Assignment #7: April 月考

Updated 1557 GMT+8 Apr 3, 2024

2024 spring, Complied by 刘子暄 环境科学与工程学院

说明：

1）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。

2）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。

3）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：Windows 11

Python编程环境: PyCharm Community Edition 2023.3

1. 题目

27706: 逐词倒放

http://cs101.openjudge.cn/practice/27706/

思路：

代码

```
n = list(input().split())
for i in n[::-1]:
    print(i,end=' ')
```

代码运行截图 （至少包含有"Accepted"）

```
n = int(input())
m = 8
lis = []
if n == 0:
    print(0)
else:
    while n != 0:
        lis.append(n % m)
        n = int((n - n % m) / m)

    a = ''
    while lis:
        a += str(lis.pop())

    print(a)
```

27951: 机器翻译

http://cs101.openjudge.cn/practice/27951/

思路：

代码

```
M, N = map(int,input().split())
sheet = list(input().split())

dic = []
counts = 0

for i in sheet:
    if i in dic:
        continue
    else:
        if len(dic) == M:
            dic.pop(0)
            dic.append(i)
            counts += 1
        else:
            dic.append(i)
            counts += 1

print(counts)
```

代码运行截图 （至少包含有"Accepted"）

#44265109提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
lis = []
n = input().split()

for i in n[::-1]:
    if i not in ['+','-','*','/']:
        lis.append(i)

    if i in ['+','-','*','/']:
        a = float(lis.pop())
        b = float(lis.pop())
        if i == '+':
            lis.append(a + b)
        elif i == '-':
            lis.append(a - b)
        elif i == '*':
            lis.append(a * b)
        elif i == '/':
            lis.append(a / b)

for il in lis:
    print(f"{il:.6f}")
```

基本信息

#:	44265109
题目:	02694
提交人:	刘子暄
内存:	3592kB
时间:	21ms
语言:	Python3
提交时间:	2024-03-17 13:40:48

English 帮助 关于

27932: Less or Equal

http://cs101.openjudge.cn/practice/27932/

思路：

代码

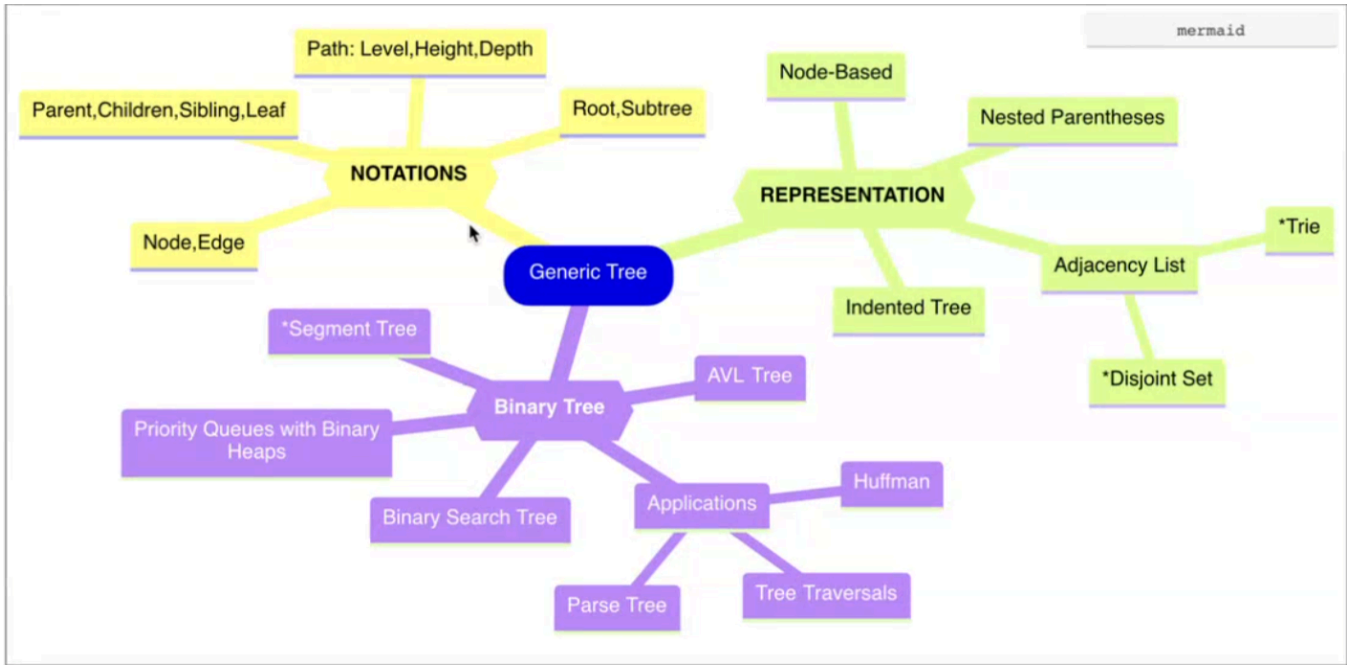
```
n, k = map(int, input().split())

a = list(map(int, input().split()))
a.sort()

if k == 0:
    x = 1 if a[0] > 1 else -1
elif k == n:
    x = a[-1]
else:
    x = a[k-1] if a[k-1] < a[k] else -1

print(x)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



27948: FBI树

http://cs101.openjudge.cn/practice/27948/

思路：

代码

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

def build_FBI(string):
    root = Node()
    if '0' not in string:
        root.value = 'I'
    elif '1' not in string:
        root.value = 'B'
    else:
        root.value = 'F'
    l = len(string) // 2
    if l > 0:
        root.left = build_FBI(string[:l])
        root.right = build_FBI(string[l:])
    return root

def post_traverse(node):
    ans = []
    if node:
        ans.extend(post_traverse(node.left))
        ans.extend(post_traverse(node.right))
        ans.append(node.value)
    return ''.join(ans)

n = int(input())
string = input()
root = build_FBI(string)
print(post_traverse(root))
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

