Assignment #5: "树"算：概念、表示、解析、遍历

Updated 2124 GMT+8 March 17, 2024

2024 spring, Complied by 苏王捷 工学院

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

（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：Spyder IDE 5.5.0

1. 题目

27638: 求二叉树的高度和叶子数目

http://cs101.openjudge.cn/practice/27638/

思路：每次更新父节点

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Tue Feb 13 16:45:46 2024

@author: Lenovo
"""
```

```python
class Node():
    def __init__(self):
        self.parent=None
        self.lchild=None
        self.rchild=None

n=int(input())
tree=[Node() for i in range(n+1)]
d,num=[0]*(n+1),0
for i in range(n):
    l,r=map(int,input().split())
    if l==-1 and r==-1:
        num+=1
    else:
        d[i]=max(d[l],d[r])+1
    node=i
    while tree[node].parent is not None:
        if d[tree[node].parent]<d[node]+1:
            d[tree[node].parent]=d[node]+1
            node=tree[node].parent
        else:
            break
    tree[i].lchild,tree[i].rchild=l,r
    tree[l].parent=tree[r].parent=i
print(max(d),num)
```

代码运行截图 （至少包含有"Accepted"）

#44179004提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```python
# -*- coding: utf-8 -*-
"""
Created on Tue Feb 13 16:45:46 2024

@author: Lenovo
"""

class Node():
    def __init__(self):
        self.parent=None
        self.lchild=None
        self.rchild=None

n=int(input())
tree=[Node() for i in range(n+1)]
d,num=[0]*(n+1),0
for i in range(n):
    l,r=map(int,input().split())
    if l==-1 and r==-1:
        num+=1
    else:
        d[i]=max(d[l],d[r])+1
    node=i
    while tree[node].parent is not None:
        if d[tree[node].parent]<d[node]+1:
            d[tree[node].parent]=d[node]+1
            node=tree[node].parent
        else:
            break
    tree[i].lchild,tree[i].rchild=l,r
    tree[l].parent=tree[r].parent=i
print(max(d),num)
```

基本信息

#:	44179004
题目:	27638
提交人:	23n2300011075(才疏学浅)
内存:	3652kB
时间:	24ms
语言:	Python3
提交时间:	2024-03-12 09:18:53

24729: 括号嵌套树

http://cs101.openjudge.cn/practice/24729/

思路：用，和（）分辨子树，递归

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Tue Mar 12 19:19:10 2024

@author: Lenovo
"""

class Node:
    def __init__(self,value):
        self.value=value
        self.childs=[]

def build(s):
    if '(' not in s:
        return Node(s)
    root=Node(s[0])
    subtrees=s[2:-1]
    stack=[]
    comma=[-1]
    for i,char in enumerate(subtrees):
        if char=='(':
            stack.append(char)
        elif char==')':
            stack.pop()
        elif char==',' and not stack:
            comma.append(i)
    comma.append(len(subtrees))
    for i in range(len(comma)-1):
        root.childs.append(build(subtrees[comma[i]+1:comma[i+1]]))
    return root

def preorder(root):
    print(root.value,end="")
    for child in root.childs:
        preorder(child)

def postorder(root):
    for child in root.childs:
        postorder(child)
    print(root.value,end="")

tree=build(input())
```

```
preorder(tree)
print()
postorder(tree)
print()
```

代码运行截图 （至少包含有"Accepted"）

#44192591提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 12 19:19:10 2024

@author: Lenovo
"""

class Node:
    def __init__(self,value):
        self.value=value
        self.childs=[]

def build(s):
    if '(' not in s:
        return Node(s)
    root=Node(s[0])
    subtrees=s[2:-1]
    stack=[]
    comma=[-1]
    for i,char in enumerate(subtrees):
        if char=='(':
            stack.append(char)
        elif char==')':
            stack.pop()
```

基本信息

#: 44192591
题目: 24728
提交人: 23n2300011075(才疏学浅)
内存: 3636kB
时间: 21ms
语言: Python3
提交时间: 2024-03-13 10:44:06

02775: 文件结构“图”

http://cs101.openjudge.cn/practice/02775/

思路：递归

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Mon Oct 16 08:54:35 2023

@author: Lenovo
"""

n=1
flag=True
s=""
def printkg(level):
    for _ in range(level):
        print("|     ", end="")
```

```
def pf(level):
    global flag
    global s
    file_set = set()
    while True:
        if s!="":
            str_input = s
            s=""
        else:
            str_input = input().strip()
        if str_input.startswith('f'):
            file_set.add(str_input)
        elif str_input.startswith('d'):
            printkg(level)
            print(str_input)
            pf(level + 1)
        elif str_input == ']':
            for file in sorted(file_set):
                printkg(level - 1)
                print(file)
            return
        elif str_input == '*':
            for file in sorted(file_set):
                print(file)
            s=input()
            if s=="#":
                flag=False
            return

while flag:
    print(f"DATA SET {n}:")
    print("ROOT")
    pf(1)
    n+=1
    print()
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

#44259647提交状态

查看 提交 统计 提问

状态: Accepted

基本信息

#: 44259647
题目: 02775
提交人: 23n2300011075(才疏学浅)
内存: 3612kB
时间: 27ms
语言: Python3
提交时间: 2024-03-17 08:16:15

