

Assignment #5: "树"算：概念、表示、解析、遍历

Updated 2124 GMT+8 March 17, 2024

2024 spring, Compiled by 赵语涵 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

操作系统：windows 11

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：按照课件上学习的内容写的，实操以后会更加理解树的写法。关于树的依序遍历，建立包含每个树类的列表，并指针到其子树。注意树的高度与深度的区别

代码

```
1 #赵语涵2300012254
2 class Tree:
3     def __init__(self):
4         self.left = None
5         self.right = None
6
7     def height(x):
8         if x == None:
```

```

9         return -1
10        return max(height(x.left),height(x.right))+1
11
12    n = int(input())
13    trees = [Tree() for i in range(n)]
14    parent = dict(zip(range(n),[False]*n))
15    leaves = 0
16    for i in range(n):
17        a,b = map(int,input().split())
18        if a != -1:
19            trees[i].left = trees[a]
20        if b != -1:
21            trees[i].right = trees[b]
22        parent[a],parent[b] = True,True
23        if a == b == -1:
24            leaves += 1
25    for i in range(n):
26        if not parent[i]:
27            print(height(trees[i]),end = ' ')
28            break
29    print(leaves)

```

代码运行截图

#44285827提交状态

[查看](#) [提交](#) [统计](#)

状态: Accepted

源代码

```

#赵语涵2300012254
class Tree:
    def __init__(self):
        self.left = None
        self.right = None

    def height(x):

```

基本信息

#: 44285827
 题目: 27638
 提交人: 23n2300012254
 内存: 3644kB
 时间: 23ms
 语言: Python3
 提交时间: 2024-03-18 15:59:34

24729: 括号嵌套树

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

思路：发现前序输出只要去掉输入中的符号即可。至于后序，是先入后出，建立列表用栈的方式解决。遇到右括号时将括号中内容弹出，遇到','时说明应该写下一个子树，则弹出栈中的子树

代码

```

1 #赵语涵2300012254
2 ori = input()
3 pre = ''
4 stack = []
5 post = ''

```

```

6  for i in ori:
7      if i not in ['(', ')', ',', ' ']:
8          pre += i
9          stack.append(i)
10     elif i == '(':
11         stack.append(i)
12     elif i == ')':
13         while (p:=stack.pop()) != '(':
14             post += p
15     else:
16         post += stack.pop()
17 while stack:
18     post += stack.pop()
19 print(pre)
20 print(post)

```

代码运行截图

#44286006提交状态

[查看](#) [提交](#) [统计](#)

状态: **Accepted**

源代码

```

#赵语涵2300012254
ori = input()
pre = ''
stack = []
post = ''
for i in ori:
    if i not in ['(', ')', ',', ' ']:
        pre += i
        stack.append(i)
    elif i == '(':
        stack.append(i)
    elif i == ')':
        while (p:=stack.pop()) != '(':
            post += p
    else:
        post += stack.pop()
while stack:
    post += stack.pop()
print(pre)
print(post)

```

基本信息

#: 44286006
 题目: 24729
 提交人: 23n2300012254
 内存: 3616kB
 时间: 24ms
 语言: Python3
 提交时间: 2024-03-18 16:13:03

02775: 文件结构“图”

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

思路：感觉后序转图思路还是很简单就是要达到要求的输出格式比较麻烦，另外注意题干所说的file名称可能不只是'file#'，由于这个点一直RE了半天

代码

```

1  #赵语涵2300012254
2  import sys
3  sys.setrecursionlimit(1<<30)
4  pre = '|      '
5  stop = False
6  def files(x,num,file):
7      global stop
8      if stop:
9          return
10     if x[0] == 'd':
11         print(pre*(num+1)+x)

```

```

12         files(input(),num+1,[])
13     elif x == ']' or x == '*':
14         for i in sorted(file):
15             print(pre*num+i)
16         if x == '*':
17             stop = True
18         return
19     else:
20         file.append(x)
21     files(input(),num,file)
22
23 start,n = True,1
24 while True:
25     stop = False
26     a = input()
27     if a == '#':
28         break
29     if not start:
30         print()
31     print(f'DATA SET {n}:')
32     print('ROOT')
33     files(a,0,[])
34     n += 1
35     start = False

```

代码运行截图

#44297622提交状态

[查看](#) [提交](#) [统计](#)

状态: **Accepted**

源代码

```

#赵语涵2300012254
import sys
sys.setrecursionlimit(1<<30)
pre = ''
stop = False
def files(x,num,file):
    if x == '(':
        files(input(),num+1,[])
    elif x == ']' or x == '*':
        for i in sorted(file):
            print(pre*num+i)
        if x == '*':
            stop = True
        return
    else:
        file.append(x)
    files(input(),num,file)

```

基本信息

#: 44297622
 题目: 02775
 提交人: 23n2300012254
 内存: 3572kB
 时间: 24ms
 语言: Python3
 提交时间: 2024-03-19 15:01:50

