

# Assignment #4: 排序、栈、队列和树

---

Updated 0005 GMT+8 March 11, 2024

2024 spring, Compiled by 夏天明 元培学院

## 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: Windows 10 | 22H2

Python编程环境: Spyder IDE 5.4.3 | Python 3.11.4 64-bit

## 1. 题目

---

### 05902: 双端队列

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

思路: 直接实现

代码

```

from collections import deque

for o in range(int(input())):
    q = deque()
    for w in range(int(input())):
        t, x = map(int, input().split())
        if t == 1:
            q.append(x)
        elif x:
            q.pop()
        else:
            q.popleft()
    print(' '.join(map(str, q)) if q else 'NULL')

```

代码运行截图

#44182717提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

from collections import deque

for o in range(int(input())):
    q = deque()
    for w in range(int(input())):
        t, x = map(int, input().split())
        if t == 1:
            q.append(x)
        elif x:
            q.pop()
        else:
            q.popleft()
    print(' '.join(map(str, q)) if q else 'NULL')

```

基本信息

#: 44182717

题目: 05902

提交人: 23n2300017735(夏天明

BrightSummer)

内存: 3604kB

时间: 38ms

语言: Python3

提交时间: 2024-03-12 15:42:46

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

## 02694: 波兰表达式

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

思路: 现建树再计算

代码

```

ops = {"+", "-", "*", "/"}

class Tree:
    def __init__(self, value=None):
        self.val = value
        self.left = None

```

```

        self.right = None

    def evaluate(self):
        if self.val not in ops:
            return self.val
        else:
            return eval(str(self.left.evaluate()) + self.val +
str(self.right.evaluate()))

    def buildFromPoland():
        root = Tree(po.pop())
        if root.val in ops:
            root.left = buildFromPoland()
            root.right = buildFromPoland()
        return root

po = input().split()
po.reverse()
print(f"{buildFromPoland().evaluate():.6f}")

```

代码运行截图

#42806601提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

ops = {"+", "-", "*", "/"}

class Tree:
    def __init__(self, value=None):
        self.val = value
        self.left = None
        self.right = None

    def evaluate(self):
        if self.val not in ops:
            return self.val
        else:
            return eval(str(self.left.evaluate()) + self.val + str(self
            .right.evaluate()))

    def buildFromPoland():
        root = Tree(po.pop())
        if root.val in ops:
            root.left = buildFromPoland()
            root.right = buildFromPoland()
        return root

po = input().split()
po.reverse()
print(f"{buildFromPoland().evaluate():.6f}")

```

基本信息

#: 42806601  
 题目: 02694  
 提交人: 23n2300017735(夏天明  
 BrightSummer)  
 内存: 3584kB  
 时间: 33ms  
 语言: Python3  
 提交时间: 2023-11-28 16:02:36

## 24591: 中序表达式转后序表达式

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

思路：利用栈实现

代码

```
import re

pre = {'(':1, '+':2, '-':2, '*':3, '/':3}
for o in range(int(input())):
    s = re.split(r"([^\d.])", input())
    ans = []
    op = []
    for token in s:
        if token == '(':
            op.append('(')
        elif token == ')':
            while (t:=op.pop()) != '(':
                ans.append(t)
        elif token and token in '+-*/':
            while op and pre[op[-1]] >= pre[token]:
                ans.append(op.pop())
            op.append(token)
        elif token:
            ans.append(token)
    while op:
        ans.append(op.pop())
    print(*ans)
```

代码运行截图

状态: Accepted

源代码

```
import re

pre = {'(':1, '+':2, '-':2, '*':3, '/':3}
for o in range(int(input())):
    s = re.split(r"([\d.])", input())
    ans = []
    op = []
    for token in s:
        if token == '(':
            op.append('(')
        elif token == ')':
            while (t:=op.pop()) != '(':
                ans.append(t)
        elif token and token in '+*/*':
            while op and pre[op[-1]] >= pre[token]:
                ans.append(op.pop())
            op.append(token)
        elif token:
            ans.append(token)
    while op:
        ans.append(op.pop())
    print(*ans)
```

基本信息

#: 43862364  
题目: 24591  
提交人: 23n2300017735(夏天明  
BrightSummer)  
内存: 4344kB  
时间: 32ms  
语言: Python3  
提交时间: 2024-02-05 11:25:03

