```

```
        while b[-1] not in c and b[-2] not in c:
            if b[-3] =='+':
                x = float(b[-1]) + float(b[-2])
                b.pop()
                b.pop()
                b.pop()
                b.append(x)
            elif b[-3] =='-':
                x = float(b[-2]) - float(b[-1])
                b.pop()
                b.pop()
                b.pop()
                b.append(x)
            elif b[-3] =='*':
                x = float(b[-1]) * float(b[-2])
                b.pop()
                b.pop()
                b.pop()
                b.append(x)
            elif b[-3] =='/':
                x = float(b[-2]) / float(b[-1])
                b.pop()
                b.pop()
                b.pop()
                b.append(x)
            if len(b) < 3:
                break
    print("{:.6f}".format(float(b[0])))
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
import collections
a = collections.deque(input().split())
b = []
if len(a) == 1:
    b.append(a[0])
else:
    c = ['+','-','*','/']
    b.append(a.popleft())
    b.append(a.popleft())
    while len(a) != 0:
        b.append(a.popleft())
        while b[-1] not in c and b[-2] not in c:
            if b[-3] =='+':
                x = float(b[-1]) + float(b[-2])
                b.pop()
                b.pop()
                b.pop()
                b.append(x)
            elif b[-3] =='-':
```

基本信息

#:	44264386
题目:	02694
提交人:	23n2300012140(zyt)
内存:	3620kB
时间:	23ms
语言:	Python3
提交时间:	2024-03-17 13:03:59

24591: 中序表达式转后序表达式

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

思路：看了题解后模仿着写了一遍

代码

```
def zuan(a):
    b = {'+':1,'-':1,'*':2,'/':2}
    re = []
    me = []
    nu = ''
    for i in a:
        if i.isnumeric() or i == '.':
          nu += i
        else:
            if nu:
                nu = float(nu)
                re.append(int(nu) if nu.is_integer() else nu)
                nu = ''
            if i in '+-*/':
                while me and me[-1] in '+-*/' and b[me[-1]] >= b[i]:
                    re.append(me.pop())
                me.append(i)
            elif i == '(':
                me.append(i)
            elif i == ')':
                while me and me[-1] != '(':
                    re.append(me.pop())
                me.pop()
    if nu:
         nu = float(nu)
         re.append(int(nu) if nu.is_integer() else nu)
    while me:
        re.append(me.pop())
    return ' '.join(str(j) for j in re)

n = int(input())
for _ in range(n):
    a = input()
    print(zuan(a))
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态：Accepted

源代码

```
def zuan(a):
    b = {'+':1,'-':1,'*':2,'/':2}
    re = []
    me = []
    nu = ''
    for i in a:
        if i.isnumeric() or i == '.':
          nu += i
        else:
            if nu:
                nu = float(nu)
                re.append(int(nu) if nu.is_integer() else nu)
                nu = ''
            if i in '+-*/':
                while me and me[-1] in '+-*/' and b[me[-1]] >= b[i]:
                    re.append(me.pop())
                me.append(i)
            elif i == '(':
                me.append(i)
            elif i == ')':
                while me and me[-1] != '(':
                    re.append(me.pop())
                me.pop()
    if nu:
         nu = float(nu)
```

基本信息

#:	44285306
题目:	24591
提交人:	23n2300012140(zyt)
内存:	3760kB
时间:	28ms
语言:	Python3
提交时间:	2024-03-18 15:15:44

22068: 合法出栈序列

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

思路：理解题目的含义花了一些时间

代码

```
import collections
def guo(b):
    if len(a) != len(b):
        return False
    else:
        duan = []
        aa = collections.deque(a)
        for i in b:
            while (duan == [] or duan[-1] != i) and aa:
                duan.append(aa.popleft())
            if duan == [] or duan[-1] != i:
                return False
            duan.pop()
        return True


a = list(input())
while True:
    try:
        b = list(input())
        if guo(b):
            print('YES')
        else:
```

```
            print('NO')
    except EOFError:
        break
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态: Accepted

源代码

```
import collections
def guo(b):
    if len(a) != len(b):
        return False
    else:
        duan = []
        aa = collections.deque(a)
        for i in b:
            while (duan == [] or duan[-1] != i) and aa:
                duan.append(aa.popleft())
            if duan == [] or duan[-1] != i:
                return False
            duan.pop()
        return True


a = list(input())
while True:
    try:
        b = list(input())
        if guo(b):
            print('YES')
        else:
```

基本信息

#:	44285690
题目:	22068
提交人:	23n2300012140(zyt)
内存:	3648kB
时间:	25ms
语言:	Python3
提交时间:	2024-03-18 15:47:20

06646: 二叉树的深度

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

思路：对类的写法还不太熟悉

代码

```
n = int(input())
dt = {i:[] for i in range(1,n+1)}
for j in range(1,n+1):
    dt[j] = list(map(int,input().split()))
re = 1
def duan(x,me):
    global re
    re = max(re,me)
    a = dt[x][0]
    b = dt[x][1]
    if a != -1:
        duan(a,me+1)
    if b != -1:
        duan(b,me+1)
if dt is None:
    print(0)
```

```
else:
    duan(1,1)
    print(re)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态：Accepted

源代码

```
n = int(input())
dt = {i:[] for i in range(1,n+1)}
for j in range(1,n+1):
    dt[j] = list(map(int,input().split()))
re = 1
def duan(x,me):
    global re
    re = max(re,me)
    a = dt[x][0]
    b = dt[x][1]
    if a != -1:
        duan(a,me+1)
    if b != -1:
        duan(b,me+1)
if dt is None:
    print(0)
else:
    duan(1,1)
    print(re)
```

基本信息

#:	44300002
题目:	06646
提交人:	23n2300012140(zyt)
内存:	3912kB
时间:	24ms
语言:	Python3
提交时间:	2024-03-19 16:55:26

02299: Ultra-QuickSort

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

思路：没有什么思路，只好用的答案

代码

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

```
            j += 1
            inv_count += len(left) - i
    merged += left[i:]
    merged += right[j:]
    return merged, inv_count
while True:
    n = int(input())
    if n == 0:
        break
    lst = []
    for _ in range(n):
        lst.append(int(input()))
    _, inversions = merge_sort(lst)
    print(inversions)
```

代码运行截图 ==（AC代码截图，至少包含有"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。==

对数算里的新事物理解得还很不深入