#44306020提交状态

查看 提交 统计 提

状态: Accepted

源代码

```
def infix_to_postfix(expression):
    precedence = {'+':1,'-':1,'*':2,'/':2}
    stack = []
    postfix = []
    number = ''

    for char in expression:
        if char.isnumeric() or char == '.':
            number += char# (重组数字)
        else:
            if number:  # 判断有没有number
                num = float(number)
                postfix.append(int(num) if num.is_integer() else num)
                number = ''
            if char in '+-*/':
                while stack and stack[-1] in '+-*/' and precedence[char]
                    postfix.append(stack.pop())
                stack.append(char)
            elif char == '(':
                stack.append(char)
            elif char == ')':
                while stack and stack[-1] != '(':
                    postfix.append(stack.pop())
                stack.pop()  # 一轮括号使用完毕

    if number:
        num = float(number)
        postfix.append(int(num) if num.is_integer() else num)#有可能还有n

    while stack:
        postfix.append(stack.pop())

    return ' '.join(str(x) for x in postfix)

n = int(input())
for _ in range(n):
    expression = input()
    print(infix_to_postfix(expression))
```

基本信息

#: 44306020
题目: 24591
提交人: 刘子暄
内存: 3700kB
时间: 28ms
语言: Python3
提交时间: 2024-03-19 22:05:17

27925: 小组队列

http://cs101.openjudge.cn/practice/27925/

思路：

注意双端队列的调用格式

代码

```
from collections import deque
t = int(input())
groups = {}
member_to_group = {}



for _ in range(t):
    members = list(map(int, input().split()))
    group_id = members[0]
    groups[group_id] = deque()
    for member in members:
        member_to_group[member] = group_id

queue = deque()
queue_set = set()


while True:
    command = input().split()
    if command[0] == 'STOP':
        break
    elif command[0] == 'ENQUEUE':
        x = int(command[1])
        group = member_to_group.get(x, None)
        if group is None:
            group = x
            groups[group] = deque([x])
            member_to_group[x] = group
        else:
            groups[group].append(x)
        if group not in queue_set:
            queue.append(group)
            queue_set.add(group)
    elif command[0] == 'DEQUEUE':
        if queue:
            group = queue[0]
            x = groups[group].popleft()
            print(x)
```

```
            if not groups[group]:
                queue.popleft()
                queue_set.remove(group)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

CS101 / 题库

题目 排名 状态 提问

#44307176提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```
def is_valid_pop_sequence(origin, output):
    if len(origin) != len(output):
        return False

    stack = []
    bank = list(origin)

    for char in output:
        while (not stack or stack[-1] != char) and bank:
            stack.append(bank.pop(0))

        if not stack or stack[-1] != char:
            return False

        stack.pop()

    return True

origin = input().strip()

while True:
    try:
        output = input().strip()
        if is_valid_pop_sequence(origin, output):
            print('YES')
        else:
            print('NO')
    except EOFError:
        break
```

基本信息

#: 44307176
题目: 22068
提交人: 刘子暄
内存: 3620kB
时间: 24ms
语言: Python3
提交时间: 2024-03-19 23:08:11

©2002-2022 POJ 京ICP备20010980号-1 English 帮助 关于

27928: 遍历树

http://cs101.openjudge.cn/practice/27928/

思路：

代码

```
class Tree:
    def __init__(self, val):
        self.val = val
        self.children = []
        self.parent = None

    def add_child(self, child):
        self.children.append(child)
        child.parent = self

    def traverse(self):
        if self.children == []:
            print(self.val)
        else:
            tmp_nodes = self.children + [self]
            tmp_nodes.sort(key=lambda x: x.val)
            for node in tmp_nodes:
                if node.val != self.val:
                    node.traverse()
                else:
                    print(node.val)

def build_tree(n, nodes):
    for _ in range(n):
        values = list(map(int, input().split()))
        root_val = values[0]
        if root_val not in nodes:
            nodes[root_val] = Tree(root_val)
        t = nodes[root_val]
        for child_val in values[1:]:
            if child_val not in nodes:
                nodes[child_val] = Tree(child_val)
            child = nodes[child_val]
            t.add_child(child)
            child.parent = t

    root = None
    for root_val in nodes:
        if not nodes[root_val].parent:
```

```
            root = nodes[root_val]
            break

    return root

nodes = {}
n = int(input())
root = build_tree(n, nodes)
if root:
    root.traverse()
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

状态: Accepted

源代码

```python
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):
```

基本信息

#:	44307277
题目:	02299
提交人:	刘子暄
内存:	37500kB
时间:	4034ms
语言:	Python3
提交时间:	2024-03-19 23:14:00

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。

没想到前四道题还不是很难，太好了

最后两道没太看懂，马上其他科目期中，考完了会好好补一补的（