## 22068: 合法出栈序列

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

思路：出栈序列是合法的当且仅当不存在cab这样的结构

代码

```
def judge(s):
    if set(s) != set(x) or len(set(s)) != len(s):
        return "NO"
    for i, char in enumerate(s):
        pre = float("inf")
        for new in s[i+1:]:
            if idx[new] < idx[char]:
                if idx[new] > pre:
                    return "NO"
                pre = idx[new]
        return "YES"

x = input()
idx = {char:i for i, char in enumerate(x)}
while True:
    try:
        s = input()
    except EOFError:
        break
```

```
print(judge(s))
```

代码运行截图

状态: **Accepted**

源代码

```
def judge(s):
    if set(s) != set(x) or len(set(s)) != len(s):
        return "NO"
    for i, char in enumerate(s):
        pre = float("inf")
        for new in s[i+1:]:
            if idx[new] < idx[char]:
                if idx[new] > pre:
                    return "NO"
                pre = idx[new]
        return "YES"

x = input()
idx = {char:i for i, char in enumerate(x)}
while True:
    try:
        s = input()
    except EOFError:
        break
    print(judge(s))
```

基本信息

#: 43639671  
题目: 22068  
提交人: 23n2300017735(夏  
BrightSummer)  
内存: 3604kB  
时间: 32ms  
语言: Python3  
提交时间: 2024-01-17 23:10:5

## 06646: 二叉树的深度

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

思路: 直接往上擦

代码

```
n = int(input())
h = [0]*11
for i in range(1, n+1):
    for child in map(int, input().split()):
        if child != -1:
            h[child] = h[i] + 1
print(max(h)+1)
```

代码运行截图

状态: Accepted

源代码

```
n = int(input())
h = [0]*11
for i in range(1, n+1):
    for child in map(int, input().split()):
        if child != -1:
            h[child] = h[i] + 1
print(max(h)+1)
```

基本信息

#: 43730173  
题目: 06646  
提交人: 23n2300017735(夏天明  
BrightSummer)  
内存: 3884kB  
时间: 21ms  
语言: Python3  
提交时间: 2024-01-26 15:59:08

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

## 02299: Ultra-QuickSort

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

思路: 归并排序求逆序数

代码

```
ans = 0
def merge(l, mid, r):
    i, j = l, mid+1
    tmp = []
    while i<=mid or j<=r:
        if j>r or (i<=mid and arr[i]<arr[j]):
            tmp.append(arr[i])
            i += 1
        else:
            tmp.append(arr[j])
            global ans
            ans += j-(l+len(tmp)-1)
            j += 1
    arr[l:r+1] = tmp

def m_sort(l, r):
    if l == r:
        return
    mid = (l+r)//2
    m_sort(l, mid)
    m_sort(mid+1, r)
    merge(l, mid, r)

while n := int(input()):
    arr = [int(input()) for i in range(n)]
    ans = 0
    m_sort(0, n-1)
```

```
print(ans)
```

代码运行截图

状态: Accepted

源代码

```
ans = 0
def merge(l, mid, r):
    i, j = l, mid+1
    tmp = []
    while i<=mid or j<=r:
        if j>r or (i<=mid and arr[i]<arr[j]):
            tmp.append(arr[i])
            i += 1
        else:
            tmp.append(arr[j])
            global ans
            ans += j-(l+len(tmp)-1)
            j += 1
    arr[l:r+1] = tmp

def m_sort(l, r):
    if l == r:
        return
    mid = (l+r)//2
    m_sort(l, mid)
    m_sort(mid+1, r)
    merge(l, mid, r)

while n := int(input()):
    arr = [int(input()) for i in range(n)]
    ans = 0
    m_sort(0, n-1)
    print(ans)
```

基本信息

#: 43891076  
题目: 02299  
提交人: 23n2300017735(夏天明  
BrightSummer)  
内存: 30420kB  
时间: 4387ms  
语言: Python3  
提交时间: 2024-02-13 00:02:25

## 2. 学习总结和收获

作业题目和最近的每日选做题目之前做过，现在趁机复习一下，并且在群里讨论问题，检查有没有没注意到的盲区。