Assignment #5: "树"算：概念、表示、解析、遍历

Updated 2124 GMT+8 March 17, 2024

2024 spring, Complied by 张羽扬 数学科学学院

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 11 家庭中文版 22H2

Python编程环境：Spyder IDE 5.2.2

C/C++编程环境：无

1. 题目

27638: 求二叉树的高度和叶子数目

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

思路：

代码

```
n=int(input())
a=[]
b=[0]*n
S=set(range(n))
T=set()
```

```
u=0
for i in range(n):
    x=[int(i) for i in input().split()]
    if x[0]==x[1]==-1:
        u=u+1
    a.append(x)
    S.discard(x[0])
    S.discard(x[1])
for i in S:
    b[i]=0
for t in range(1,n+1):
    T.clear()
    for i in S:
        T.add(a[i][0])
        T.add(a[i][1])
    T.discard(-1)
    S.clear()
    S.update(T)
    if len(S)==0:
        break
print(str(t-1)+' '+str(u))
```

代码运行截图

状态：Accepted

基本信息

#: 44378959
题目: 27638
提交人: 2200010796Delphinida(2200010796)
内存: 3628kB
时间: 23ms
语言: Python3
提交时间: 2024-03-24 15:00:17

源代码

```
n=int(input())
a=[]
b=[0]*n
S=set(range(n))
T=set()
u=0
for i in range(n):
    x=[int(i) for i in input().split()]
    if x[0]==x[1]==-1:
        u=u+1
    a.append(x)
    S.discard(x[0])
    S.discard(x[1])
for i in S:
    b[i]=0
for t in range(1,n+1):
    T.clear()
    for i in S:
        T.add(a[i][0])
        T.add(a[i][1])
    T.discard(-1)
    S.clear()
    S.update(T)
    if len(S)==0:
        break
print(str(t-1)+' '+str(u))
```

24729: 括号嵌套树

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

思路：一开始思路想错了于是花了很久时间，但最后写出了很简洁的代码

代码

```
a=str(input())
b=a.replace(')','')
b=b.replace('(','')
b=b.replace(',','')
print(b)
c=a.replace(',','')
i=0
d=''
while len(c)!=1:
    if c[i]=='(' and c[i+1]==')':
        c=c[:i]+c[i+2:]
        i=i-1
    elif c[i]!='(' and c[i]!=')' and c[i+1]!='(':
        d=d+c[i]
        c=c[:i]+c[i+1:]
        i=i-1
    else:
        i=i+1
d=d+c
print(d)
```

代码运行截图

状态: Accepted

源代码

```
a=str(input())
b=a.replace(')','')
b=b.replace('(','')
b=b.replace(',','')
print(b)
c=a.replace(',','')
i=0
d=''
while len(c)!=1:
    if c[i]=='(' and c[i+1]==')':
        c=c[:i]+c[i+2:]
        i=i-1
    elif c[i]!='(' and c[i]!=')' and c[i+1]!='(':
        d=d+c[i]
        c=c[:i]+c[i+1:]
        i=i-1
    else:
        i=i+1
d=d+c
print(d)
```

基本信息

#: 44380612
题目: 24729
提交人: 2200010796Delphinida(2200010796)
内存: 3656kB
时间: 25ms
语言: Python3
提交时间: 2024-03-24 15:51:49

02775: 文件结构“图”

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

思路：

代码

```
#
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

25140: 根据后序表达式建立队列表达式

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

思路：建立解析树 然后用队列层序遍历

代码

```
class TreeNode:
    def __init__(self, x):
        self.val = x
        self.left = None
        self.right = None
n=int(input())
for i in range(n):
    a=input()
    x=[]
    t=len(a)
    for i in range(t):
        if a[i].islower():
            x.append(TreeNode(a[i]))
        else:
            u=x.pop()
            v=x.pop()
            t=TreeNode(a[i])
            t.left=u
            t.right=v
            x.append(t)
    u=x[0]
    c=[u]
    for i in c:
```

```
        if i.right!=None:
            c.append(i.right)
        if i.left!=None:
            c.append(i.left)
    d=[i.val for i in c]
    d.reverse()
    ans=''.join(d)
    print(ans)
```

代码运行截图

状态: Accepted

源代码

