

Assignment #5: "树"算：概念、表示、解析、遍历

Updated 2124 GMT+8 March 17, 2024

2024 spring, Compiled by 郑铭毅 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

Windows 11

PyCharm

操作系统： macOS Ventura 13.4.1 (c)

Python编程环境： Spyder IDE 5.2.2, PyCharm 2023.1.4 (Professional Edition)

C/C++编程环境： Mac terminal vi (version 9.0.1424), g++/gcc (Apple clang version 14.0.3, clang-1403.0.22.14.1)

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

```

ans=0
n=int(input())
left_son,right_son=[],[]
t=0
for i in range(n):
    a,b=map(int,input().split())
    left_son.append(a)
    right_son.append(b)
    if a==-1 and b==-1:
        t+=1
for i in range(n):
    if i not in left_son and i not in right_son:
        s=i
def f(k,d):
    global ans
    if left_son[k]!=-1:
        f(left_son[k],d+1)
    if right_son[k]!=-1:
        f(right_son[k],d+1)
    ans=max(ans,d)
f(s,0)
if n!=1:
    print(ans,t)
if n==1:
    print(0,1)

```

代码运行截图

OpenJudge
题目ID, 标题, 描述
2300010872
信箱
账号

CS101 / 题库
题目
排名
状态
提问

#44377882提交状态
查看
提交
统计
提问

状态: Accepted

源代码

```

ans=0
n=int(input())
left_son,right_son=[],[]
t=0
for i in range(n):
    a,b=map(int,input().split())
    left_son.append(a)
    right_son.append(b)
    if a==-1 and b==-1:
        t+=1
for i in range(n):
    if i not in left_son and i not in right_son:
        s=i
def f(k,d):
    global ans
    if left_son[k]!=-1:
        f(left_son[k],d+1)
    if right_son[k]!=-1:
        f(right_son[k],d+1)
    ans=max(ans,d)
f(s,0)
if n!=1:
    print(ans,t)
if n==1:
    print(0,1)

```

基本信息
: 44377882
题目: 27638
提交人: 2300010872
内存: 3908kB
时间: 25ms
语言: Python3
提交时间: 2024-03-24 14:13:50

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

24729: 括号嵌套树

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

思路:

代码

```
t=input()
stack=[]
s=''
for i in range(len(t)):
    if t[i]!='(' and t[i]!=')' and t[i]!=',:':
        s+=t[i]
print(s)
def parse_tree(t):
    stack=[]
    node=None
    for i in t:
        if i.isalpha():
            node={'father':i,'children':[]}
            if stack:
                stack[-1]['children'].append(node)
        elif i=='(':
            if node:
                stack.append(node)
                node=None
        elif i==')':
            if stack:
                node=stack.pop()
    return node
def f(node):
    l=[]
    for c in node['children']:
        l.append(f(c))
    l.append(node['father'])
    return ''.join(l)
print(f(parse_tree(t)))
```

OpenJudge

题目ID, 标题, 描述

2300010872 信箱 账号

 CS101 / 题库

[题目](#) [排名](#) [状态](#) [提问](#)

#44379309提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
t=input()
stack=[]
s=''
for i in range(len(t)):
    if t[i]!='(' and t[i]!=')' and t[i]!='.':
        s+=t[i]
print(s)
def parse_tree(t):
    stack=[]
    node=None
    for i in t:
        if i.isalpha():
            node={'father':i,'children':[]}
            if stack:
                stack[-1]['children'].append(node)
        elif i=='(':
            if node:
                stack.append(node)
                node=None
            elif i==')':
                if stack:
                    node=stack.pop()
        return node
    def f(node):
        l=[]
        for c in node['children']:
            l.append(f(c))
        l.append(node['father'])
        return ''.join(l)
    print(f(parse_tree(t)))
```

基本信息

#: 44379309
题目: 24729
提交人: 2300010872
内存: 3688kB
时间: 24ms
语言: Python3
提交时间: 2024-03-24 15:14:42

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

02775: 文件结构“图”

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

思路:

代码

```

from sys import exit

class d:
    def __init__(self, name):
        self.name = name
        self.files = []
        self.dirs = []

    def g(self):
        l = [self.name]
        for i in self.dirs:
            s = i.g()
            l.extend(' | ' + t for t in s)
        for i in sorted(self.files):
            l.append(i)
        return l

n = 0
while True:
    n += 1
    stack = [d('ROOT')]
    while (s := input()) != '*':
        if s == '#':
            exit()
        elif s[0] == 'f':
            stack[-1].files.append(s)
        elif s[0] == 'd':
            stack.append(d(s))
            stack[-2].dirs.append(stack[-1])
        else:
            stack.pop()
    print(f'DATA SET {n}:')
    print(*stack[0].g(), sep='\n')
    print()

```

代码运行截图

OpenJudge 题目ID, 标题, 描述 2300010872 依赖 账号

CS101 / 题库

题目 排名 状态 提问

#44394678 提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

from sys import exit

class d:
    def __init__(self, name):
        self.name = name
        self.files = []
        self.dirs = []

    def g(self):
        l = [self.name]
        for i in self.dirs:
            s = i.g()
            l.extend(' | ' + t for t in s)
        for i in sorted(self.files):
            l.append(i)
        return l

n = 0
while True:
    n += 1
    stack = [d('ROOT')]
    while (s := input()) != '*':
        if s == '#':
            exit()
        elif s[0] == 'f':
            stack[-1].files.append(s)
        elif s[0] == 'd':
            stack.append(d(s))
            stack[-2].dirs.append(stack[-1])
        else:
            stack.pop()
    print(f'DATA SET {n}:')
    print(*stack[0].g(), sep='\n')
    print()

