Assignment #4: 排序、栈、队列和树

Updated 0005 GMT+8 March 11, 2024

2024 spring, Complied by 赵语涵 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：windows11

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：就是把list写成类然后通过pop(0)等方法模拟deque

代码

```
#赵语涵2300012254

class Deque:
    def __init__(self):
        self.items = []
    def left(self):
        self.items.pop(0)
    def right(self):
        return self.items.pop()
    def putin(self,value):
        self.items.append(value)
```

```
12
13  for t in range(int(input())):
14      num = Deque()
15      for n in range(int(input())):
16          answer = None
17          x,y = map(int,input().split())
18          if x == 1:
19              num.putin(y)
20          elif x== 2:
21              if y == 0:
22                  num.left()
23              else:
24                  num.right()
25      if num.items:
26          print(' '.join(map(str,num.items)))
27      else:
28          print('NULL')
```

代码运行截图

#44172369提交状态

查看　提交　统计

状态: Accepted

源代码

```
#赵语涵2300012254

class Deque:
    def __init__(self):
        self.items = []
    def left(self):
        self.items.pop(0)
```

基本信息

#:	44172369
题目:	05902
提交人:	23n2300012254
内存:	3664kB
时间:	43ms
语言:	Python3
提交时间:	2024-03-11 18:59:27

02694: 波兰表达式

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

思路：运用递归的方法，从前向后遍历，遇到运算符时计算其后2个数字，如果不是数字继续向后找新运算符的数字进行计算

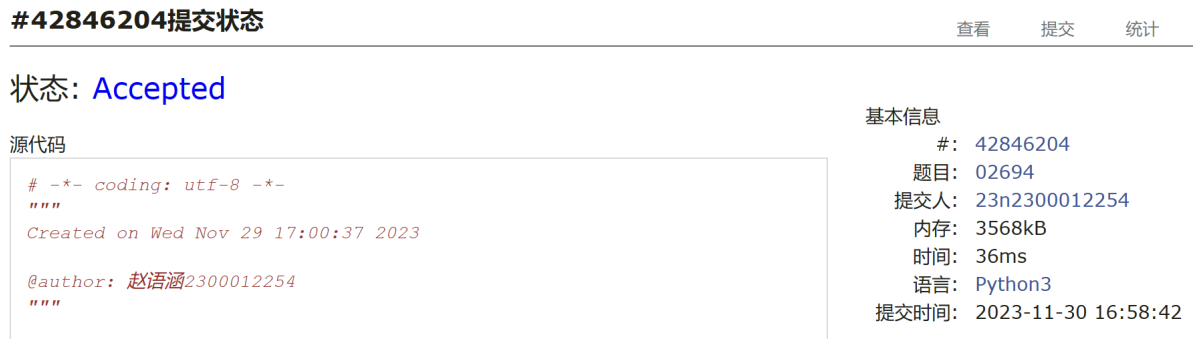
代码

```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Wed Nov 29 17:00:37 2023
4
5  @author: 赵语涵2300012254
6  """
7
8  def cal(a,x,y):
9      if a == '+':
```

```
        return x+y
    elif a == '-':
        return x-y
    elif a == '*':
        return x*y
    elif a == '/':
        return x/y

def ope(ori,a):
    x,y = ori[a+1],ori[a+2]
    try:
        x = float(x)
        try:
            y = float(y)
            ori[a] = cal(ori[a],x,y)
            ori.remove(ori[a+1])
            ori.remove(ori[a+1])
        except:
            ori = ope(ori,a+2)
            ori = ope(ori,a)
    except:
        ori = ope(ori,a+1)
        ori = ope(ori,a)
    return ori
ori = list(input().split())
answer = ope(ori,0)[0]
print('%.6f'%answer)
```

