

Assignment #5: "树"算：概念、表示、解析、遍历

Updated 2124 GMT+8 March 17, 2024

2024 spring, Compiled by ==狄晨阳 生命科学学院==

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

==（请改为同学的操作系统、编程环境等）==

操作系统：Windows11

Python编程环境：Spyder IDE 5.4.3

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：先读取数据建树，然后假设每个节点为根进行遍历找出最长路径为树高，同时如果此时的“根节点”没有子节点则叶子结点数加一

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Fri Mar 22 13:31:30 2024
```

```

@author: 20311
"""

n=int(input())
tree={}
for i in range(n):
    tree[i]=list(map(int,input().split()))

h=0
l=0
def go(i,p):
    global h

    h=max(h,p)
    a,b=tree[i]
    if a!=-1:
        go(a,p+1)
    if b!=-1:
        go(b,p+1)

for i in range(n):
    if tree[i]==[-1,-1]:
        l+=1
    go(i,0)
print(h,l)

```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```

# -*- coding: utf-8 -*-
"""
Created on Fri Mar 22 13:31:30 2024

@author: 20311
"""

n=int(input())
tree={}
for i in range(n):
    tree[i]=list(map(int,input().split()))

h=0
l=0
def go(i,p):
    global h

    h=max(h,p)
    a,b=tree[i]
    if a!=-1:
        go(a,p+1)
    if b!=-1:
        go(b,p+1)

for i in range(n):
    if tree[i]==[-1,-1]:
        l+=1
    go(i,0)
print(h,l)

```

基本信息

#: 44336320
 题目: 27638
 提交人: 23n2300012138(yukino)
 内存: 3660kB
 时间: 23ms
 语言: Python3
 提交时间: 2024-03-22 13:45:42

24729: 括号嵌套树

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

思路：显然，前序遍历只需去除原字符串中的所有字母以外的字符，而对后序遍历的处理采用了栈的方式，如果遇到'，'代表一刻子树到了末端，则弹出到'（为止，但不删除'（，而遇到'）'代表同一分支下的所有子树已遍历完毕，就弹出到'（为止并删去'（即可

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Fri Mar 22 13:46:43 2024

@author: 20311
"""

s=input()
ans1=''
for w in s:
    if w in 'QWERTYUIOPASDFGHJKLZXCVBNM':
        ans1+=w
print(ans1)

stack=[]
ans2=''
for w in s:
    if w!=')' and w!=',':
        stack.append(w)
    elif w==',':
        while stack and stack[-1]!='(':
            ans2+=stack.pop()
    else:
        while stack and stack[-1]!='(':
            ans2+=stack.pop()
        stack.pop()

while stack:
    ans2+=stack.pop()
print(ans2)
```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Fri Mar 22 13:46:43 2024

@author: 20311
"""

s=input()
ans1=''
for w in s:
    if w in 'QWERTYUIOPASDFGHJKLZXCVBNM':
        ans1+=w
print(ans1)

stack=[]
ans2=''
for w in s:
    if w!='(' and w!=',':
        stack.append(w)
    elif w==',':
        while stack and stack[-1]!='(':
            ans2+=stack.pop()
    else:
        while stack and stack[-1]!='(':
            ans2+=stack.pop()
        stack.pop()

while stack:
    ans2+=stack.pop()
print(ans2)
```

基本信息

#: 44336567
题目: 24729
提交人: 23n2300012138(yukino)
内存: 3648kB
时间: 23ms
语言: Python3
提交时间: 2024-03-22 14:02:53

02775: 文件结构“图”

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

思路：先读取数据，运用了建树的方式，但是存在重名的情况就只好写了嵌套列表来处理，导致代码比较长，然后遍历树，用1个变量记录深度以便输出时在前面加上“| ”

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Fri Mar 22 14:12:53 2024

@author: 20311
"""

def operate(t,d,m):
    global tree
    a=tree[t][m]
    if a:
        for s in a:
            if s[0]=='f':
                ans.append([s,d])
            elif s[0]=='d':
                for b in ans:
                    if b[0]==s:
                        m+=1
                ans.append([s,d+1])
                operate(s,d+1,m)
```