```

基本信息

#: 44394678
 题目: 02775
 提交人: 2300010872
 内存: 3652kB
 时间: 26ms
 语言: Python3
 提交时间: 2024-03-25 13:51:47


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

25140: 根据后序表达式建立队列表达式

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

思路:

代码运行截图

 **CS101 / 题库**

[题目](#) [排名](#) [状态](#) [提问](#)

#44398933提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
s1='abcdefghijklmnopqrstuvwxyz'
class treenode:
    def __init__(self,name):
        self.name=name
        self.leftson=None
        self.rightson=None

def parse_tree(s):
    stack=[]
    for i in s:
        node=treenode(i)
        if i not in s1:
            node.rightson=stack.pop()
            node.leftson=stack.pop()
            stack.append(node)
        node=stack.pop()
    return node
def output(root):
    ans=[]
    stack=[root]
    while stack:
        node=stack.pop(0)
        ans.append(node.name)
        if node.leftson:
            stack.append(node.leftson)
        if node.rightson:
            stack.append(node.rightson)
    return ans
n=int(input())
for _ in range(n):
    s=input()
    root=parse_tree(s)
    l=output(root)
    a=''
    for i in range(len(l)):
        a+=l[len(l)-1-i]
    print(a)
```

基本信息

#: 44398933

题目: 25140

提交人: 2300010872

内存: 3688kB

时间: 28ms

语言: Python3

提交时间: 2024-03-25 19:13:58

```

s1='abcdefghijklmnopqrstuvwxyz'

class treenode:
    def __init__(self,name):
        self.name=name
        self.leftson=None
        self.rightson=None

def parse_tree(s):
    stack=[]
    for i in s:
        node=treenode(i)
        if i not in s1:
            node.rightson=stack.pop()
            node.leftson=stack.pop()
            stack.append(node)
    node=stack.pop()
    return node

def output(root):
    ans=[]
    stack=[root]
    while stack:
        node=stack.pop(0)
        ans.append(node.name)
        if node.leftson:
            stack.append(node.leftson)
        if node.rightson:
            stack.append(node.rightson)
    return ans

n=int(input())
for _ in range(n):
    s=input()
    root=parse_tree(s)
    l=output(root)
    a=''
    for i in range(len(l)):
        a+=l[len(l)-1-i]
    print(a)

```

24750: 根据二叉树中后序序列建树

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

没过，但是做了很久，oj上测试数据也通过了，由于作业ddl快到了就先提交，会参考答案修改思路：

代码


```

class Treenode:
    def __init__(self,name):
        self.name=name
        self.leftson=None
        self.rightson=None
    def output(self):
        s=''
        s+=self.name
        if self.leftson:
            s+=self.leftson.output()
        if self.rightson:
            s+=self.rightson.output()
        return s

def parse_tree(s1,s2):
    stack=[]
    root=[]
    for char in s1:
        node=Treenode(char)
        stack.append(node)
        if len(stack)==2:
            if root and s2.index(stack[0].name)+1==s2.index(root[-1].name):
                root.append(stack[0])
                while len(root) > 1:
                    son = root.pop()
                    root[-1].rightson = son
                stack[0] = root[0]
                stack.append(node)
                node.leftson = root.pop()
            else:
                node.leftson=stack[0]
        if len(stack)==3:
            if s2.index(stack[-2].name)==s2.index(node.name)+1:
                stack[-2].rightson=node
                stack.pop(0)
                stack.pop()
            elif s2.index(stack[-2].name)>s2.index(node.name)+1:
                root.append(stack[-2])
                stack.pop(0)
                stack.pop(0)
            else:
                if not root or s2.index(root[0].name)>s2.index(node.name):
                    node.leftson=stack[0]
                    stack.pop(0)
                else:
                    root.append(stack[1])
                    while len(root)>1:
                        son=root.pop()
                        root[-1].rightson=son
                    stack[0]=root[0]
                    stack[1]=node
                    node.leftson=root.pop()
        root.append(stack[-1])
        while len(root) > 1:
            son = root.pop()
            root[-1].rightson = son
    return root[0]

a=input()
b=input()
s1,s2=[],[]
for i in range(len(a)):
    s1.append(a[i])
    s2.append(b[i])
print(parse_tree(s1,s2).output())

```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

22158: 根据二叉树前中序序列建树

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

思路:

代码

```
1  #  
2
```

代码运行截图 (AC代码截图，至少包含有"Accepted")

2. 学习总结和收获

感觉本周作业难度提升了很多，对于不是很熟悉树的我来说挺痛苦的。但是也是非常有收获的一周，从最开始的看题解都不太理解，到后面能够自己琢磨着把题目做出来，有了挺大的进步，对树这一结构也有了更深的理解。