代码运行截图



#42846204提交状态

查看 提交 统计

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Wed Nov 29 17:00:37 2023

@author: 赵语涵2300012254
"""
```

基本信息

#: 42846204
题目: 02694
提交人: 23n2300012254
内存: 3568kB
时间: 36ms
语言: Python3
提交时间: 2023-11-30 16:58:42

24591: 中序表达式转后序表达式

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

思路：这道题感觉还挺麻烦的，是在看数算书看到这一章节时一起做的。用list可以append与pop的队列特点，将数字以字符串形式放入列表（没有计算就没有int必要，况且有小数），在遇到运算符时，向结果中放入运算符直到弹出的运算符等级更小，遇到括号时，结果中加入左右括号中所有运算符。最后把剩余的运算符全部添加到输出结果后

代码

```python
#赵语涵2300012254
op,comp = ['+','-','*','/','(',')'],{'(':1,'+':2,'-':2,'*':3,'/':3}
def change(x):
    results = []
    ind,ops,stop = 0,[],False
    while ind < len(x):
        num = ''
        while (a:=x[ind]) not in op:
            num += a
            ind += 1
            if ind==len(x):
                stop = True
                break
        if num != '':
            results.append(num)
        if stop:
            break
        if a == ')':
            while (b:=ops.pop()) != '(':
                results.append(b)
        elif a == '(':
            ops.append(a)
        else:
            while True:
                if ops == []:
                    break
                if comp[(b:=ops.pop())]>=comp[a]:
                    results.append(b)
                else:
                    ops.append(b)
                    break
            ops.append(a)
        ind += 1
    while ops:
        results.append(ops.pop())
    return ' '.join(results)
n = int(input())
for i in range(n):
    print(change(input()))
```

代码运行截图

#43991889提交状态

查看 提交 统计

状态: Accepted

源代码

```
#赵语涵2300012254
op,comp = ['+','-','*','/','(',')'],{'(':1,'+':2,'-':2,'*':3,'/':3}
def change(x):
    results = []
    ind,ops,stop = 0,[],False
    while ind < len(x):
        num = ''
```

基本信息

#: 43991889
题目: 24591
提交人: 23n2300012254
内存: 4844kB
时间: 28ms
语言: Python3
提交时间: 2024-02-26 19:30:33

22068: 合法出栈序列

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

思路：对于一个进栈的字符，有弹出和保留2种可能，由于可能有很多测试数据，最开始的思路是像八皇后一样用递归枚举出所有合法排列，但是最终MLE了；对于每个单独考虑，用双指针的方法一一比对原序列和待检序列

代码

```
#赵语涵2300012254
ori = input()
l = len(ori)
while True:
    try:
        test = input()
        i,j,stack,stop = 0,0,[],False
        while i < l:
            if len(test) != l:
                break
            while ori[i] != test[j]:
                stack.append(ori[i])
                i += 1
                if i == l:
                    stop = True
                    break
            if stop:
                j==0
                break
            stack.append(ori[i])
            while stack:
                k = stack.pop()
                if k == test[j]:
                    j += 1
                else:
                    stack.append(k)
                    break
            i += 1
        print(['NO','YES'][j==l])
```

```
    except:
        break
```

代码运行截图

#44187333提交状态

查看 提交 统计

状态: Accepted

源代码

```
#赵语涵2300012254
ori = input()
l = len(ori)
while True:
    try:
        test = input()
        i,j,stack,stop = 0,0,[],False
```

基本信息

#: 44187333
题目: 22068
提交人: 23n2300012254
内存: 3664kB
时间: 25ms
语言: Python3
提交时间: 2024-03-12 19:54:59

06646: 二叉树的深度

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

思路：class用于对一个节点进行定义（有左、右子树），类似于用了很多的相互指向的字典的感觉。对于树高度的计算，沿着边一直到没有子树为止并记录节点数量。注意树的高度不包括根节点而深度包括根节点

代码