```

def calc(x):
    if x[0]=='d':
        return '0'
    elif x[0]=='f':
        return x

c=0
tree={'ROOT': [[]]}
theme=['ROOT']
ans=[['ROOT',0]]
while True:
    x=input()
    r=theme[-1]
    if x=='*':
        c+=1
        for t in tree:
            for l in tree[t]:
                l.sort(key=lambda x:calc(x))
        operate('ROOT',0,0)

        if c!=1:
            print()
        print('DATA SET {}'.format(c))
        for a in ans:
            p=''
            for _ in range(a[1]):
                p+='|'
            p+=a[0]
            print(p)

        tree={'ROOT': [[]]}
        theme=['ROOT']
        ans=[['ROOT',0]]

    if x=='#':
        break

    if x[0]=='f':
        tree[r][len(tree[r])-1].append(x)
    elif x[0]=='d':
        theme.append(x)
        tree[r][len(tree[r])-1].append(x)
        if x not in tree:
            tree[x]=[[]]
        else:
            tree[x].append([])
    elif x==']':
        theme.pop()

```

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 Fri Mar 22 14:12:53 2024

@author: 20311
"""
def operate(t,d,m):
    global tree
    a=tree[t][m]
    if a:
        for s in a:
            if s[0]==f'f':
                ans.append([s,d])
            elif s[0]==f'd':
                for b in ans:
                    if b[0]==s:
                        m+=1
                ans.append([s,d+1])
                operate(s,d+1,m)

def calc(x):
    if x[0]==f'd':
        return f'0'
    elif x[0]==f'f':
        return x

c=0
tree={f'ROOT': [[]]}
theme=[f'ROOT']
ans=[(f'ROOT',0)]
while True:
    x=input()
    r=theme[-1]
    if x==f'x':
        c+=1
        for t in tree:
            for l in tree[t]:
                l.sort(key=lambda x:calc(x))
            operate(f'ROOT',0,0)

        if c!=1:
            print()
            print(f'DATA SET {c}:'.format(c))
            for a in ans:
                p=f''
                for _ in range(a[1]):
                    p+=f'|'
                p+=a[0]
                print(p)

            tree={f'ROOT': [[]]}
            theme=[f'ROOT']
            ans=[(f'ROOT',0)]

        if x==f'#':
            break

    if x[0]==f'f':
        tree[r][len(tree[r])-1].append(x)
    elif x[0]==f'd':
        theme.append(x)
        tree[r][len(tree[r])-1].append(x)
        if x not in tree:
            tree[x]=[]
        else:
            tree[x].append([])
    elif x==f']':
        theme.pop()
```

基本信息

#: 44345466
题目: 02775
提交人: 23n2300012138(yukino)
内存: 3724kB
时间: 25ms
语言: Python3
提交时间: 2024-03-22 21:12:56

25140: 根据后序表达式建立队列表达式

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

思路：本想继续使用字典建树但是数据多了以后直接TLE了于是就使用了类来写

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Sat Mar 23 22:48:57 2024

@author: 20311
"""

class Tree:
    def __init__(self,value):
        self.value=value
        self.l=None
        self.r=None

def build(s):
    stack=[]
    for char in s:
        node=Tree(char)
        if char.isupper():
            node.r=stack.pop()
            node.l=stack.pop()
        stack.append(node)
    return stack[0]

def traverse(x):
    l=[x]
    ans=''
    while l:
        node=l.pop(0)
        ans+=(node.value)
        if node.l:
            l.append(node.l)
        if node.r:
            l.append(node.r)
    return ans[::-1]

n=int(input())
for _ in range(n):
    i=build(input().strip())
    print(traverse(i))
```

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 Sat Mar 23 22:48:57 2024

@author: 20311
"""

class Tree:
    def __init__(self,value):
        self.value=value
        self.l=None
        self.r=None

    def build(s):
        stack=[]
        for char in s:
            node=Tree(char)
            if char.isupper():
                node.r=stack.pop()
                node.l=stack.pop()
                stack.append(node)
        return stack[0]

    def traverse(x):
        l=[x]
        ans=''
        while l:
            node=l.pop(0)
            ans+=(node.value)
            if node.l:
                l.append(node.l)
            if node.r:
                l.append(node.r)
        return ans[::-1]

n=int(input())
for _ in range(n):
    i=build(input().strip())
    print(traverse(i))
```

