Assignment #4: 排序、栈、队列和树

Updated 0005 GMT+8 March 11, 2024

2024 spring, Complied by ==狄晨阳 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：Windows11

Python编程环境：Spyder IDE 5.4.3

C/C++编程环境：无

1. 题目

05902: 双端队列

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

思路：按照读取的数据分类处理即可

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Thu Mar 14 18:47:43 2024

@author: 20311
"""
```

```
t=int(input())
for _ in range(t):
    n=int(input())
    l=[]
    for z in range(n):
        a,b=map(int,input().split())
        if a==1:
            l.append(b)
        elif a==2:
            if b==0:
                del l[0]
            elif b==1:
                del l[-1]
    if len(l)!=0:
        print(' '.join(map(str,l)))
    else:
        print('NULL')
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Thu Mar 14 18:47:43 2024

@author: 20311
"""

t=int(input())
for _ in range(t):
    n=int(input())
    l=[]
    for z in range(n):
        a,b=map(int,input().split())
        if a==1:
            l.append(b)
        elif a==2:
            if b==0:
                del l[0]
            elif b==1:
                del l[-1]
    if len(l)!=0:
        print(' '.join(map(str,l)))
    else:
        print('NULL')
```

基本信息

#:	44216458
题目:	05902
提交人:	23n2300012138(yukino)
内存:	3712kB
时间:	41ms
语言:	Python3
提交时间:	2024-03-14 18:53:07

02694: 波兰表达式

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

思路：学习了一下题解中用栈来处理的做法，并学习了一下eval函数的用法

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Thu Mar 14 18:54:01 2024
```

```
@author: 20311
"""

l=input().split()
stack=[]
while len(l)>0:
    a=l.pop(-1)
    if a in '+-*/':
        b=stack.pop(-1)
        c=stack.pop(-1)
        stack.append(str(eval(b+a+c)))
    else:
        stack.append(a)
print('{:.6f}'.format(float(stack[0])))
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Thu Mar 14 18:54:01 2024

@author: 20311
"""

l=input().split()
stack=[]
while len(l)>0:
    a=l.pop(-1)
    if a in '+-*/':
        b=stack.pop(-1)
        c=stack.pop(-1)
        stack.append(str(eval(b+a+c)))
    else:
        stack.append(a)
print('{:.6f}'.format(float(stack[0])))
```

基本信息

#:	44216988
题目:	02694
提交人:	23n2300012138(yukino)
内存:	3608kB
时间:	22ms
语言:	Python3
提交时间:	2024-03-14 19:21:24

24591: 中序表达式转后序表达式

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

思路：实在是无从下手于是看了一下题解中的思路，按着思路写了一遍，可以完成样例，但代码又臭又长，而且提交的时候还一直RE，最后debug很久才发现是while后面的条件中and前后的条件顺序放反了，没有先判断列表是否为空就判断了其最后一项的性质导致报错，然后也是跟着题解改良了一下代码结构最后成了这样

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Thu Mar 14 19:26:02 2024

@author: 20311
```

```
"""
def compare(a,b):
    if a in '*/' and b in '+-':
        return False
    else:
        return True
def add():
    if num_buffer:
        stack_out.append(''.join(num_buffer))
        num_buffer.clear()
n=int(input())
for _ in range(n):
    l=list(input())
    stack=[]
    stack_out=[]
    num_buffer=[]
    for a in l:
        if a=='(':
            add()
            stack.append(a)
        elif a==')':
            add()
            while stack and stack[-1]!='(':
                stack_out.append(stack.pop())
            stack.pop()
        elif a in '+-*/':
            add()
            while stack and compare(a,stack[-1]) and stack[-1]!='(' :
                stack_out.append(stack.pop())
            stack.append(a)
        elif a=='.' or a in '0123456789':
            num_buffer.append(a)
    add()
    while stack:
        stack_out.append(stack.pop())
    print(' '.join(stack_out))
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态: Accepted

基本信息

#: 44236149
题目: 24591
提交人: 23n2300012138(yukino)
内存: 3672kB
时间: 28ms
语言: Python3
提交时间: 2024-03-16 01:04:05

