Assignment #8: 图论：概念、遍历，及 树算

Updated 1919 GMT+8 Apr 8, 2024

2024 spring, Complied by 王申睿——物理学院

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 typorahttps://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： macOS Ventura 13.4.1 (c)

Python编程环境： Spyder IDE 5.2.2, PyCharm 2023.1.4 (Professional Edition)

C/C++编程环境： Mac terminal vi (version 9.0.1424), g++/gcc (Apple clang version 14.0.3, clang-1403.0.22.14.1)

1. 题目

19943: 图的拉普拉斯矩阵

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

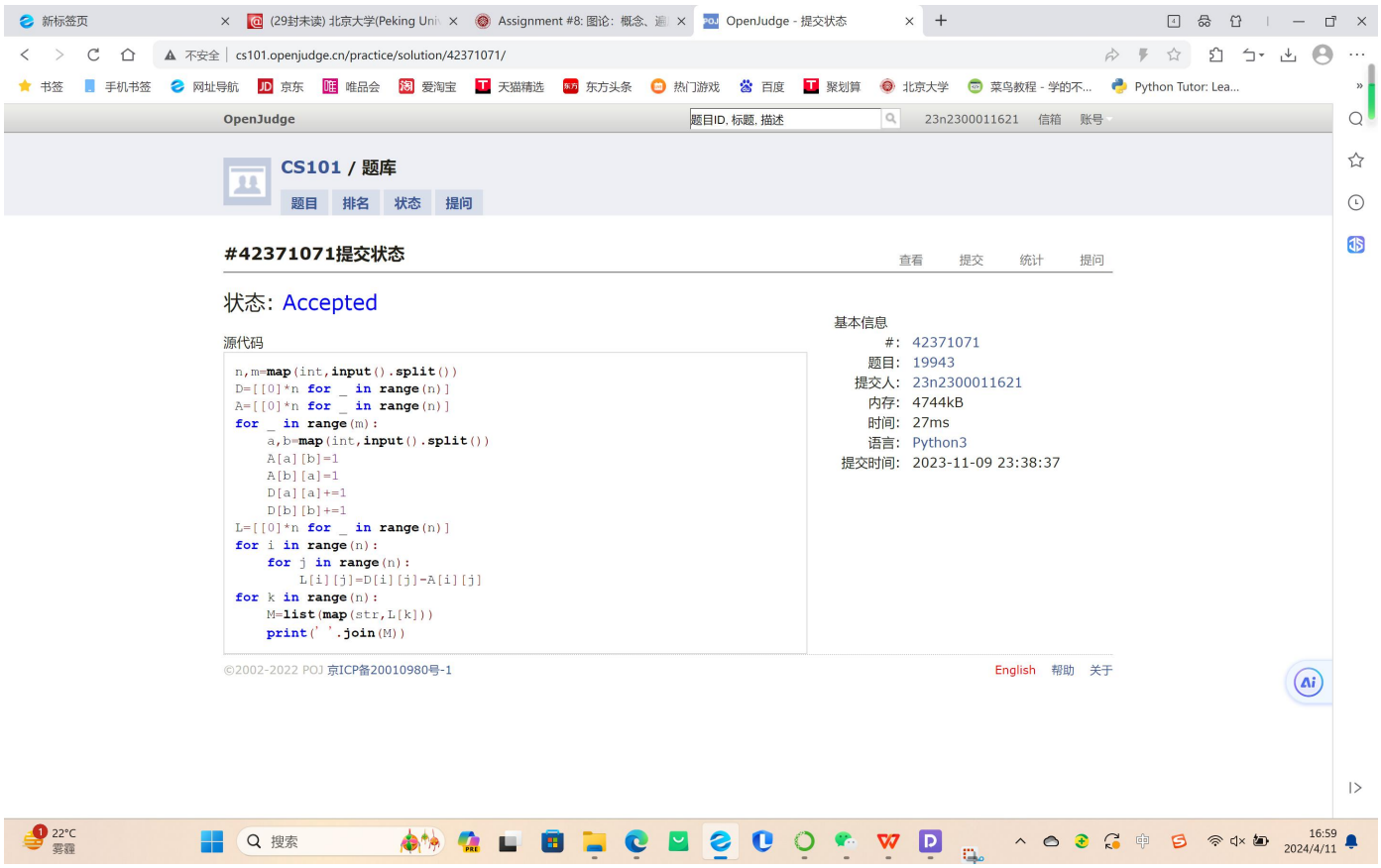
请定义Vertex类， Graph类，然后实现

思路:

