

Assignment #4: 排序、栈、队列和树

Updated 0005 GMT+8 March 11, 2024

2024 spring, Compiled 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

```
1 #
2 from collections import deque
3 n = int(input())
4 for i in range(n):
5     k = int(input())
```

```

6     temp = deque()
7     for j in range(k):
8         a,b = map(str,input().split())
9         if a== "1":
10            temp.append(b)
11        elif a == "2":
12            if b == "0":
13                temp.popleft()
14            elif b == "1":
15                temp.pop()
16        if len(temp)!=0:
17            print(" ".join(temp))
18        else:
19            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

```

1  #
2  s = input().split()
3
4  operator = ["+", "-", "*", "/"]
5  pstack = []
6  while len(s):
7      a = s.pop()
8      if a in operator:

```

```

9         y1 = pstack.pop()
10        y2 = pstack.pop()
11        pstack.append(str(eval(y1+a+y2)))
12    else:
13        pstack.append(a)
14
15    h = pstack[0]
16    print("%.6f"%float(h))

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

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))

```

基本信息

#: 44289930

题目: 02694

提交人: 23n2300011503

内存: 3572kB

时间: 23ms

语言: Python3

提交时间: 2024-03-18 20:33:07

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

24591: 中序表达式转后序表达式

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

思路:

代码

```

1  #
2  def infix_to_postfix(expression):
3      precedence = {"+":1, "-":1, "*":2, "/":2, "(":0}
4      op = []
5      output = []
6      number = []
7
8      for x in expression:
9          if x.isdigit() or x == ".":
10             number.append(x)
11          elif x in "+-*/":
12             if number:
13                 output.append("".join(number))
14                 number.clear()
15             while op and precedence[op[-1]]>=precedence[x]:
16                 output.append(op.pop())

```

```

17         op.append(x)
18     elif x == "(":
19
20         op.append(x)
21     elif x == ")":
22         if number:
23             output.append("".join(number))
24             number.clear()
25             while op and op[-1]!="(":
26                 output.append(op.pop())
27             op.pop()
28     if number:
29         output.append("".join(number))
30         number.clear()
31     output.extend(op[::-1])
32     return output
33
34 n = int(input())
35 for _ in range(n):
36     temp = infix_to_postfix(input())
37     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

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))

```

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

22068: 合法出栈序列

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

思路:

代码

```
1  #
2  s = input()
3
4  while True:
5      try:
6          k = input()
7      except EOFError:
8          break
9      else:
10         if len(s)==len(k):
11             pstack = []
12             index = 0
13             for x in s:
14                 if x != k[index]:
15                     pstack.append(x)
16                 else:
17                     index+=1
18                     while pstack and index<len(k):
19                         if pstack[-1]==k[index]:
20                             pstack.pop()
21                             index+=1
22                     else:
23                         break
24             if pstack:
25                 pstack.reverse()
26                 if pstack == k[index:]:
27                     print("YES")
28                 else:
29                     print("NO")
30             else:
31                 print("YES")
32         else:
33             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

```
1 #
2 class Node:
3     def __init__(self):
4         self.left = None
5         self.right = None
6
7 def Treedepth(node):
8     if node == None:
9         return 0
10    else:
11        return max(Treedepth(node.left), Treedepth(node.right)) + 1
12
13 n = int(input())
14
15 Nodes = [Node() for i in range(n)]
16 parents = [True for i in range(n)]
17 for i in range(n):
18     l, r = map(int, input().split())
19     if l != -1:
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

02299: Ultra-QuickSort

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

思路：

代码

```
11      k = 0
```

```

12         cnt = a+b
13
14         while il<len(left) and ir<len(right):
15             if left[il]<=right[ir]:
16                 que[k]=left[il]
17                 k+=1
18                 il+=1
19             else:
20                 que[k]=right[ir]
21                 cnt += (len(left) - il)
22                 k+=1
23                 ir+=1
24
25         while il<len(left):
26             que[k]=left[il]
27             il+=1
28             k+=1
29         while ir<len(right):
30             que[k]=right[ir]
31             ir+=1
32             k+=1
33         return cnt
34     return 0
35
36 while True:
37     n = int(input())
38     if n == 0:
39         break
40     que1 = []
41     for i in range(n):
42         que1.append(int(input()))
43     print(merge_sort(que1))
44
45

```

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())
    if n == 0:
        break
    que = list(map(int, input().split()))
    print(merge_sort(que))
```

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。