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Thu Mar 14 19:26:02 2024

@author: 20311
"""
def compare(a,b):
    if a in '*/' and b in '+-':
        return False
    else:
        return True
def add():
    if num_buffer:
        stack_out.append(''.join(num_buffer))
        num_buffer.clear()
n=int(input())
for _ in range(n):
    l=list(input())
    stack=[]
    stack_out=[]
    num_buffer=[]
    for a in l:
        if a=='(':
            add()
            stack.append(a)
        elif a==')':
            add()
            while stack and stack[-1]!='(':
                stack_out.append(stack.pop())
            stack.pop()
        elif a in '+-*/':
            add()
            while stack and compare(a,stack[-1]) and stack[-1]!='(' :
                stack_out.append(stack.pop())
            stack.append(a)
        elif a=='.' or a in '0123456789':
            num_buffer.append(a)
    add()
    while stack:
        stack_out.append(stack.pop())
    print(' '.join(stack_out))
```

22068: 合法出栈序列

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

思路：本想通过模拟之外的方式来完成但很难考虑到所有情况，最后还是写了模拟的程序来判断。

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Sun Mar 17 00:59:13 2024

@author: 20311
"""
def judge(s):
    if len(x)!=len(s):
        return False
    y=list(x)
    stack=[]
    for i in s:
```

```
        while (not stack or stack[-1]!=i) and y:
            stack.append(y.pop(0))

        if stack and stack[-1]!=i:
            return False

        stack.pop()
    return True


x=input()
d={x[i]:i for i in range(len(x))}
while True:
    try:
        if judge(input()):
            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

```
# -*- coding: utf-8 -*-
"""
Created on Sun Mar 17 00:59:13 2024

@author: 20311
"""
def judge(s):
    if len(x)!=len(s):
        return False
    y=list(x)
    stack=[]
    for i in s:
        while (not stack or stack[-1]!=i) and y:
            stack.append(y.pop(0))

        if stack and stack[-1]!=i:
            return False

        stack.pop()
    return True


x=input()
d={x[i]:i for i in range(len(x))}
while True:
    try:
        if judge(input()):
            print('YES')
        else:
            print('NO')

    except EOFError:
        break
```

基本信息

#:	44259454
题目:	22068
提交人:	23n2300012138(yukino)
内存:	3640kB
时间:	24ms
语言:	Python3
提交时间:	2024-03-17 01:39:20

06646: 二叉树的深度

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

思路：读取数据用字典构成的链表储存二叉树，然后遍历，额外用一个参数记录当前深度，用一个全局变量来记录最大深度最后输出

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Sun Mar 17 01:44:00 2024

@author: 20311
"""

n=int(input())
d={x:[] for x in range(1,n+1)}
for x in range(1,n+1):
    d[x]=list(map(int,input().split()))

dp=1
def go(a,b):
    global dp
    dp=max(dp,b)
    i=d[a][0]
    j=d[a][1]
    if i!=-1:
        go(i,b+1)
    if j!=-1:
        go(j,b+1)

go(1,1)
print(dp)
```

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 Sun Mar 17 01:44:00 2024

@author: 20311
"""

n=int(input())
d={x:[] for x in range(1,n+1)}
for x in range(1,n+1):
    d[x]=list(map(int,input().split()))

dp=1
def go(a,b):
    global dp
    dp=max(dp,b)
    i=d[a][0]
    j=d[a][1]
    if i!=-1:
        go(i,b+1)
    if j!=-1:
        go(j,b+1)

go(1,1)
print(dp)
```

基本信息

#:	44259494
题目:	06646
提交人:	23n2300012138(yukino)
内存:	3620kB
时间:	22ms
语言:	Python3
提交时间:	2024-03-17 01:55:32

02299: Ultra-QuickSort

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

思路：本想使用数学方法但一直TLE便看了题解，学习了一下归并排序的算法

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Mon Mar 18 14:37:01 2024

@author: 20311
"""

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"）==

状态: Accepted

源代码

```python
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 18 14:37:01 2024

@author: 20311
"""

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)
```

基本信息

#:	44284914
题目:	02299
提交人:	23n2300012138(yukino)
内存:	32348kB
时间:	3927ms
语言:	Python3
提交时间:	2024-03-18 14:38:10

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。==

有

本次的题目难度明显提高，需要花更多的时间来学习，同时也接触到了一些新的算法，debug中也发现了一些以后也可能遇到的错误类型，总而言之还是要多练多想