```
class TreeNode:
    def __init__(self, x):
        self.val = x
        self.left = None
        self.right = None
n=int(input())
for i in range(n):
    a=input()
    x=[]
    t=len(a)
    for i in range(t):
        if a[i].islower():
            x.append(TreeNode(a[i]))
        else:
            u=x.pop()
            v=x.pop()
            t=TreeNode(a[i])
            t.left=u
            t.right=v
            x.append(t)
    u=x[0]
    c=[u]
    for i in c:
        if i.right!=None:
            c.append(i.right)
        if i.left!=None:
            c.append(i.left)
    d=[i.val for i in c]
    d.reverse()
    ans=''.join(d)
    print(ans)
```

基本信息

#: 44409496
题目: 25140
提交人: 2200010796Delphinida(2200010796)
内存: 3648kB
时间: 29ms
语言: Python3
提交时间: 2024-03-26 17:27:54

24750: 根据二叉树中后序序列建树

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

思路:

代码

```
class TreeNode:
    def __init__(self, x):
        self.val = x
        self.left = None
        self.right = None
def buildTree(inorder, postorder):
    if not inorder or not postorder:
```

```
        return None
    root_val = postorder.pop()
    root = TreeNode(root_val)
    root_index = inorder.index(root_val)
    root.right = buildTree(inorder[root_index + 1:], postorder)
    root.left = buildTree(inorder[:root_index], postorder)
    return root
def f(x):
    u=[]
    if x:
        u.append(x.val)
        u+=f(x.left)
        u+=f(x.right)
    return u
a=list(input())
b=list(input())
x=buildTree(a,b)
y=f(x)
print(''.join(y))
```

代码运行截图

状态：Accepted

源代码

```
class TreeNode:
    def __init__(self, x):
        self.val = x
        self.left = None
        self.right = None
def buildTree(inorder, postorder):
    if not inorder or not postorder:
        return None
    root_val = postorder.pop()
    root = TreeNode(root_val)
    root_index = inorder.index(root_val)
    root.right = buildTree(inorder[root_index + 1:], postorder)
    root.left = buildTree(inorder[:root_index], postorder)
    return root
def f(x):
    u=[]
    if x:
        u.append(x.val)
        u+=f(x.left)
        u+=f(x.right)
    return u
a=list(input())
b=list(input())
x=buildTree(a,b)
y=f(x)
print(''.join(y))
```

基本信息

#: 44382740
题目: 24750
提交人: 2200010796Delphinida(2200010796)
内存: 3668kB
时间: 27ms
语言: Python3
提交时间: 2024-03-24 16:54:33

22158: 根据二叉树前中序序列建树

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

思路：按自己的思路写了一下前面那个题的代码

代码

```
class TreeNode:
    def __init__(self, x):
        self.val = x
        self.left = None
        self.right = None
def f(a,b):
    if a=='' or b=='':
        return
    x=TreeNode(a[0])
    i=b.find(x.val)
    u=b[:i]
    v=b[i+1:]
    for i in range(1,len(a)):
        if u.find(a[i])==-1:
            m=a[1:i]
            n=a[i:]
            break
    else:
        m=a[1:]
        n=''
    x.left=f(m,u)
    x.right=f(n,v)
    return x
def g(x):
    a=[]
    if x==None:
        return a
    a=a+g(x.left)
    a=a+g(x.right)
    a.append(x.val)
    return a
while 1:
    try:
        a=input()
        b=input()
        x=f(a,b)
        print(''.join(g(x)))
    except EOFError:
        break
```

代码运行截图

状态：Accepted

源代码

```
class TreeNode:
    def __init__(self, x):
        self.val = x
        self.left = None
        self.right = None
def f(a,b):
    if a=='' or b=='':
        return
    x=TreeNode(a[0])
    i=b.find(x.val)
    u=b[:i]
    v=b[i+1:]
    for i in range(1,len(a)):
        if u.find(a[i])==-1:
            m=a[1:i]
            n=a[i:]
            break
    else:
        m=a[1:]
        n=''
    x.left=f(m,u)
    x.right=f(n,v)
    return x
def g(x):
    a=[]
    if x==None:
        return a
    a=a+g(x.left)
    a=a+g(x.right)
    a.append(x.val)
    return a
while 1:
    try:
        a=input()
        b=input()
```

基本信息

#: 44383783
题目: 22158
提交人: 2200010796Delphinida(2200010796)
内存: 3568kB
时间: 30ms
语言: Python3
提交时间: 2024-03-24 17:30:43

2. 学习总结和收获

学会了如何建树，从文件结构图的题解中发现了树的节点的值灵活性，不仅可以用整变量/浮点数，还可以用列表来储存多个该储存的内容