代码

```
n,m=map(int,input().split())

D=[[0]*n for _ in range(n)]

A=[[0]*n for _ in range(n)]

for _ in range(m):

    a,b=map(int,input().split())

    A[a][b]=1

    A[b][a]=1

    D[a][a]+=1

    D[b][b]+=1

L=[[0]*n for _ in range(n)]

for i in range(n):

    for j in range(n):

        L[i][j]=D[i][j]-A[i][j]
```

```
for k in range(n):
    M=list(map(str,L[k]))
    print(' '.join(M))
```



18160: 最大连通域面积

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

思路:

代码

"""

已参考题解

"""

```
liantongshu=0#面积数

dx=[1,0,1,0,-1,1,-1,-1]

dy=[1,1,0,-1,1,-1,0,-1]#方向向量

def mianji(matrix,x,y):

    global liantongshu



    if matrix[x][y]!="W":
```

```
    return      #表示跳过这一格。
```

```
liantongshu+=1
```

```
matrix[x][y]="."
```

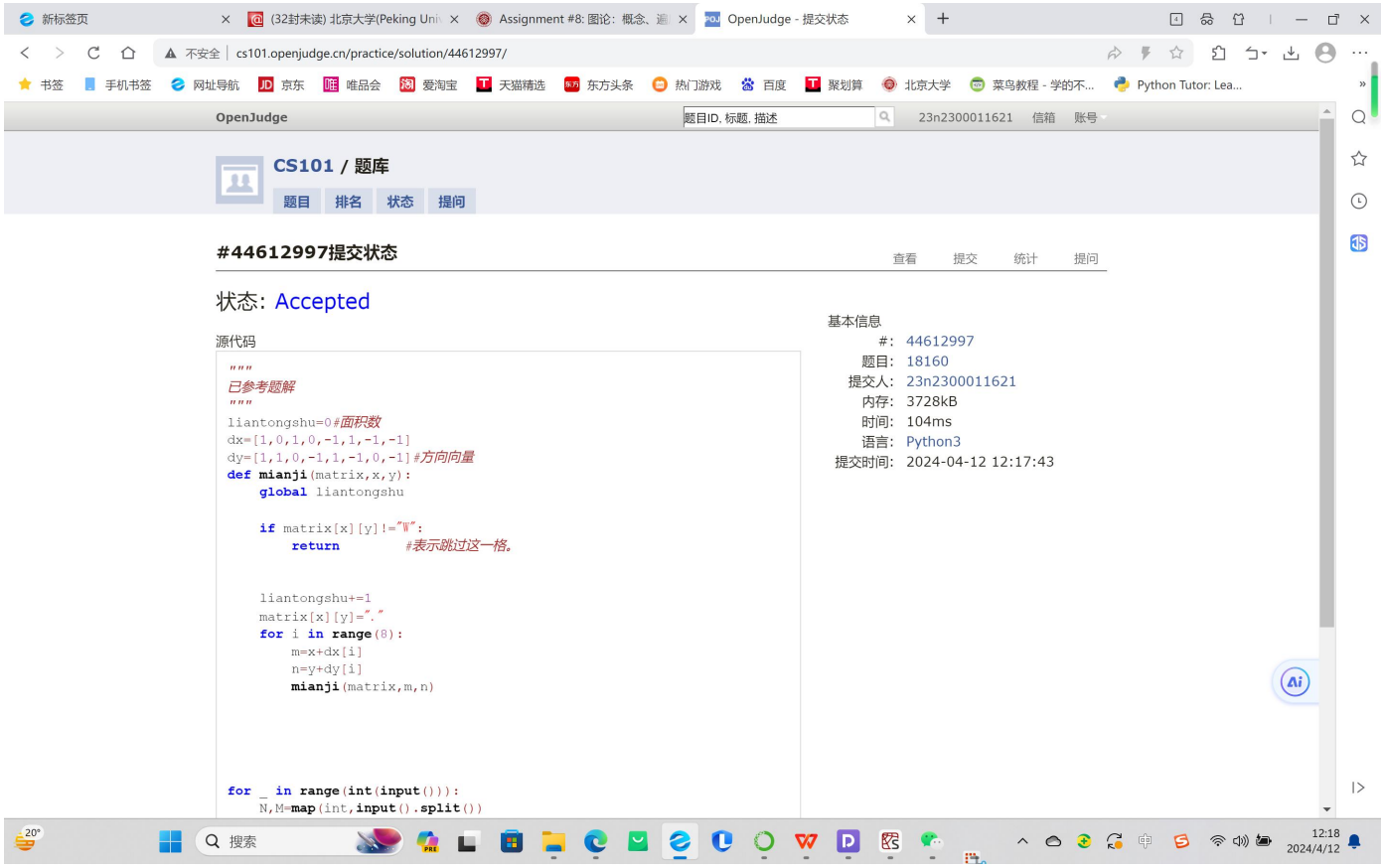
```
for i in range(8):
```

```
    m=x+dx[i]
```

```
    n=y+dy[i]
```

```
    mianji(matrix,m,n)
```

```
for _ in range(int(input())):
    N,M=map(int,input().split())
    a=[[0 for _ in range(M+2)]]#保护圈
    max_value=0#最大值初始化
    for _ in range(N):
        a.append([0]+list(input())+[0])
    a.append([0 for _ in range(M+2)])#保护圈
    for j in range(1,N+1):
        for k in range(1,M+1):
            liantongshu=0
            mianji(a,j,k)
            if liantongshu>max_value:
                max_value=liantongshu
    print(str(max_value))
```



