Assignment #4: 排序、栈、队列和树

Updated 0005 GMT+8 March 11, 2024

2024 spring, Complied by 田济维 物理学院

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

（python pycharm）

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

```
#
from collections import deque
n = int(input())
for i in range(n):
    k = int(input())
```

```
    temp = deque()
    for j in range(k):
        a,b = map(str,input().split())
        if a== "1":
            temp.append(b)
        elif a == "2":
            if b == "0":
                temp.popleft()
            elif b == "1":
                temp.pop()
    if len(temp)!=0:
        print(" ".join(temp))
    else:
        print("NULL")
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
from collections import deque
n = int(input())
for i in range(n):
    k = int(input())
    temp = deque()
    for j in range(k):
        a,b = map(str,input().split())
        if a== "1":
            temp.append(b)
        elif a == "2":
            if b == "0":
                temp.popleft()
            elif b == "1":
                temp.pop()
    if len(temp)!=0:
        print(" ".join(temp))
    else:
        print("NULL")
```

基本信息

#:	44289455
题目:	05902
提交人:	23n2300011503
内存:	3652kB
时间:	41ms
语言:	Python3
提交时间:	2024-03-18 20:13:30

02694: 波兰表达式

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

思路:

代码

```
#
s = input().split()

operator = ["+","-","*","/"]
pstack = []
while len(s):
    a = s.pop()
    if a in operator:
```

```
        y1 = pstack.pop()
        y2 = pstack.pop()
        pstack.append(str(eval(y1+a+y2)))
    else:
        pstack.append(a)

h = pstack[0]
print("%.6f"%float(h))
```

代码运行截图 （至少包含有"Accepted"）

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,"(":0}
    op = []
    output = []
    number = []

    for x in expression:
        if x.isdigit() or x == ".":
            number.append(x)
        elif x in "+-*/":
            if number:
                output.append("".join(number))
                number.clear()
            while op and precedence[op[-1]]>=precedence[x]:
                output.append(op.pop())
```

```
            op.append(x)
        elif x == "(":

            op.append(x)
        elif x == ")":
            if number:
                output.append("".join(number))
                number.clear()
            while op and op[-1]!="(":
                output.append(op.pop())
            op.pop()
    if number:
        output.append("".join(number))
        number.clear()
    output.extend(op[::-1])
    return output

n = int(input())
for _ in range(n):
    temp  = infix_to_postfix(input())
    print(" ".join(temp))
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

#44290383提交状态

查看 提交 统计 提问

状态: Accepted

基本信息

#: 44290383
题目: 24591
提交人: 23n2300011503
内存: 3668kB
时间: 26ms
语言: Python3
提交时间: 2024-03-18 20:53:13

源代码

```
def infix_to_postfix(expression):
    precedence = {"+":1,"-":1,"*":2,"/":2,"(":0}
    op = []
    output = []
    number = []

    for x in expression:
        if x.isdigit() or x == ".":
            number.append(x)
        elif x in "+-*/":
            if number:
                output.append("".join(number))
                number.clear()
            while op and precedence[op[-1]]>=precedence[x]:
                output.append(op.pop())
            op.append(x)
        elif x == "(":

            op.append(x)
        elif x == ")":
            if number:
                output.append("".join(number))
                number.clear()
            while op and op[-1]!="(":
                output.append(op.pop())
            op.pop()
    if number:
        output.append("".join(number))
        number.clear()
    output.extend(op[::-1])
    return output

n = int(input())
for _ in range(n):
    temp  = infix_to_postfix(input())
    print(" ".join(temp))
```


English 帮助 关于

22068: 合法出栈序列

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

思路:

代码

```
#
s = input()

while True:
    try:
        k =input()
    except EOFError:
        break
    else:
        if len(s)==len(k):
            pstack = []
            index = 0
            for x in s:
                if x != k[index]:
                    pstack.append(x)
                else:
                    index+=1
                    while pstack and index<len(k):
                        if pstack[-1]==k[index]:
                            pstack.pop()
                            index+=1
                        else:
                            break
            if pstack:
                pstack.reverse()
                if pstack == k[index:]:
                    print("YES")
                else:
                    print("NO")
            else:
                print("YES")
        else:
            print("NO")
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

状态: Accepted

源代码