```
#赵语涵2300012254
class Tree:
    def __init__(self):
        self.left = None
        self.right = None
def height(node):
    if node == None:
        return 0
    else:
        return max(height(node.left),height(node.right))+1

n = int(input())
t = [Tree() for _ in range(n)]
for i in range(n):
    l_i,r_i = map(int,input().split())
    if l_i != -1:
        t[i].left = t[l_i-1]
    if r_i != -1:
        t[i].right = t[r_i-1]
print(height(t[0]))
```

代码运行截图

#44210342提交状态

查看 提交 统计

状态: Accepted

源代码

```
#赵语涵2300012254
class Tree:
    def __init__(self):
        self.left = None
        self.right = None
def height(node):
    if node == None:
```

基本信息

#: 44210342
题目: 06646
提交人: 23n2300012254
内存: 3648kB
时间: 23ms
语言: Python3
提交时间: 2024-03-14 13:06:11

02299: Ultra-QuickSort

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

思路：对于本题要计算交换的思路，按题面直接翻译做题想到的是用冒泡排序，然而此做法会超时或超空间。群里有讨论使用递归排序不过没有想到怎么写它的代码，于是学习了题解中的写法。其中优化步骤是在需要将靠左的某一项交换到后面时，由于保证了mid到left[i]和right[j]都小于left[i]，于是直接count+=len(left)-i即可。

代码

```
#赵语涵2300012254
def merge_sort(x):
    l = len(x)
    if l <= 1:
        return x,0
    mid = l//2
    left,right = x[:mid],x[mid:]
    left,c_l = merge_sort(left)
    right,c_r = merge_sort(right)
    merged,c = merge(left,right)
    return merged,c_l+c+c_r

def merge(left,right):
    i,j = 0,0
    merged = []
    count = 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
            count += len(left)-i
    merged += left[i:]
    merged += right[j:]
    return merged,count

```

```python
while True:
    t = int(input())
    if t == 0:
        break
    x = [int(input()) for _ in range(t)]
    print(merge_sort(x)[1])
```

代码运行截图

#44228600提交状态 查看 提交 统计

状态: Accepted

源代码

```python
#赵语涵2300012254
def merge_sort(x):
    l = len(x)
    if l <= 1:
        return x,0
    mid = l//2
    left,right = x[:mid],x[mid:]
```

基本信息

#: 44228600
题目: 02299
提交人: 23n2300012254
内存: 32556kB
时间: 3853ms
语言: Python3
提交时间: 2024-03-15 16:42:19

2. 学习总结和收获

关于树类型的题，基本是看了课件才会做（也是做的第一个建树的题），对树的结构与python表示方式有了大致的了解，但是不是很熟练地应用。排序题目也是同理，虽然在书上看到了各种排序的类型但是并未进行针对练习。后续需要多练习相关类型题进行补充

3.额外练习题目

02783:Holiday Hotel

OpenJudge - 02783:Holiday Hotel

思路：将距离排序（可以用heapq或者sort），从小到大遍历/弹出以后如果价格比当前的最小价格小，说明该价格没有在其更近距离内更小价格的hotel存在，那么可选hotel数+1.

代码

```python
#赵语涵2300012254
import heapq
while True:
    n = int(input())
    if n == 0:
        break
    hotels = [];heapq.heapify(hotels)
    for i in range(n):
        heapq.heappush(hotels, list(map(int,input().split())))
    small,num = 10001,0
    while True:
```

```
        try:
            a = heapq.heappop(hotels)
            x = a[1]
            if x < small:
                small = x
                num += 1
        except:
            print(num)
            break
```

代码运行截图

#44167129提交状态

查看 提交 统计

状态: Accepted

源代码

```
#赵语涵2300012254
import heapq
while True:
    n = int(input())
    if n == 0:
        break
    hotels = [];heapq.heapify(hotels)
```

基本信息

#: 44167129
题目: 02783
提交人: 23n2300012254
内存: 6392kB
时间: 357ms
语言: Python3
提交时间: 2024-03-11 11:07:27