

Assignment #5: "树"算：概念、表示、解析、遍历

Updated 2124 GMT+8 March 17, 2024

2024 spring, Compiled by ==陈亚偲 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

Python编程环境：Spyder IDE 5.2.2

1. 题目

27638: 求二叉树的高度和叶子数目

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

思路：

双向关系（定义p来记录父节点，快很多）

递归算高度，比较简单（根本意义上是dfs，但似乎bfs更具美感？）

代码

```
#
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 19 14:13:42 2024

@author: 陈亚偲2300011106
```

```

"""
class N:
    def __init__(self,num):
        self.n=num
        self.r=None
        self.l=None
        self.p=None#用m换t的操作

n=int(input())
w=[N(i) for i in range(n)]#i 是编号，也可以不要

for i in range(n):
    a,b=map(int,input().split())
    if a!=-1:
        w[i].l=w[a]
        w[a].p=w[i]
    if b!=-1:
        w[i].r=w[b]
        w[b].p=w[i]
def end():
    ct=0
    for i in range(n):
        if w[i].r==None and w[i].l==None:
            ct+=1
    return ct
def root():
    for i in range(n):
        if w[i].p==None:
            return i
c=w[root()]
def h(temp):
    if temp==None:
        return 0
    else:
        return max([h(temp.l),h(temp.r)])+1

p=h(c)-1
q=end()
print(f'{p} {q}')

```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 19 14:13:42 2024

@author: 陈亚偲2300011106
"""
class N:
    def __init__(self,num):
        self.n=num
        self.r=None
        self.l=None
        self.p=None#用m换t的操作

n=int(input())
w=[N(i) for i in range(n)]#i 是编号, 也可以不要

for i in range(n):
    a,b=map(int,input().split())
    if a!=-1:
        w[i].l=w[a]
        w[a].p=w[i]
    if b!=-1:
        w[i].r=w[b]
        w[b].p=w[i]
def end():
    ct=0
    for i in range(n):
        if w[i].r==None and w[i].l==None:
            ct+=1
    return ct
def root():
    for i in range(n):
        if w[i].p==None:
            return i
c=w[root()]
def h(temp):
    if temp==None:
        return 0
    else:
        return max([h(temp.l),h(temp.r)])+1

p=h(c)-1
q=end()
print(f' {p} {q}')
```

24729: 括号嵌套树

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

思路:

，用不到，直接建树即可，然后用up和down输出

代码

```
#
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 26 00:01:49 2024

@author: abrac
"""
u=[]
d=[]
class node:
    def __init__(self,name):#针对dir
        self.n=name
        self.c=[]
        self.av=False #是否可以再接受子节点
def up(m):
    u.append(m.n)
    for i in m.c:
        up(i)
    return
def down(m):
    for i in m.c:
        down(i)
    d.append(m.n)
b=list(input())
boza=len(b)
a=[]
for i in range(boza):
    if b[i]=='(':
        a[-1].av=True
    elif b[i]==')':
        for j in range(len(a)-1,-1,-1):
            if a[j].av:
                a[j].av=False
                break
    elif b[i]!='.':
        a.append(node(b[i]))
        for j in range(len(a)-1,-1,-1):
            if a[j].av:
                a[j].c.append(a[-1])
                break
up(a[0])
down(a[0])
print(''.join(u))
print(''.join(d))
```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 26 00:01:49 2024

@author: abrac
"""
u=[]
d=[]
class node:
    def __init__(self,name):#针对dir
        self.n=name
        self.c=[]
        self.av=False #是否可以再接受子节点
    def up(m):
        u.append(m.n)
        for i in m.c:
            up(i)
        return
    def down(m):
        for i in m.c:
            down(i)
        d.append(m.n)
b=list(input())
boza=len(b)
a=[]
for i in range(boza):
    if b[i]=='(':
        a[-1].av=True
    elif b[i]==')':
        for j in range(len(a)-1,-1,-1):
            if a[j].av:
                a[j].av=False
                break
        break
    elif b[i]!='(',')':
        a.append(node(b[i]))
        for j in range(len(a)-1,-1,-1):
            if a[j].av:
                a[j].c.append(a[-1])
                break
up(a[0])
down(a[0])
print(''.join(u))
print(''.join(d))
```

02775: 文件结构“图”

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

思路:

大回溯！通过树的'available' (av) 参数来考察它是否可以接受其他的d和f，root是永远av=True的d，每次都是通过回溯来锁定第一个av=True的d，进而放到它的子节点里面

