

# Assignment #9: 图论：遍历，及 树算

Updated 1739 GMT+8 Apr 14, 2024

2024 spring, Compiled by ==陈亚偲 工学院==

## 说明：

- 1) 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。
- 2) 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。
- 3) 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：Windows

Python编程环境：Spyder IDE 5.2.2

## 1. 题目

### 04081: 树的转换

<http://cs101.openjudge.cn/dsapre/04081/>

思路：

设计两个数，先建树，再转化，最后各自求高度

代码

```
#
class node:
    def __init__(self,num):
        self.num=num
        self.p=0 #This is an index
        self.s=[] #This is an unsorted list of indexes
class bio:
    def __init__(self,num):
        self.num=num
        self.p=None #This is a bio
        self.l=None
        self.r=None
a=[node(0)]
kk=list(input())
target=0
```

```

for i in kk:
    if i == 'u':
        target=a[target].p
    if i == 'd':
        a.append(node(len(a)))
        a[-1].p=target
        a[target].s.append(a[-1].num)
        target=a[-1].num
b=[bio(i) for i in range(a[-1].num+1)]
def build(root):#root is an index which starts with 0
    if a[root].s:
        p=len(a[root].s)
        b[root].l=b[a[root].s[0]]
        build(a[root].s[0])
        if p>1:
            for i in range(1,p):
                b[a[root].s[i]].p=b[a[root].s[i-1]]
                b[a[root].s[i-1]].r=b[a[root].s[i]]
                build(a[root].s[i])
    return
build(0)
def bcx(root):
    ppp=[]
    if a[root].s:
        for i in a[root].s:
            ppp.append(1+bcx(i))
        return max(ppp)
    return 0
def swj(root):
    if root!=None:
        return 1+max(swj(root.l),swj(root.r))
    return 0
print(str(bcx(0))+ ' => ' +str(swj(b[0])-1))

```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```
class node:
    def __init__(self,num):
        self.num=num
        self.p=0 #This is an index
        self.s=[] #This is an unsorted list of indexes
class bio:
    def __init__(self,num):
        self.num=num
        self.p=None #This is a bio
        self.l=None
        self.r=None
a=[node(0)]
kk=list(input())
target=0
for i in kk:
    if i == 'u':
        target=a[target].p
    if i == 'd':
        a.append(node(len(a)))
        a[-1].p=target
        a[target].s.append(a[-1].num)
        target=a[-1].num
b=[bio(i) for i in range(a[-1].num+1)]
def build(root):#root is an index which starts with 0
    if a[root].s:
        p=len(a[root].s)
        b[root].l=b[a[root].s[0]]
        build(a[root].s[0])
        if p>1:
            for i in range(1,p):
                b[a[root].s[i]].p=b[a[root].s[i-1]]
                b[a[root].s[i-1]].r=b[a[root].s[i]]
                build(a[root].s[i])
    return
build(0)
def bcx(root):
    ppp=[]
    if a[root].s:
        for i in a[root].s:
            ppp.append(1+bcx(i))
        return max(ppp)
    return 0
def swj(root):
    if root!=None:
        return 1+max(swj(root.l),swj(root.r))
```

基本信息

#: 44769680  
题目: 04081  
提交人: 23n2300011106(boza)  
内存: 3752kB  
时间: 28ms  
语言: Python3  
提交时间: 2024-04-23 22:57:01

## 08581: 扩展二叉树

<http://cs101.openjudge.cn/dsapre/08581/>

思路:

按要求建树

代码

```
#
class node:
    def __init__(self,num):
        self.num=num
        self.p=None
        self.l=None
        self.r=None
p=list(input())
a=[node(p[0])]
target=a[0]
def build(b,t):#t is a node, b is a list
```

```

    if b:
        if t.l==None:
            a.append(node(b[0]))
            t.l=a[-1]
            a[-1].p=t
            if a[-1].num!='.':
                build(b[1:],a[-1])
            else:
                build(b[1:],t)
        elif t.r==None:
            a.append(node(b[0]))
            t.r=a[-1]
            a[-1].p=t
            if a[-1].num!='.':
                build(b[1:],a[-1])
            else:
                build(b[1:],t)
        else:
            t=t.p
            build(b,t)
    return
build(p[1:],target)
mids=[]
def mid(z):
    if z.num != '.' :
        mid(z.l)
        mids.append(z.num)
        mid(z.r)
    else:
        return
mid(a[0])
print(''.join(mids))
l11=[]
def last(z):
    if z.num != '.' :
        last(z.l)
        last(z.r)
        l11.append(z.num)
    else:
        return
last(a[0])
print(''.join(l11))

```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```
class node:
    def __init__(self,num):
        self.num=num
        self.p=None
        self.l=None
        self.r=None

p=list(input())
a=[node(p[0])]
target=a[0]
def build(b,t):#t is a node, b is a list
    if b:
        if t.l==None:
            a.append(node(b[0]))
            t.l=a[-1]
            a[-1].p=t
            if a[-1].num!='.':
                build(b[1:],a[-1])
            else:
                build(b[1:],t)
        elif t.r==None:
            a.append(node(b[0]))
            t.r=a[-1]
            a[-1].p=t
            if a[-1].num!='.':
                build(b[1:],a[-1])
            else:
                build(b[1:],t)
        else:
            t=t.p
            build(b,t)
    return
build(p[1:],target)
mids=[]
def mid(z):
    if z.num != '.' :
        mid(z.l)
        mids.append(z.num)
        mid(z.r)
    else:
        return
mid(a[0])
print(''.join(mids))
l1l=[]
def last(z):
    if z.num != '.' :
```

基本信息

#: 44770214  
题目: 08581  
提交人: 23n2300011106(boza)  
内存: 6672kB  
时间: 30ms  
语言: Python3  
提交时间: 2024-04-23 23:54:04

## 22067: 快速堆猪

<http://cs101.openjudge.cn/practice/22067/>

思路:

代码

```
#
```

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

## 04123: 马走日

dfs, <http://cs101.openjudge.cn/practice/04123>

思路:

代码

```
#
```

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

## 28046: 词梯

bfs, <http://cs101.openjudge.cn/practice/28046/>

思路:

代码

```
#
```

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

## 28050: 骑士周游

dfs, <http://cs101.openjudge.cn/practice/28050/>

思路:

代码

#

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

---

==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。==

对树的理解逐渐加深, 要点在于按照信息建树, 再把树输出即可