

Assignment #5: "树"算：概念、表示、解析、遍历

Updated 2124 GMT+8 March 17, 2024

2024 spring, Compiled by 数学科学学院 王镜廷 2300010724

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 专业版

Python编程环境：VSCode 1.86.2, with extension Python and python version 3.12.2

1. 题目

27638: 求二叉树的高度和叶子数目

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

用时：约10分钟

思路：

开两个list分别记录每个节点的子树高度和子树叶子数目，每个点处的取值由其左右子节点的值确定。因为题目中没给树根，所以最终取所有高度中最大者和所有叶子数目中最大者。


```

class node :
    def __init__(self, left, right) :
        self.left = left
        self.right = right

h = []
depth = []
size = []
resd = 0
ress = 0

def getdepth(s) :
    global h
    global depth
    if s == -1 :
        return -1
    if depth[s] != 0 :
        return depth[s]
    depth[s] = max(getdepth(h[s].left), getdepth(h[s].right)) + 1
    return depth[s]

def getsizes(s) :
    global h
    global size
    if s == -1 :
        return 0
    if h[s].left == -1 and h[s].right == -1 :
        size[s] = 1
        return 1
    if size[s] != 0 :
        return size[s]
    size[s] = getsizes(h[s].left) + getsizes(h[s].right)
    return size[s]

n = int(input())
depth = [0] * n
size = [0] * n
for i in range(n) :
    u, v = map(int, input().split())
    h.append(node(u, v))
for i in range(n) :
    resd = max(resd, getdepth(i))
for i in range(n) :
    ress = max(ress, getsizes(i))
print(resd, end = " ")

```

```
print(ress)
```

代码运行截图（至少包含有"Accepted"）

#44179351提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
class node :
    def __init__(self, left, right) :
        self.left = left
        self.right = right

h = []
depth = []
size = []
resd = 0
ress = 0
def getdepth(s) :
    global h
    global depth
    if s == -1 :
        return -1
    if depth[s] != 0 :
        return depth[s]
    depth[s] = max(getdepth(h[s].left), getdepth(h[s].right)) + 1
```

基本信息

#: 44179351
题目: 27638
提交人: 23n2300010724
内存: 3720kB
时间: 21ms
语言: Python3
提交时间: 2024-03-12 09:48:18

24729: 括号嵌套树

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

用时: 约20到30分钟

思路:

在输入串中从左向右处理，记录两个栈分别存储已经完成的节点（还有左括号）和还没有完成的节点处的取值。

（本题在每日选做里的题号是24728）

代码

```

class node :
    def __init__(self, sons, val) :
        self.sons = sons
        self.val = val
    def ToPostString(self) :
        if self.sons == [] :
            return self.val
        return "".join(son.ToPostString() for son in self.sons) + self.val
    def __str__(self) :
        if self.sons == [] :
            return self.val
        return self.val + "".join(str(son) for son in self.sons)

```

```

class stack :
    def __init__(self, item) :
        self.item = item
    def top(self) :
        if self.item == [] :
            return None
        else :
            return self.item[len(self.item) - 1]
    def pop(self) :
        if self.item == [] :
            return
        else :
            self.item.pop()
    def getTopAndPop(self) :
        if self.item == [] :
            return None
        else :
            x = self.top()
            self.pop()
            return x
    def isempty(self) :
        if self.item == [] :
            return True
        else :
            return False
    def push(self, elem) :
        self.item.append(elem)
    def size(self) :
        return len(self.item)
    def __str__(self) :
        return str(list(str(item) for item in self.item ))

```

```

def build(s) :

```

```

l = 0
stackval = stack([])
stacknode = stack([])
for l in range(len(s)) :
    if s[l] == "(" :
        stacknode.push("(")
    elif s[l] == ")" or l == len(s):
        tmp = []
        while(stacknode.top() != "(") :
            tmp.append(stacknode.getTopAndPop())
        stacknode.pop()
        tmp.reverse()
        stacknode.push(node(tmp, stackval.getTopAndPop()))
    elif s[l] == "," :
        continue
    elif l < len(s) - 1 and s[l + 1] == "(" :
        stackval.push(s[l])
    else :
        stacknode.push(node([], s[l]))
    #print(l, stackval, stacknode, sep = " ")
return stacknode.top()

s = input()
v = build(s)
print(v)
print(v.ToPostString())

```

代码运行截图（至少包含有"Accepted"）

#44282163提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

class node :
    def __init__(self, sons, val) :
        self.sons = sons
        self.val = val
    def ToPostString(self) :
        if self.sons == [] :
            return self.val
        return "".join(son.ToPostString() for son in self.sons) + self.val
    def __str__(self) :
        if self.sons == [] :
            return self.val
        return self.val + "".join(str(son) for son in self.sons)

class stack :
    def __init__(self, item) :
        self.item = item
    def top(self) :
        if self.item == [] :