sy383: 最大权值连通块

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

思路:

代码

```
class Vertex():

    def __init__(self,val,wei):

        self.value=val

        self.weight=wei
```

```
        self.parent=None

def find(x):

    if x.parent is None:

        return x

    else:

        return find(x.parent)

n,m=map(int,input().split())

a=list(map(int,input().split()))

b=[Vertex(i,a[i]) for i in range(len(a))]

for _ in range(m):

    c,d=map(int,input().split())

    if find(b[c])!=find(b[d]):

        find(b[c]).parent=find(b[d])

e=[]
```

```
b.sort(key= lambda s:find(s).value)

current_parent=None

for i in range(len(b)):

    if find(b[i])!=current_parent:

        e.append(b[i].weight)

        current_parent=find(b[i])

    else:

        e[-1]+=b[i].weight

print(max(e))
```

03441: 4 Values whose Sum is 0

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

思路:

代码

```
from collections import defaultdict




n = int(input())

a = []

t = 0

qian = defaultdict(int)




for _ in range(n):

    a.append(list(map(int, input().split())))




for i in range(n):

    for j in range(n):

        qian[a[i][0] + a[j][1]] += 1
```

```
for k in range(n):

    for l in range(n):

        if -(a[k][2] + a[l][3]) in qian.keys():

            t+=qian[-(a[k][2] + a[l][3])]



print(t)
```

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:

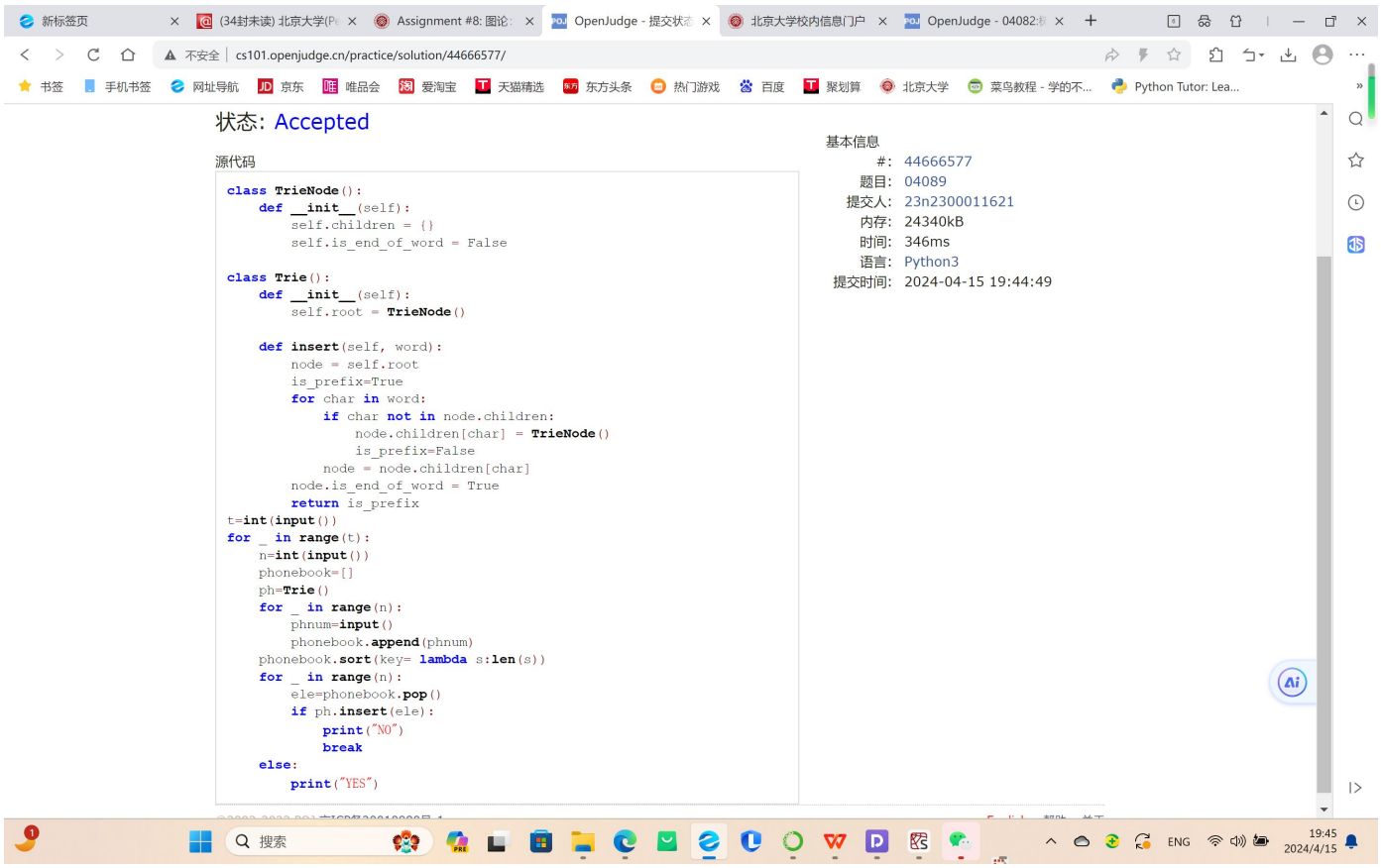
代码

```
class TrieNode():

    def __init__(self):

        self.children = {}
```

```
        self.is_end_of_word = False




class Trie():

    def __init__(self):

        self.root = TrieNode()




    def insert(self, word):

        node = self.root

        is_prefix=True

        for char in word:

            if char not in node.children:

                node.children[char] = TrieNode()

                is_prefix=False

            node = node.children[char]
```

```
        node.is_end_of_word = True

        return is_prefix

t=int(input())

for _ in range(t):

    n=int(input())

    phonebook=[]

    ph=Trie()

    for _ in range(n):

        phnum=input()

        phonebook.append(phnum)

    phonebook.sort(key= lambda s:len(s))

    for _ in range(n):

        ele=phonebook.pop()

        if ph.insert(ele):
```

```
            print("NO")

            break

    else:

        print("YES")
```



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

sy383：最大权值，用并查集改造节点类方便思考。

字典树的原理：每个节点都会引用一个字典作为其枝叶。如果在相同位置没有遇到新的字符，则不会添加新的节点和分支；如果有，则另起炉灶。根节点始终为空。

明天热学期中，没时间debug镜面映射了，打脸了。