```
s = input()

while True:
    try:
        k =input()
    except EOFError:
        break
    else:
        if len(s)==len(k):
            pstack = []
            index = 0
            for x in s:
                if x != k[index]:
                    pstack.append(x)
                else:
                    index+=1
                    while pstack and index<len(k):
                        if pstack[-1]==k[index]:
                            pstack.pop()
                            index+=1
                        else:
                            break
            if pstack:
                pstack.reverse()
                if pstack == k[index:]:
                    print("YES")
                else:
                    print("NO")
            else:
                print("YES")
        else:
            print("NO")
```

基本信息

#: 44295219
题目: 22068
提交人: 23n2300011503
内存: 3624kB
时间: 24ms
语言: Python3
提交时间: 2024-03-19 10:33:00

06646: 二叉树的深度

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

思路:

代码

```
#
class Node:
    def __init__(self):
        self.left = None
        self.right = None

def Treedepth(node):
    if node == None:
        return 0
    else:
        return max(Treedepth(node.left),Treedepth(node.right))+1

n = int(input())

Nodes = [Node() for i in range(n)]
parents = [True for i in range(n)]
for i in range(n):
    l,r = map(int,input().split())
    if l !=-1:
```

```
        Nodes[i].left = Nodes[l-1]
        parents[l-1]=False
    if r !=-1:
        Nodes[i].right = Nodes[r-1]
        parents[r-1]=False
s = parents.index(True)
print(Treedepth(Nodes[s]))
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

状态: Accepted

源代码

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

def Treedepth(node):
    if node == None:
        return 0
    else:
        return max(Treedepth(node.left),Treedepth(node.right))+1

n = int(input())

Nodes = [Node() for i in range(n)]
parents = [True for i in range(n)]
for i in range(n):
    l,r = map(int,input().split())
    if l !=-1:
        Nodes[i].left = Nodes[l-1]
        parents[l-1]=False
    if r !=-1:
        Nodes[i].right = Nodes[r-1]
        parents[r-1]=False
s = parents.index(True)
print(Treedepth(Nodes[s]))
```

基本信息

#: 43968032
题目: 06646
提交人: 23n2300011503
内存: 3648kB
时间: 21ms
语言: Python3
提交时间: 2024-02-23 15:33:21

02299: Ultra-QuickSort

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

思路:

代码

```
#
def merge_sort(que):
    l = len(que)
    if l>1:
        left = que[:l//2]
        right = que[l//2:]
        a = merge_sort(left)
        b = merge_sort(right)
        il = 0
        ir = 0
        k = 0
```

```
        cnt = a+b

        while il<len(left) and ir<len(right):
            if left[il]<=right[ir]:
                que[k]=left[il]
                k+=1
                il+=1
            else:
                que[k]=right[ir]
                cnt += (len(left) - il)
                k+=1
                ir+=1

        while il<len(left):
            que[k]=left[il]
            il+=1
            k+=1
        while ir<len(right):
            que[k]=right[ir]
            ir+=1
            k+=1
        return cnt
    return 0

while True:
    n = int(input())
    if n == 0:
        break
    que1 = []
    for i in range(n):
        que1.append(int(input()))
    print(merge_sort(que1))


```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

状态：Accepted

源代码

```
def merge_sort(que):
    l = len(que)
    if l>1:
        left = que[:l//2]
        right = que[l//2:]
        a = merge_sort(left)
        b = merge_sort(right)
        il = 0
        ir = 0
        k = 0
        cnt = a+b

        while il<len(left) and ir<len(right):
            if left[il]<=right[ir]:
                que[k]=left[il]
                k+=1
                il+=1
            else:
                que[k]=right[ir]
                cnt += (len(left) - il)
                k+=1
                ir+=1

        while il<len(left):
            que[k]=left[il]
            il+=1
            k+=1
        while ir<len(right):
            que[k]=right[ir]
            ir+=1
            k+=1
        return cnt
    return 0

while True:
    n = int(input())
```

基本信息

#:	44295782
题目:	02299
提交人:	23n2300011503
内存:	28452kB
时间:	3832ms
语言:	Python3
提交时间:	2024-03-19 11:17:46

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。

逐渐有点吃力了，需要加大对数算的投入时间。