

Assignment #5: "树"算：概念、表示、解析、遍历

Updated 2124 GMT+8 March 17, 2024

2024 spring, Compiled by ==黄源森, 工学院==

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：W11

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：

代码

```
#
def dfs(i,p):
    global d
    a,b=l[i]
    if a==-1 and b==-1:
        d=max(d,p)
    return
```

```

        if a!=-1:
            dfs(a,p+1)
        if b!=-1:
            dfs(b,p+1)
n=int(input())
l=[]
s=set()
c=0
for i in range(n):
    a,b=map(int,input().split())
    l.append([a,b])
    if a==-1 and b==-1:
        c+=1
    s.add(a)
    s.add(b)
for i in range(n):
    if i not in s:
        break

d=0
dfs(i,0)
print(d,c)

```

代码运行截图 == (至少包含有"Accepted") ==

状态: **Accepted**

源代码

```

def dfs(i,p):
    global d
    a,b=l[i]
    if a==-1 and b==-1:
        d=max(d,p)
        return
    if a!=-1:
        dfs(a,p+1)
    if b!=-1:
        dfs(b,p+1)
n=int(input())
l=[]
s=set()
c=0
for i in range(n):
    a,b=map(int,input().split())
    l.append([a,b])
    if a==-1 and b==-1:
        c+=1
    s.add(a)
    s.add(b)
for i in range(n):
    if i not in s:
        break

d=0
dfs(i,0)
print(d,c)

```

基本信息

#: 43891149
 题目: 27638
 提交人: 23n2300011031
 内存: 3648kB
 时间: 25ms
 语言: Python3
 提交时间: 2024-02-13 08:22:28

24729: 括号嵌套树

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

思路：

代码

```
#
s=input()
ans=[]
for u in s:
    if u.isalpha():
        ans.append(u)
print(*ans,sep='')
ans=[]
stack=[]
for u in s:
    if u==')':
        while 1:
            x=stack.pop()
            if x.isalpha():
                ans.append(x)
            if x=='(':
                break
    elif u==',':
        x=stack.pop()
        ans.append(x)
    else:
        stack.append(u)
ans+=s[0]
print(*ans,sep='')
```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```
s=input()
ans=[]
for u in s:
    if u.isalpha():
        ans.append(u)
print(*ans,sep='')
ans=[]
stack=[]
for u in s:
    if u=='(':
        while 1:
            x=stack.pop()
            if x.isalpha():
                ans.append(x)
            if x=='(':
                break
        elif u==',':
            x=stack.pop()
            ans.append(x)
        else:
            stack.append(u)
ans+=s[0]
print(*ans,sep='')
```

基本信息

#: 43889037
题目: P0740
提交人: 23n2300011031
内存: 5096kB
时间: 27ms
语言: Python3
提交时间: 2024-02-12 11:05:33

©2002-2022 POJ 京ICP备20010980号-1

English 帮助

02775: 文件结构“图”

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

思路:

代码

```
#
from collections import defaultdict
k=1
while 1:
    f=0
    g=0
    a1file=set()
    ans=[]
    ans.append("ROOT")
    temp=defaultdict(list)
    while 1:
        x=input()
        if x=='*':
            break
        if x=='#':
            f=1
            break
        if x[0]=='f':
            if g:
```

```

        temp[g].append("|" + "*"g+x)
    else:
        alfile.add(x)
    if x[0]=='d':
        g+=1
        ans.append("|" + "*"g+x)
    if x==' ']:
        g-=1
        for u in sorted(temp[g+1]):
            ans.append(u)
        temp[g+1]=[]
if f:
    break
print(f"DATA SET {k}:")
k+=1
al=list(alfile)
for u in ans:
    print(u)
for u in sorted(al):
    print(u)
print()

```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

状态: Accepted

源代码

```

from collections import defaultdict
k=1
while 1:
    f=0
    g=0
    alfile=set()
    ans=[]
    ans.append("ROOT")
    temp=defaultdict(list)
    while 1:
        x=input()
        if x=='*':
            break
        if x=='#':
            f=1
            break
        if x[0]=='f':
            if g:
                temp[g].append("|" + "*"g+x)
            else:
                alfile.add(x)
        if x[0]=='d':
            g+=1
            ans.append("|" + "*"g+x)
        if x==' ']:
            g-=1
            for u in sorted(temp[g+1]):
                ans.append(u)
            temp[g+1]=[]
    if f:
        break
    print(f"DATA SET {k}:")
    k+=1
    al=list(alfile)
    for u in ans:
        print(u)
    for u in sorted(al):
        print(u)
    print()

```

基本信息

#: 44261404
 题目: 02775
 提交人: 23n2300011031
 内存: 3652kB
 时间: 25ms
 语言: Python3
 提交时间: 2024-03-17 10:30:48

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