代码

```
#
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 25 19:35:56 2024

@author: 陈亚偲2300011106
"""
tab='|      '
class node:
    def __init__(self,name):#针对dir
        self.n=name
        self.cd=[]
        self.cf=[]
        self.level=-1
        self.av=True #是否可以再接受子节点
    def show(m):
        print(tab*(m.level)+m.n)
        for i in m.cd:
            show(i)
        for i in sorted(m.cf):
            print(tab*(m.level)+i)
        return
first=True#开始接受数据
temp=[node('d')]
lisa=1
lv=0
while True:
    a=input()
    if a[0]=='#':
        break
    temp.append(node(a))
    if a[0]=='*':
        num=len(temp)
        for i in range(1,num):
            if temp[i].n[0]=='d':
                lv+=1
                temp[i].level=lv
                for j in range(i-1,-1,-1):
                    if temp[j].av:
                        temp[j].cd.append(temp[i])
                        break
            elif temp[i].n[0]=='f':
                temp[i].level=lv
                temp[i].av=False
                for j in range(i-1,-1,-1):
                    if temp[j].av:
                        temp[j].cf.append(temp[i].n)
                        break
            elif temp[i].n[0]=='':
                lv-=1
```

```
        temp[i].av=False
        for j in range(i-1,-1,-1):
            if temp[j].av:
                temp[j].av=False
                break
    if not first:
        print()
    else:
        first=False
    print('DATA SET '+str(lisa)+':')
    lisa+=1
    temp[0].n='ROOT'
    show(temp[0])
    #print(temp[0].cf)
    lv=0
    temp=[node('d')]
```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 25 19:35:56 2024

@author: 陈亚偲2300011106
"""
tab='|'
class node:
    def __init__(self,name):#针对dir
        self.n=name
        self.cd=[]
        self.cf=[]
        self.level=-1
        self.av=True #是否可以再接受子节点
    def show(m):
        print(tab*(m.level)+m.n)
        for i in m.cd:
            show(i)
        for i in sorted(m.cf):
            print(tab*(m.level)+i)
        return
first=True#开始接受数据
temp=[node('d')]
lisa=1
lv=0
while True:
    a=input()
    if a[0]=='#':
        break
    temp.append(node(a))
    if a[0]=='*':
        num=len(temp)
        for i in range(1,num):
            if temp[i].n[0]=='d':
                lv+=1
                temp[i].level=lv
                for j in range(i-1,-1,-1):
                    if temp[j].av:
                        temp[j].cd.append(temp[i])
                        break
            elif temp[i].n[0]=='f':
                temp[i].level=lv
                temp[i].av=False
                for j in range(i-1,-1,-1):
                    if temp[j].av:
```

25140: 根据后序表达式建立队列表达式

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

思路:

代码

```
#
```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

24750: 根据二叉树中后序序列建树

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

思路:

中序被父节点分开, 每半里面离父节点最近的是那一半的root (父节点的子节点)

代码

```
#
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 26 15:34:22 2024

@author: 陈亚德 2300011106
"""

a=input()
b=list(input())#后序
k=len(b)
u=[]
class t:
    def __init__(self,name):
        self.n=name
        self.r=None
        self.l=None
c=[t(i) for i in b]
def d(m,p):
    temp=p.split(m)
    ll=list(temp[0])
    rr=list(temp[1])
    if ll:
        lm=max([b.index(i) for i in ll])
        c[b.index(m)].l=c[lm]
    if rr:
        rm=max([b.index(i) for i in rr])
        c[b.index(m)].r=c[rm]
    return
def bb(m,p):
    d(m,p)
    temp=p.split(m)
```

```
    if c[b.index(m)].r:
        bb(c[b.index(m)].r.n,temp[1])
    if c[b.index(m)].l:
        bb(c[b.index(m)].l.n,temp[0])
    return
def up(m):
    if m!=None:
        u.append(m.n)
        up(m.l)
        up(m.r)
    return
bb(c[-1].n,a)
up(c[-1])
print(''.join(u))
```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 26 15:34:22 2024

@author: 陈亚德 2300011106
"""

a=input()
b=list(input()) #后序
k=len(b)
u=[]
class t:
    def __init__(self,name):
        self.n=name
        self.r=None
        self.l=None
c=[t(i) for i in b]
def d(m,p):
    temp=p.split(m)
    ll=list(temp[0])
    rr=list(temp[1])
    if ll:
        lm=max([b.index(i) for i in ll])
        c[b.index(m)].l=c[lm]
    if rr:
        rm=max([b.index(i) for i in rr])
        c[b.index(m)].r=c[rm]
    return
def bb(m,p):
    d(m,p)
    temp=p.split(m)
    if c[b.index(m)].r:
        bb(c[b.index(m)].r.n,temp[1])
    if c[b.index(m)].l:
        bb(c[b.index(m)].l.n,temp[0])
    return
def up(m):
    if m!=None:
        u.append(m.n)
        up(m.l)
        up(m.r)
    return
bb(c[-1].n,a)
up(c[-1])
print(''.join(u))
```