25140: 根据后序表达式建立队列表达式

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

思路：首先弄明白两种表达式的计算方法（即手动怎么达成目的序列，主要是了解队列表达式），然后再用编程语言表示。用树的结构表示运算符与其指向的操作数，队列表达式从后向前看是从根节点开始弹出节点、左、右子树节点，并且在写完同一层节点后才进入下一轮的表达式。首先按照栈的方法构造表达式树，从该层节点开始把它的左右子树节点加入队列前端，这样就可以在处理完这一层后再处理下一层。

代码

```

1  #赵语涵2300012254
2
3  from collections import deque
4
5  class Tree():
6      def __init__(self,value):
7          self.left = None
8          self.right = None
9          self.value = value
10
11  for _ in range(int(input())):
12      store = deque()
13      for x in input(): #根据后序表达式建树
14          if ord(x) > 96:
15              store.append(Tree(x))
16          else:
17              t = Tree(x)
18              t.right,t.left = store.pop(),store.pop()
19              store.append(t)
20      out = ''
21      while store:
22          x = store.pop()
23          out += x.value
24          if x.left != None:
25              store.appendleft(x.left)
26          if x.right != None:
27              store.appendleft(x.right)
28      print(out[::-1])

```

代码运行截图

#44304937提交状态

[查看](#) [提交](#) [统计](#)

状态: Accepted

源代码

```

#赵语涵2300012254

from collections import deque

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

```

基本信息

#: 44304937
 题目: 25140
 提交人: 23n2300012254
 内存: 3668kB
 时间: 27ms
 语言: Python3
 提交时间: 2024-03-19 21:14:11

24750: 根据二叉树中后序序列建树

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

思路：这个题以前做过类似的，是用一堆指针标示位置表示，看起来很麻烦思路也很绕。看了题解之后，用先建树再按表达式写出树的方法，就很简洁

代码

```
1  #赵语涵2300012254
2  mid,post = list(input()),list(input())
3
4  class Tree():
5      def __init__(self,x):
6          self.value = x
7          self.left = None
8          self.right = None
9
10     root,rootnode = True,None
11     def build(mid):
12         global root,post,rootnode
13         if mid == [] or post == []:
14             return None
15         x = post.pop()
16         next_pos = mid.index(x)
17         tree = Tree(x)
18         if root:
19             rootnode = tree
20             root = False
21         tree.right = build(mid[next_pos+1:])
22         tree.left = build(mid[:next_pos])
23         return tree
24
25     result = []
26     def pre(node):
27         if not node:
28             return
29         result.append(node.value)
30         if node.left:
31             pre(node.left)
32         if node.right:
33             pre(node.right)
34
35     build(mid)
36     pre(rootnode)
37     print(''.join(result))
```

代码运行截图

#44409050提交状态

[查看](#) [提交](#) [统计](#)

状态: **Accepted**

源代码

```
#赵语涵2300012254
mid,post = list(input()),list(input())

class Tree():
    def __init__(self,x):
        self.value = x
        self.left = None
```

基本信息

#: 44409050
题目: 24750
提交人: 23n2300012254
内存: 3664kB
时间: 25ms
语言: Python3
提交时间: 2024-03-26 17:04:03

22158: 根据二叉树前中序序列建树

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

思路：根据上一题进行改写即可。需要注意的是，上题输入有后序，所以根节点用pop得到，而本题前序则利用deque数据类型popleft得到。以及在写入写出时左右子树的递归顺序进行交换（后序pop首先得到右子树的根节点，前序popleft首先得到左子树的根节点）

代码

```
1  #赵语涵2300012254
2  from collections import deque
3  class Tree():
4      def __init__(self,x):
5          self.value = x
6          self.left = None
7          self.right = None
8
9  def build(mid):
10     global root,post,rootnode
11     if mid == [] or post == []:
12         return None
13     x = deque.popleft(pre)
14     next_pos = mid.index(x)
15     tree = Tree(x)
16     if root:
17         rootnode = tree
18         root = False
19     tree.left = build(mid[:next_pos])
20     tree.right = build(mid[next_pos+1:])
21     return tree
22
23 def post(node):
24     if not node:
25         return
26     result.appendleft(node.value)
27     if node.right:
28         post(node.right)
29     if node.left:
30         post(node.left)
31
32 while True:
33     try:
34         pre,mid = list(input()),list(input())
35         pre = deque(pre)
36         root,rootnode = True,None
37         build(mid)
38         result = deque()
39         post(rootnode)
40         print(''.join(result))
```

```
41     except EOFError:
42         break
```

代码运行截图

#44409450提交状态

查看提交统计

状态: Accepted

源代码

```
#赵语涵2300012254
from collections import deque
class Tree():
    def __init__(self,x):
        self.value = x
        self.left = None
        self.right = None
```

基本信息

#: 44409450
题目: 22158
提交人: 23n2300012254
内存: 3616kB
时间: 24ms
语言: Python3
提交时间: 2024-03-26 17:25:21

2. 学习总结和收获

本周写的题目较少，纠结于自己的想法的题花费了较长时间。以后用时较长时看题解并从中学习应用会更有效率