```

基本信息

#: 44282163
 题目: 24729
 提交人: 23n2300010724
 内存: 3760kB
 时间: 24ms
 语言: Python3
 提交时间: 2024-03-18 09:00:50

02775: 文件结构“图”

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

用时：约20分钟

思路：

先按条件建树，之后将树转化为字符串

代码

```
h = []
```

```
class stack :
    def __init__(self, item) :
        self.item = item
    def top(self) :
        if self.item == [] :
            return None
        else :
            return self.item[len(self.item) - 1]
    def pop(self) :
        if self.item == [] :
            return
        else :
            self.item.pop()
    def getTopAndPop(self) :
        if self.item == [] :
            return None
        else :
            x = self.top()
            self.pop()
            return x
    def isempty(self) :
        if self.item == [] :
            return True
        else :
            return False
    def push(self, elem) :
        self.item.append(elem)
    def size(self) :
        return len(self.item)

class dir :
    def __init__(self, files, dirs, val) :
        self.files = files
        self.dirs = dirs
        self.val = val
    def addfile(self, val) :
        self.files.append(val)
        self.files = sorted(self.files)
    def adddir(self, val) :
        self.dirs.append(val)
    def ToStr(self) :
        res = [self.val]
        for dir in self.dirs :
            res += ["|      " + this for this in dir.ToStr()]
```



```

        for file in self.files :
            res.append(file)
        return res

def work() :
    global h
    stackDir = stack([dir([], [], "ROOT")])
    for item in h :
        if item[0] == "f" :
            u = stackDir.getTopAndPop()
            u.addfile(item)
            stackDir.push(u)
        if item[0] == "d" :
            stackDir.push(dir([], [], item))
        if item[0] == "]" :
            u = stackDir.getTopAndPop()
            v = stackDir.getTopAndPop()
            v.adddir(u)
            stackDir.push(v)
    for i in stackDir.top().ToStr() :
        print(i)

```

```

Case = 1
while True :
    s = input()
    if s == "*" :
        print(f"DATA SET {Case}:")
        work()
        Case += 1
        print("")
        h = []
    if s == "#" :
        break
    h.append(s)

```

代码运行截图（AC代码截图，至少包含有"Accepted"）

状态: Accepted

源代码

```
h = []

class stack :
    def __init__(self, item) :
        self.item = item
    def top(self) :
        if self.item == [] :
            return None
        else :
            return self.item[len(self.item) - 1]
    def pop(self) :
        if self.item == [] :
            return
        else :
            self.item.pop()
    def getTopAndPop(self) :
        if self.item == [] :
            return None
```

基本信息

#: 44260856

题目: 02775

提交人: 23n2300010724

内存: 3672kB

时间: 27ms

语言: Python3

提交时间: 2024-03-17 10:05:31

25140: 根据后序表达式建立队列表达式

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

用时: 约15分钟

思路:

先从后序表达式建树, 之后输出队列表达式 (即树的层序遍历)

代码

```

class node :
    def __init__(self, left, right, val) :
        self.left = left
        self.right = right
        self.val = val

def ToQueueExpr(s) :
    q = [s]
    res = []
    while q != [] :
        u = q[0]
        del(q[0])
        res.append(u.val)
        if u.left != None :
            q.append(u.left)
        if u.right != None :
            q.append(u.right)
    res.reverse()
    return "".join(item for item in res)

def PostToNode(s) :
    qnode = []
    upper = "QWERTYUIOPASDFGHJKLZXCVBNM"
    for l in range(len(s)) :
        if s[l] in upper :
            x = qnode[len(qnode) - 1]
            qnode.pop()
            y = qnode[len(qnode) - 1]
            qnode.pop()
            qnode.append(node(y, x, s[l]))
        else :
            qnode.append(node(None, None, s[l]))
    return qnode[0]

n = int(input())
for i in range(n) :
    s = input()
    print(ToQueueExpr(PostToNode(s)))