基本信息

#: 44370757
题目: 25140
提交人: 23n2300012138(yukino)
内存: 4012kB
时间: 31ms
语言: Python3
提交时间: 2024-03-23 23:01:44

24750: 根据二叉树中后序序列建树

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

思路: 开始想出了利用后序表达式将中序表达式分为左右子树的递归方法, 但是不知道如何分割后序表达式, 后来意识到左子树的长度就是后续的前部分, 就利用递归建了树

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Sat Mar 23 23:08:51 2024

@author: 20311
"""

def divide(z,h):
    if not z or not h:
        return []

    rt=h[-1]
```



```

d=z.index(rt)
lz=z[:d]
rz=z[d+1:]

lh=h[:len(lz)]
rh=h[len(lz):-1]

tree=[rt]
tree.extend(divide(lz, lh))
tree.extend(divide(rz, rh))

return tree

zx=list(input().strip())
hx=list(input().strip())
tree=divide(zx,hx)
print(''.join(tree))

```

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 Sat Mar 23 23:08:51 2024

@author: 20311
"""

def divide(z,h):
    if not z or not h:
        return []

    rt=h[-1]
    d=z.index(rt)
    lz=z[:d]
    rz=z[d+1:]

    lh=h[:len(lz)]
    rh=h[len(lz):-1]

    tree=[rt]
    tree.extend(divide(lz, lh))
    tree.extend(divide(rz, rh))

    return tree

zx=list(input().strip())
hx=list(input().strip())
tree=divide(zx,hx)
print(''.join(tree))

```

基本信息

#: 44371269
 题目: 24750
 提交人: 23n2300012138(yukino)
 内存: 3628kB
 时间: 24ms
 语言: Python3
 提交时间: 2024-03-24 00:04:28

22158: 根据二叉树前中序序列建树

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

思路: 基本沿用上一题的思路, 改变一下读取顺序和写入顺序即可

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Sun Mar 24 00:12:19 2024

@author: 20311
"""

def divide(qx,zx):
    if not qx or not zx:
        return []

    root=qx[0]
    d=zx.index(root)

    lz=zx[:d]
    rz=zx[d+1:]

    lq=qx[1:len(lz)+1]
    rq=qx[len(lz)+1:]

    hx=[]
    hx.extend(divide(lq,lz))
    hx.extend(divide(rq,rz))
    hx.append(root)

    return hx

while True:
    try:
        qx=list(input().strip())
        zx=list(input().strip())
        hx=divide(qx,zx)
        print(''.join(hx))

    except EOFError:
        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 Sun Mar 24 00:12:19 2024

@author: 20311
"""

def divide(qx,zx):
    if not qx or not zx:
        return []

    root=qx[0]
    d=zx.index(root)

    lz=zx[:d]
    rz=zx[d+1:]

    lq=qx[1:len(lz)+1]
    rq=qx[len(lz)+1:]

    hx=[]
    hx.extend(divide(lq,lz))
    hx.extend(divide(rq,rz))
    hx.append(root)

    return hx

while True:
    try:
        qx=list(input().strip())
        zx=list(input().strip())
        hx=divide(qx,zx)
        print(' '.join(hx))

    except EOFError:
        break
```

基本信息

#: 44371397
题目: 22158
提交人: 23n2300012138(yukino)
内存: 3656kB
时间: 23ms
语言: Python3
提交时间: 2024-03-24 00:23:37

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。==

有

本次作业中对树有了更深的认知，掌握了多种建树的方式，对前中后序遍历也有了更深的理解并了解了如何通过代码来实现