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Mon Oct 16 08:54:35 2023

@author: Lenovo
"""

n=1
flag=True
s=""
def printkg(level):
    for _ in range(level):
        print("|     ", end="")

def pf(level):
    global flag
    global s
    file_set = set()
    while True:
        if s!="":
            str_input = s
            s=""
        else:
            str_input = input().strip()
        if str_input.startswith('f'):
            file_set.add(str_input)
        elif str_input.startswith('d'):
            printkg(level)
            print(str_input)
            pf(level + 1)
```

25140: 根据后序表达式建立队列表达式

http://cs101.openjudge.cn/practice/25140/

思路：用栈建树

代码

```
#
from collections import deque
class Node:
    def __init__(self,value):
        self.value=value
        self.lchild=None
        self.rchild=None

def build(string):
    stack=[]
    for char in string:
        if "a"<=char<="z":
            stack.append(Node(char))
        else:
            r,l=stack.pop(),stack.pop()
            node=Node(char)
            node.lchild=l
            node.rchild=r
            stack.append(node)
```

```
    return stack.pop()

def order(root):
    ans=[]
    q=deque()
    q.append(root)
    while q:
        node=q.popleft()
        ans.append(node.value)
        if node.lchild is not None:
            q.append(node.lchild)
        if node.rchild is not None:
            q.append(node.rchild)
    return ans[::-1]

n=int(input())
for _ in range(n):
    s=input()
    tree=build(s)
    print("".join(order(tree)))
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

#44193609提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
from collections import deque
class Node:
    def __init__(self,value):
        self.value=value
        self.lchild=None
        self.rchild=None

def build(string):
    stack=[]
    for char in string:
        if "a"<=char<="z":
            stack.append(Node(char))
        else:
            r,l=stack.pop(),stack.pop()
            node=Node(char)
            node.lchild=l
            node.rchild=r
            stack.append(node)
    return stack.pop()

def order(root):
    ans=[]
    q=deque()
    q.append(root)
    while q:
        node=q.popleft()
        ans.append(node.value)
        if node.lchild is not None:
            q.append(node.lchild)
        if node.rchild is not None:
            q.append(node.rchild)
    return ans[::-1]

n=int(input())
for _ in range(n):
    s=input()
    tree=build(s)
    print("".join(order(tree)))
```

基本信息

#: 44193609
题目: 25140
提交人: 23n2300011075(才疏学浅)
内存: 3672kB
时间: 27ms
语言: Python3
提交时间: 2024-03-13 11:48:27

24750: 根据二叉树中后序序列建树

http://cs101.openjudge.cn/practice/24750/

思路：找到root，分出左右子树

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Tue Feb 13 16:25:05 2024

@author: Lenovo
"""

def loge(mid,last,length):
    if length==0:
        return
    if length==1:
        print(mid,end="")
        return
    top=last[-1]
    print(top,end="")
    for i in range(length-1,-1,-1):
        if mid[i]==top:
            break
    loge(mid[:i],last[:i],i)
    loge(mid[i+1:],last[i:-1],length-i-1)

mid=input()
last=input()
length=len(mid)
loge(mid,last,length)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

#43892558提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Tue Feb 13 16:25:05 2024

@author: Lenovo
"""

def loge(mid,last,length):
    if length==0:
        return
    if length==1:
        print(mid,end="")
        return
    top=last[-1]
    print(top,end="")
    for i in range(length-1,-1,-1):
        if mid[i]==top:
            break
    loge(mid[:i],last[:i],i)
    loge(mid[i+1:],last[i:-1],length-i-1)

mid=input()
last=input()
length=len(mid)
loge(mid,last,length)
```

基本信息

#:	43892558
题目:	24750
提交人:	23n2300011075(才疏学浅)
内存:	3620kB
时间:	22ms
语言:	Python3
提交时间:	2024-02-13 16:43:28

22158: 根据二叉树前中序序列建树

http://cs101.openjudge.cn/practice/22158/

思路：找到root，分出左右子树

代码

```
#
class Node:
    def __init__(self):
        self.value=None
        self.lchild=None
        self.rchild=None

def build(preorder,inorder,l):
    if l==0:
        return None
    node=Node()
    root=preorder[0]
    node.value=root
    i=inorder.find(root)
    node.lchild=build(preorder[1:i+1],inorder[:i],i)
    node.rchild=build(preorder[i+1:],inorder[i+1:],l-i-1)
    return node

def postorder(root):
    if root is None:
        return
```

```
    postorder(root.lchild)
    postorder(root.rchild)
    print(root.value,end="")

while True:
    try:
        preorder=input()
        inorder=input()
        tree=build(preorder,inorder,len(inorder))
        postorder(tree)
        print()
    except EOFError:
        break
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

#44192906提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
class Node:
    def __init__(self):
        self.value=None
        self.lchild=None
        self.rchild=None

def build(preorder,inorder,l):
    if l==0:
        return None
    node=Node()
    root=preorder[0]
    node.value=root
    i=inorder.find(root)
    node.lchild=build(preorder[1:i+1],inorder[:i],i)
    node.rchild=build(preorder[i+1:],inorder[i+1:],l-i-1)
    return node

def postorder(root):
    if root is None:
        return
    postorder(root.lchild)
    postorder(root.rchild)
    print(root.value,end="")

while True:
    try:
        preorder=input()
        inorder=input()
        tree=build(preorder,inorder,len(inorder))
        postorder(tree)
        print()
    except EOFError:
        break
```

基本信息

#:	44192906
题目:	22158
提交人:	23n2300011075(才疏学浅)
内存:	3620kB
时间:	21ms
语言:	Python3
提交时间:	2024-03-13 11:05:08

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。

近期题目练习颇少，觉得笔试没底