Assignment #5: "树"算：概念、表示、解析、遍历

Updated 2124 GMT+8 March 17, 2024

2024 spring, Complied by ==同学的姓名、院系==

赵云天 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：win10

Python编程环境：Spyder IDE 5.2.2

C/C++编程环境：

1. 题目

27638: 求二叉树的高度和叶子数目

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

思路：对类的理解逐渐增强了

代码

```
class tree():
    def __init__(self):
        self.left = None
        self.right = None

def height(x):
```

```
    if x is None:
        return -1
    return max(height(x.left),height(x.right))+1

def numb(x):
    if x is None:
        return 0
    if x.left is None and x.right is None:
        return 1
    return numb(x.left)+numb(x.right)

n = int(input())
a = [tree() for _ in range(n)]
b = [False]*n

for i in range(n):
    lefti,righti = map(int,input().split())
    if lefti != -1:
        a[i].left = a[lefti]
        b[lefti] = True
    if righti != -1:
        a[i].right = a[righti]
        b[righti] = True

rootn = b.index(False)
root = a[rootn]

h = height(root)
nu = numb(root)

print(f'{h} {nu}')
```

代码运行截图 ==（至少包含有"Accepted"）==

#44395125提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
class tree():
    def __init__(self):
        self.left = None
        self.right = None

def height(x):
    if x is None:
        return -1
    return max(height(x.left),height(x.right))+1

def numb(x):
    if x is None:
        return 0
    if x.left is None and x.right is None:
        return 1
    return numb(x.left)+numb(x.right)

n = int(input())
a = [tree() for _ in range(n)]
b = [False]*n

for i in range(n):
    lefti,righti = map(int,input().split())
```

基本信息

#: 44395125
题目: 27638
提交人: 23n2300012140(zyt)
内存: 3680kB
时间: 24ms
语言: Python3
提交时间: 2024-03-25 14:41:50

24729: 括号嵌套树

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

思路：是、对一些点理解得不好

代码

```
class TreeNode:
    def __init__(self, value):
            self.value = value
            self.children = []

def parse_tree(s):
    stack = []
    node = None
    for char in s:
        if char.isalpha():
            node = TreeNode(char)
            if stack:
                stack[-1].children.append(node)
        elif char == '(':
            if node:
                stack.append(node)
                node = None
        elif char == ')':
            if stack:
                node = stack.pop()
    return node

def preorder(node):
    output = [node.value]
    for child in node.children:
        output.extend(preorder(child))
    return ''.join(output)

def postorder(node):
    output = []
    for child in node.children:
        output.extend(postorder(child))
    output.append(node.value)
    return ''.join(output)

def main():
    s = input().strip()
    s = ''.join(s.split())
    root = parse_tree(s)
    print(preorder(root))
    print(postorder(root))
```

```
if __name__ == "__main__":
    main()
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
class TreeNode:
    def __init__(self, value):
            self.value = value
            self.children = []

def parse_tree(s):
    stack = []
    node = None
    for char in s:
        if char.isalpha():
            node = TreeNode(char)
            if stack:
                stack[-1].children.append(node)
        elif char == '(':
            if node:
                stack.append(node)
                node = None
```

基本信息

#:	44398591
题目:	24729
提交人:	23n2300012140(zyt)
内存:	5816kB
时间:	26ms
语言:	Python3
提交时间:	2024-03-25 18:48:24

02775: 文件结构“图”

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

思路：空格数目需要注意

代码

```
class Node:
    def __init__(self, name):
        self.name = name
        self.dirs = []
        self.files = []
def print_structure(node, indent=0):
    prefix = '|     ' * indent
    print(prefix + node.name)
    for dir in node.dirs:
        print_structure(dir, indent + 1)
    for file in sorted(node.files):
        print(prefix + file)
dataset = 1
datas = []
temp = []
while True:
    line = input()
    if line == '#':
        break
    if line == '*':
        datas.append(temp)
```

```
            temp = []
        else:
            temp.append(line)
    for data in datas:
        print(f'DATA SET {dataset}:')
        root = Node('ROOT')
        stack = [root]
        for line in data:
            if line[0] == 'd':
                dir = Node(line)
                stack[-1].dirs.append(dir)
                stack.append(dir)
            elif line[0] == 'f':
                stack[-1].files.append(line)
            elif line == ']':
                stack.pop()
        print_structure(root)
        if dataset < len(datas):
            print()
        dataset += 1
```

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, name):
        self.name = name
        self.dirs = []
        self.files = []
def print_structure(node, indent=0):
    prefix = '|     ' * indent
    print(prefix + node.name)
    for dir in node.dirs:
        print_structure(dir, indent + 1)
    for file in sorted(node.files):
        print(prefix + file)
dataset = 1
datas = []
temp = []
while True:
    line = input()
    if line == '#':
        break
    if line == '*':
        datas.append(temp)
        temp = []
    else:
        temp.append(line)
for data in datas:
    print(f'DATA SET {dataset}:')
```

基本信息

#:	44398921
题目:	02775
提交人:	23n2300012140(zyt)
内存:	3636kB
时间:	22ms
语言:	Python3
提交时间:	2024-03-25 19:12:58

