

Assignment #8: 图论：概念、遍历，及 树算

Updated 1919 GMT+8 Apr 8, 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

19943: 图的拉普拉斯矩阵

matrices, <http://cs101.openjudge.cn/practice/19943/>

请定义Vertex类，Graph类，然后实现

思路：

和树类似，但parent不唯一

代码

```
#
n,m=map(int,input().split())
a=[]
for i in range(n):
    a.append([0]*n)
class g:
    def __init__(self,num):
        self.num=num
        self.v=[]
b=[g(i) for i in range(n)]
def boza(c,d):
    e=c
    f=d
    b[e].v.append(f)
```

```

        b[f].v.append(e)
    return
for i in range(m):
    p,q=map(int,input().split())
    boza(p,q)
for i in range(n):
    a[i][i]=len(b[i].v)
    for j in b[i].v:
        a[i][j]=-1
for i in a:
    print(*i)

```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```

n,m=map(int,input().split())
a=[]
for i in range(n):
    a.append([0]*n)
class g:
    def __init__(self,num):
        self.num=num
        self.v=[]
b=[g(i) for i in range(n)]
def boza(c,d):
    e=c
    f=d
    b[e].v.append(f)
    b[f].v.append(e)
    return
for i in range(m):
    p,q=map(int,input().split())
    boza(p,q)
for i in range(n):
    a[i][i]=len(b[i].v)
    for j in b[i].v:
        a[i][j]=-1
for i in a:
    print(*i)

```

基本信息

#: 44677895
 题目: 19943
 提交人: 23n2300011106(boza)
 内存: 3672kB
 时间: 27ms
 语言: Python3
 提交时间: 2024-04-16 21:00:55

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

18160: 最大连通域面积

matrix/dfs similar, <http://cs101.openjudge.cn/practice/18160>

思路:

广搜, 挨个看, 找最大值

代码

```

#
hahaha=int(input())#aza boza ciza duza eza
dc={'W':0,'.':1}
ans=[]
for abababa in range(hahaha):
    n,m=map(int,input().split())
    boza=[]
    a=[[1]*(m+2)]

```

```

for i in range(n):
    a.append([1]+[dc[j] for j in list(input())]+[1])
a.append([1]*(m+2))
ciza=True
duza=True
while ciza:
    ciza=False
    for i in range(1,n+1):
        if not ciza:
            for j in range(1,m+1):
                if a[i][j]==0:
                    a[i][j]=2
                    ciza=True
                    duza=True
                    break
while duza:
    duza=False
    for i in range(1,n+1):
        for j in range(1,m+1):
            if a[i][j]==2:
                a[i][j]=3
                for k in range(-1,2):
                    for w in range(-1,2):
                        if a[i+k][j+w]==0:
                            a[i+k][j+w]=2
    for i in range(1,n+1):
        if 2 in a[i]:
            duza=True
            break
    if not duza:
        ct=0
        for i in range(1,n+1):
            for j in range(1,m+1):
                if a[i][j]==3:
                    ct+=1
        boza.append(ct)
b=[boza[0]]
for i in range(1,len(boza)):
    b.append(boza[i]-boza[i-1])
ans.append(max(b))
for i in ans:
    print(i)

```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

基本信

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Thu Nov 30 20:52:55 2023

@author: 陈亚恩2300011106
"""
hahaha=int(input())#aza boza ciza duza eza
dc={'W':0,'.':1}
ans=[]
for abababa in range(hahaha):
    n,m=map(int,input().split())
    boza=[]
    a=[[1]*(m+2)]
    for i in range(n):
        a.append([1]+[dc[j] for j in list(input())]+[1])
    a.append([1]*(m+2))
    ciza=True
    duza=True
    while ciza:
        ciza=False
        for i in range(1,n+1):
            if not ciza:
                for j in range(1,m+1):
                    if a[i][j]==0:
                        a[i][j]=2
                        ciza=True
                        duza=True
                        break
    while duza:
        duza=False
        for i in range(1,n+1):
            for j in range(1,m+1):
                if a[i][j]==2:
                    a[i][j]=3
                    for k in range(-1,2):
                        for w in range(-1,2):
                            if a[i+k][j+w]==0:
                                a[i+k][j+w]=2
        for i in range(1,n+1):
            if 2 in a[i]:
                duza=True
                break
    if not duza:
        ct=0
        for i in range(1,n+1):
            for j in range(1,m+1):
```

是
提
日
提

sy383: 最大权值连通块

<https://sunnywhy.com/sfbj/10/3/383>

思路:

bfs, 每走一步加一个权值

代码

```

#
n,m=map(int,input().split())
class g:
    def __init__(self,num,w):
        self.num=num
        self.w=w
        self.v=[]
        self.haha=False # whether been visited
b=[int(i) for i in input().split()]
a=[g(i,b[i]) for i in range(n)]
for i in range(m):
    p,q=map(int,input().split())
    a[p].v.append(q)
    a[q].v.append(p)
ans=0
ct=0
def dfs(a,root):#root is a number
    global ct
    ct+=a[root].w
    a[root].haha=True
    for i in a[root].v:
        if not a[i].haha:
            dfs(a,i)
    return
for i in range(n):
    if not a[i].haha:
        ct=0
        dfs(a,i)
        ans=max(ans,ct)
print(ans)