```

代码运行截图（AC代码截图，至少包含有"Accepted"）

状态: **Accepted**

源代码

```
class node :
    def __init__(self, left, right, val) :
        self.left = left
        self.right = right
        self.val = val

def ToQueueExpr(s) :
    q = [s]
    res = []
    while q != [] :
        u = q[0]
        del(q[0])
        res.append(u.val)
        if u.left != None :
            q.append(u.left)
        if u.right != None :
            q.append(u.right)
    res.reverse()
```

基本信息

#: 44198289
题目: 25140
提交人: 23n2300010724
内存: 3648kB
时间: 27ms
语言: Python3
提交时间: 2024-03-13 16:33:31

24750: 根据二叉树中后序序列建树

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

用时: 约5分钟 (因为复用了以前的一些代码, 建树部分相当于每日选做里面的25145猜二叉树)

思路:

首先根据中后序序列建树 (即每次找到当前串对应的根节点, 之后递归处理左右子树), 之后按前序序列输出

代码

```

class node :
    def __init__ (self, left, right, val) :
        self.left = left
        self.right = right
        self.val = val
    def __str__(self):
        s1 = ""
        if self.left != None :
            s1 = str(self.left)
        s2 = ""
        if self.right != None :
            s2 = str(self.right)
        return self.val + s1 + s2

def getIndex(c, s) :
    for i in range(len(s)) :
        if c == s[i] :
            return i
    return -1

def BuildTree_post_in(sp, si) :
    if sp == "" :
        return None
    l = getIndex(sp[len(sp) - 1], si)
    # print(sp, si, l, sep = " ")
    return node(BuildTree_post_in(sp[0 : l], si[0 : l]), BuildTree_post_in(sp[l : len(sp) - 1],

s1 = input()
s2 = input()
print(BuildTree_post_in(s2, s1))

```

代码运行截图（AC代码截图，至少包含有"Accepted"）

状态: **Accepted**

源代码

```
class node :
    def __init__ (self, left, right, val) :
        self.left = left
        self.right = right
        self.val = val
    def __str__(self):
        s1 = ""
        if self.left != None :
            s1 = str(self.left)
        s2 = ""
        if self.right != None :
            s2 = str(self.right)
        return self.val + s1 + s2

def getIndex(c, s) :
    for i in range(len(s)) :
        if c == s[i] :
            return i
    return -1

def BuildTree_post_in(sp, si) :
    if sp == "" :
```

基本信息
#: 44282237
题目: 24750
提交人: 23n2300010724
内存: 3684kB
时间: 23ms
语言: Python3
提交时间: 2024-03-18 09:17:02

22158: 根据二叉树前中序序列建树

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

用时: 约15分钟

思路:
首先依题意建树, 方法与上一题类似, 之后按后序输出

代码

```

class node :
    def __init__ (self, left, right, val) :
        self.left = left
        self.right = right
        self.val = val

def getIndex(c, s) :
    for i in range(len(s)) :
        if c == s[i] :
            return i
    return -1

def ToPostString(s) :
    if s == None :
        return ""
    return ToPostString(s.left) + ToPostString(s.right) + s.val

def BuildTree_pre_in(sp, si) :
    if sp == "" :
        return None
    l = getIndex(sp[0], si)
    # print(sp, si, l, sep = " ")
    return node(BuildTree_pre_in(sp[1 : l + 1], si[0 : l]), BuildTree_pre_in(sp[l + 1 : len(sp)], si[l + 1 : len(si)])

while True :
    try :
        s1 = input()
        s2 = input()
        print(ToPostString(BuildTree_pre_in(s1, s2)))
    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

```
class node :
    def __init__ (self, left, right, val) :
        self.left = left
        self.right = right
        self.val = val

def getIndex(c, s) :
    for i in range(len(s)) :
        if c == s[i] :
            return i
    return -1

def ToPostString(s) :
    if s == None :
        return ""
    return ToPostString(s.left) + ToPostString(s.right) + s.val
```

基本信息

#: [44196970](#)
题目: [22158](#)
提交人: [23n2300010724](#)
内存: 3624kB
时间: 22ms
语言: [Python3](#)
提交时间: 2024-03-13 15:43:41

2. 学习总结和收获

完成了（截至这份文件写完的时候的）所有春季选做，在树的练习中更熟悉了类的写法，同时也对常见的转化有了更多了解。