25140: 根据后序表达式建立队列表达式

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

思路：虽然用树，但是思路是类似的

代码

```
class TreeNode:
    def __init__(self, value):
        self.value = value
        self.left = None
        self.right = None

def build_tree(postfix):
    stack = []
    for char in postfix:
        node = TreeNode(char)
        if char.isupper():
            node.right = stack.pop()
            node.left = stack.pop()
        stack.append(node)
    return stack[0]

def level_order_traversal(root):
    queue = [root]
    traversal = []
    while queue:
        node = queue.pop(0)
        traversal.append(node.value)
        if node.left:
            queue.append(node.left)
        if node.right:
            queue.append(node.right)
    return traversal

n = int(input().strip())
for _ in range(n):
    postfix = input().strip()
    root = build_tree(postfix)
    queue_expression = level_order_traversal(root)[::-1]
    print(''.join(queue_expression))
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态：Accepted

源代码

```
class TreeNode:
    def __init__(self, value):
        self.value = value
        self.left = None
        self.right = None

def build_tree(postfix):
    stack = []
    for char in postfix:
        node = TreeNode(char)
        if char.isupper():
            node.right = stack.pop()
            node.left = stack.pop()
        stack.append(node)
    return stack[0]

def level_order_traversal(root):
    queue = [root]
    traversal = []
    while queue:
        node = queue.pop(0)
        traversal.append(node.value)
        if node.left:
            queue.append(node.left)
        if node.right:
```

基本信息

#: 44400817
题目: 25140
提交人: 23n2300012140(zyt)
内存: 3664kB
时间: 27ms
语言: Python3
提交时间: 2024-03-25 21:14:23

24750: 根据二叉树中后序序列建树

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

思路：感觉是很标准的模板题

代码

```
def build_tree(inorder, postorder):
    if not inorder or not postorder:
        return []

    root_val = postorder[-1]
    root_index = inorder.index(root_val)

    left_inorder = inorder[:root_index]
    right_inorder = inorder[root_index + 1:]

    left_postorder = postorder[:len(left_inorder)]
    right_postorder = postorder[len(left_inorder):-1]

    root = [root_val]
    root.extend(build_tree(left_inorder, left_postorder))
    root.extend(build_tree(right_inorder, right_postorder))

    return root

def main():
    inorder = input().strip()
    postorder = input().strip()
    preorder = build_tree(inorder, postorder)
```

```
    print(''.join(preorder))
if __name__ == "__main__":
    main()
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

#44400919提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
def build_tree(inorder, postorder):
    if not inorder or not postorder:
        return []

    root_val = postorder[-1]
    root_index = inorder.index(root_val)

    left_inorder = inorder[:root_index]
    right_inorder = inorder[root_index + 1:]

    left_postorder = postorder[:len(left_inorder)]
    right_postorder = postorder[len(left_inorder):-1]

    root = [root_val]
    root.extend(build_tree(left_inorder, left_postorder))
    root.extend(build_tree(right_inorder, right_postorder))

    return root

def main():
    inorder = input().strip()
    postorder = input().strip()
```

基本信息

#: 44400919
题目: 24750
提交人: 23n2300012140(zyt)
内存: 3656kB
时间: 23ms
语言: Python3
提交时间: 2024-03-25 21:20:48

22158: 根据二叉树前中序序列建树

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

思路：感觉也是很标准的模板题

代码

```
class TreeNode:
    def __init__(self, value):
        self.value = value
        self.left = None
        self.right = None

def build_tree(preorder, inorder):
    if not preorder or not inorder:
        return None
    root_value = preorder[0]
    root = TreeNode(root_value)
    root_index_inorder = inorder.index(root_value)
    root.left = build_tree(preorder[1:1+root_index_inorder],
inorder[:root_index_inorder])
    root.right = build_tree(preorder[1+root_index_inorder:],
inorder[root_index_inorder+1:])
```

```
    return root

def postorder_traversal(root):
    if root is None:
        return ''
    return postorder_traversal(root.left) + postorder_traversal(root.right)
+root.value
while True:
    try:
        preorder = input().strip()
        inorder = input().strip()
        root = build_tree(preorder, inorder)
        print(postorder_traversal(root))
    except EOFError:
        break
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

#44401012提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
class TreeNode:
    def __init__(self, value):
        self.value = value
        self.left = None
        self.right = None

def build_tree(preorder, inorder):
    if not preorder or not inorder:
        return None
    root_value = preorder[0]
    root = TreeNode(root_value)
    root_index_inorder = inorder.index(root_value)
    root.left = build_tree(preorder[1:1+root_index_inorder],
inorder[:root_index_inorder])
    root.right = build_tree(preorder[1+root_index_inorder:],
inorder[root_index_inorder+1:])
    return root

def postorder_traversal(root):
    if root is None:
        return ''
    return postorder_traversal(root.left) + postorder_traversal(root.ri
while True:
```

基本信息

#: 44401012
题目: 22158
提交人: 23n2300012140(zyt)
内存: 3532kB
时间: 23ms
语言: Python3
提交时间: 2024-03-25 21:26:26

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。==

相较上一次作业对树和类有了更好的理解

但是对各种模板题的场景依然不够熟悉