22158: 根据二叉树前中序序列建树

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

思路:

输入reverse, 输出left和right顺序翻转, 加上try except, 其他与上一题完全一致

代码

```
#
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 26 15:34:22 2024

@author: 陈亚德 2300011106
"""
while True:
    try:
        b=list(input())
        b.reverse()
        a=input()
        #
        k=len(b)
        u=[]
        class t:
            def __init__(self,name):
                self.n=name
                self.r=None
                self.l=None
        c=[t(i) for i in b]
        def d(m,p):
            temp=p.split(m)
            ll=list(temp[0])
            rr=list(temp[1])
            if ll:
                lm=max([b.index(i) for i in ll])
                c[b.index(m)].l=c[lm]
            if rr:
                rm=max([b.index(i) for i in rr])
                c[b.index(m)].r=c[rm]
            return
        def bb(m,p):
            d(m,p)
            temp=p.split(m)
            if c[b.index(m)].r:
                bb(c[b.index(m)].r.n,temp[1])
            if c[b.index(m)].l:
                bb(c[b.index(m)].l.n,temp[0])
            return
        def up(m):
            if m!=None:
                up(m.l)
                up(m.r)
                u.append(m.n)
            return
        bb(c[-1].n,a)
        up(c[-1])
        print(''.join(u))
    except:
        break
```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 26 15:34:22 2024

@author: 陈亚偲 2300011106
"""
while True:
    try:
        b=list(input())
        b.reverse()
        a=input()
        #
        k=len(b)
        u=[]
        class t:
            def __init__(self,name):
                self.n=name
                self.r=None
                self.l=None
        c=[t(i) for i in b]
        def d(m,p):
            temp=p.split(m)
            ll=list(temp[0])
            rr=list(temp[1])
            if ll:
                lm=max([b.index(i) for i in ll])
                c[b.index(m)].l=c[lm]
            if rr:
                rm=max([b.index(i) for i in rr])
                c[b.index(m)].r=c[rm]
            return
        def bb(m,p):
            d(m,p)
            temp=p.split(m)
            if c[b.index(m)].r:
                bb(c[b.index(m)].r.n,temp[1])
            if c[b.index(m)].l:
                bb(c[b.index(m)].l.n,temp[0])
            return
        def up(m):
            if m!=None:
                up(m.l)
                up(m.r)
                u.append(m.n)
            return
    except:
```

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。==

一定一定一定注意函数def和数据接收改函数用的列表时的顺序!!! (其实可以函数扩号里面加上列表, 但如果偷懒就一定注意顺序!!!) 【de了一上午bug】

树的题不用树总会出现一堆奇奇怪怪的wa

以下是02775文件图的WA代码

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 25 14:35:42 2024

@author: 陈亚偲 2300011106
"""
from collections import defaultdict
tab='|      '
lisa=1
level=0
dt=defaultdict(list)
end=False
ans=[]
go=False
anss=[]
while True:
    a=input()
    if a[0]=='#':
        break
    if end:
        lisa+=1
        level=0
        dt=defaultdict(list)
        end=False
        ans=[]
        go=False
        boza=[]
    if a=='*':
        anss.append('DATA SET '+str(lisa)+':')
        anss.append('ROOT')
        end=True
        if dt.keys():
            kk=max(dt.keys())
            for i in range(kk,-1,-1):
                if dt[i]:
                    for j in sorted(dt[i]):
                        ans.append(tab*i+j)
            for i in ans:
                anss.append(i)
            continue
        if a[0]=='f' :
            if not go:
                dt[level].append(a)
            else:
                boza.append(tab*level+a)
        elif a[0]=='d':
            level+=1
```

```

        go=True
        ans.append(tab*level+a)
        boza=[]
    elif a[0]==']':
        level-=1
        go=False
        for i in sorted(boza):
            ans.append(i)
        boza=[]

for i in anss:
    if 'DATA SET' in i and i!='DATA SET 1:':
        print()
    print(i)

```

后来用树又做了一遍，终于AC了，树非常直观

最后一题用左右翻转的方法直接沿用上一题的代码，说明前序和后续.reverse()的区别就在于左右反向