```
#
class Node:
    def __init__(self, val, l, r):
        self.val = val
        self.left = l
        self.right = r
class Tree:
    def create(self, s):
        stack = []
        for u in s:
            if u.isupper():
                a = stack.pop()
                b = stack.pop()
                t = Node(u, b, a)
                stack.append(t)
            else:
                stack.append(u)
        return stack[0]
    def dfs(self, x, layer):
        if type(x) != Node:
            a1[layer].append(x)
            return
        a1[layer].append(x.val)
        self.dfs(x.left, layer+1)
        self.dfs(x.right, layer+1)
for _ in range(int(input())):
    s = input()
    t = Tree()
    x = t.create(s)
    a1 = [[] for _ in range(100)]
    t.dfs(x, 0)
    ans = []
    for u in a1:
        ans.extend(u)
    print(*reversed(ans), sep=' ')
```

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, val, l, r):
        self.val = val
        self.left = l
        self.right = r
class Tree:
    def create(self, s):
        stack = []
        for u in s:
            if u.isupper():
                a = stack.pop()
                b = stack.pop()
                t = Node(u, b, a)
                stack.append(t)
            else:
                stack.append(u)
        return stack[0]
    def dfs(self, x, layer):
        if type(x) != Node:
            al[layer].append(x)
            return
        al[layer].append(x.val)
        self.dfs(x.left, layer+1)
        self.dfs(x.right, layer+1)
for _ in range(int(input())):
    s = input()
    t = Tree()
    x = t.create(s)
    al = [[] for _ in range(100)]
    t.dfs(x, 0)
    ans = []
    for u in al:
        ans.extend(u)
    print(*reversed(ans), sep='')
```

基本信息

#: 44192759
题目: 25140
提交人: 23n2300011031
内存: 3616kB
时间: 29ms
语言: Python3
提交时间: 2024-03-13 10:55:59

24750: 根据二叉树中后序序列建树

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

思路:

代码

```
#
#include<iostream>
#include<string>
using namespace std;
string f(string a, string b){
    int t = b.length();
    if(t == 0) return "";
    if(t == 1) return a;
    char h = b[t-1];
    int tt = a.find(h);
    return h + f(a.substr(0, tt), b.substr(0, tt)) + f(a.substr(tt+1), b.substr(tt, t-tt-1));
}
int main(){
    string a, b;
    cin >> a;
    cin >> b;
    cout << f(a, b);
```

```
}
```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

状态: Accepted

源代码

```
#include<iostream>
#include<string>
using namespace std;
string f(string a,string b){
    int t=b.length();
    if(t==0) return "";
    if(t==1) return a;
    char h=b[t-1];
    int tt=a.find(h);
    return h+f(a.substr(0,tt),b.substr(0,tt))+f(a.substr(tt+1),b.su
}
int main(){
    string a,b;
    cin>>a;
    cin>>b;
    cout<<f(a,b);
}
```

基本信息

#: 43891159
题目: 24750
提交人: 23n2300011031
内存: 252kB
时间: 3ms
语言: G++
提交时间: 2024-02-13 08:32:22

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

22158: 根据二叉树前中序序列建树

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

思路:

代码

```
#
def f(a,b):
    if len(a)==0:
        return ''
    if len(a)==1:
        return a
    else:
        x=a[0]
        i=b.find(x)
        return f(a[1:i+1],b[:i])+f(a[i+1:],b[i+1:])+x
while 1:
    try:
        a=input()
        b=input()
        print(f(a, b))
    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

```
def f(a,b):
    if len(a)==0:
        return ''
    if len(a)==1:
        return a
    else:
        x=a[0]
        i=b.find(x)
        return f(a[1:i+1],b[:i])+f(a[i+1:],b[i+1:])+x
while 1:
    try:
        a=input()
        b=input()
        print(f(a, b))
    except:
        break
```

基本信息

#: 44192251
题目: 22158
提交人: 23n2300011031
内存: 3608kB
时间: 20ms
语言: Python3
提交时间: 2024-03-13 10:14:48

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。==

现在每日选做的题都重做一遍，其他数算小组也有些好题，数算应该既要记住模版也要有拓展思维