```

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

完美通过

100% 数据通过测试

运行时长: 0 ms

语言: Python

```
1  n,m=map(int,input().split())
2  class g:
3      def __init__(self,num,w):
4          self.num=num
5          self.w=w
6          self.v=[]
7          self.haha=False # whether been visited
8  b=[int(i) for i in input().split()]
9  a=[g(i,b[i]) for i in range(n)]
10 for i in range(m):
11     p,q=map(int,input().split())
12     a[p].v.append(q)
13     a[q].v.append(p)
14 ans=0
15 ct=0
16 def dfs(a,root):#root is a number
17     global ct
18     ct+=a[root].w
19     a[root].haha=True
20     for i in a[root].v:
21         if not a[i].haha:
22             dfs(a,i)
23     return
24 for i in range(n):
25     if not a[i].haha:
26         ct=0
27         dfs(a,i)
```

03441: 4 Values whose Sum is 0

data structure/binary search, <http://cs101.openjudge.cn/practice/03441>

思路:

mle, 不会改进了, 下为mle代码

代码

```
#
n=int(input())
a=[]
b=[]
```

```

c=[]
d=[]
e=[]
f=[]
for i in range(n):
    x,y,z,w=map(int,input().split())
    a.append(x)
    b.append(y)
    c.append(z)
    d.append(w)
for i in range(n):
    for j in range(n):
        e.append(a[i]+b[j])
        f.append(-c[i]-d[j])
e.sort()
f.sort()
index1=0
index2=0
ct=0
k=n*n
while index1<k and index2<k:
    if e[index1]==f[index2]:
        t1=0
        t2=0
        tt=e[index1]
        while index1<k and e[index1]==tt:
            index1+=1
            t1+=1
        while index2<k and f[index2]==tt:
            index2+=1
            t2+=1
        ct+=t1*t2
    elif e[index1]>f[index2]:
        index2+=1
    else:
        index1+=1
print(ct)

```

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

04089: 电话号码

trie, <http://cs101.openjudge.cn/practice/04089/>

Trie 数据结构可能需要自学下。

思路：想象树在生长，可以长出旁支，如果旁支捋到底，就No，否则Yes

代码

```
#
n=int(input())
ans=['YES','NO']
class bcx:
    def __init__(self,num):
        self.num=num
        self.c={}
        self.end=False
        self.haha=False
def getin(x,z): #x is a bcx and z is a str
    global a
    if z[0] in x.c.values():
        for j in x.c.keys():
            if x.c[j]==z[0]:
                if z[1:]:
                    getin(a[j],z[1:])
                else:
                    a[-1].haha=True
                break
    else:
        a.append(bcx(z[0]))
        x.c[len(a)-1]=z[0]
        if z[1:]:
            getin(a[-1],z[1:])
        else:
            a[-1].end=True
    return
for iii in range(n):
    boza=False
    m=int(input())
    a=[bcx('root')]
    for i in range(m):
        getin(a[0],input())
    for i in a:
        if i.haha:
            boza=True
            break
        if i.end and i.c:
            boza=True
            break
    print(ans[boza])
```

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

状态: Accepted

源代码

```
n=int(input())
ans=['YES','NO']
class bcx:
    def __init__(self,num):
        self.num=num
        self.c={}
        self.end=False
        self.haha=False
    def getin(x,z): #x is a bcx and z is a str
        global a
        if z[0] in x.c.values():
            for j in x.c.keys():
                if x.c[j]==z[0]:
                    if z[1:]:
                        getin(a[j],z[1:])
                    else:
                        a[-1].haha=True
                        break
            else:
                a.append(bcx(z[0]))
                x.c[len(a)-1]=z[0]
                if z[1:]:
                    getin(a[-1],z[1:])
                else:
                    a[-1].end=True
        return
for iii in range(n):
    boza=False
    m=int(input())
    a=[bcx('root')]
    for i in range(m):
        getin(a[0],input())
    for i in a:
        if i.haha:
            boza=True
            break
        if i.end and i.c:
            boza=True
            break
    print(ans[boza])
```

04082: 树的镜面映射

<http://cs101.openjudge.cn/practice/04082/>

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。==

trie数据结构还没学, 但题目ac了, 估计想法差不多

桶对于我是一大难点, 目前还搞不清楚, 五一再弄

图和树非常像, 但parent不唯一

遇到函数记得写return

建树(图)的时候, 为了方便, 参数可以多写些