

Assignment #8: 图论：概念、遍历，及 树算

Updated 1919 GMT+8 Apr 8, 2024

2024 spring, Compiled by 吴至超 城环

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：windows11

Python编程环境：pycharm2023.2.3

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=(int(x) for x in input().split())#n顶点数，m边数
class graph:#创建一个图表
    def __init__(self,n):
        self.name=str(n)
        self.Dlens=[]
        self.Achildren=[]
        for i in range(n):
            self.Achildren.append([0]*n)
            self.Dlens.append([0]*n)
    def subtract(self):
        for o in range(n):#按照行来
            for m in range(n):
```

```

        self.Dlens[o][m]-=self.Achildren[o][m]
    return self.Dlens

class vertex:
    def __init__(self,x):
        self.id=x

datas=[]
while True:
    try:
        a=[int(x) for x in input().split()]
        if not a:
            break
        datas.append(a)
    except EOFError:
        break
Graph=graph(n)
for m in datas:
    Graph.Dlens[m[0]][m[0]]+=1
    Graph.Dlens[m[1]][m[1]]+=1
    Graph.Achildren[m[0]][m[1]]=1
    Graph.Achildren[m[1]][m[0]]=1
pri=Graph.subtract()
for lis in pri:
    print(" ".join(map(str,lis)))

```

代码运行截图 == (至少包含有"Accepted") ==

题目	排名	状态	提问
<div> <div>#44594292提交状态</div> <div> 查看 提交 统计 提问 </div> </div>			
<div> <div>状态: Accepted</div> <div> <div>源代码</div> <pre> n,m=(int(x) for x in input().split())#n:顶点数, m:边数 class graph:#创建一个图表 def __init__(self,n): self.name=str(n) self.Dlens=[] self.Achildren=[] for i in range(n): self.Achildren.append([0]*n) self.Dlens.append([0]*n) def subtract(self): for o in range(n):#按照行来 for m in range(n): self.Dlens[o][m]-=self.Achildren[o][m] return self.Dlens class vertex: def __init__(self,x): self.id=x datas=[] while True: try: a=[int(x) for x in input().split()] if not a: break datas.append(a) except EOFError: </pre> </div> <div> <div>基本信息</div> <div> # : 44594292 题目: 19943 提交人: 23n2300013289 内存: 3704kB 时间: 26ms 语言: Python3 提交时间: 2024-04-10 15:55:40 </div> </div> </div>			

18160: 最大连通域面积

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

思路：dfs，现在最外围套上一圈

代码

```
#
T=int(input())
neighbour=[[-1,-1],[0,-1],[1,-1],[-1,0],[1,0],[-1,1],[0,1],[1,1]]
for i in range(T):
    N,M=map(int,input().split())#N行，M列
    #第一步：搭建好迷宫，给迷宫最外围围上一圈"."，便于dfs中的结束判断
    graph=[]
    graph.append(["."] * (M + 2))#上下圈
    count=[]
    for m in range(N):
        a=["."]*(M+2)
        a[1:M+1]=input()#左右圈，自动切割好了
        graph.append(a)
    graph.append(["."]*(M+2))#上下圈

    for idx_line in range(1,N+1):
        for idx_row in range(1,M+1):
            if graph[idx_line][idx_row]=="w":
                cnt=0
                def dfs(re_x, re_y): # re_x表示起始dfs的x，re_y表示起始dfs的y
                    global cnt
                    if graph[re_x][re_y] == "w":
                        cnt += 1
                        graph[re_x][re_y] = "."
                        for c in neighbour:
                            dfs(re_x + c[0], re_y + c[1])

                dfs(idx_line,idx_row)
                count.append(cnt)
    if count:
        print(max(count))
    else:
        print(0)
```

代码运行截图 == (至少包含有"Accepted") ==

#44604853提交状态

查看

提交

统计

提问

状态: Accepted

源代码

```
T=int(input())
neighbour=[[-1,-1],[0,-1],[1,-1],[-1,0],[1,0],[-1,1],[0,1],[1,1]]
for i in range(T):
    N,M=map(int,input().split())#N行, M列
    #第一步: 搭建好迷宫, 给迷宫最外圈围上一圈".", 便于dfs中的结束判断
    graph=[]
    graph.append(["." * (M + 2)])#上下圈
    count=[]
    for m in range(N):
        a=["." * (M+2)]
        a[1:M+1]=input()#左右圈, 自动切割好了
        graph.append(a)
    graph.append(["." * (M+2)])#上下圈

    for idx_line in range(1,N+1):
        for idx_row in range(1,M+1):
            if graph[idx_line][idx_row]=="W":
                cnt=0
                def dfs(re_x, re_y): # re_x表示起始dfs的x, re_y表示起始dfs的y
                    global cnt
                    if graph[re_x][re_y] == "W":
                        cnt += 1
                        graph[re_x][re_y] = "."
                        for c in neighbour:
                            dfs(re_x + c[0], re_y + c[1])
                dfs(idx_line,idx_row)
```

基本信息

#: 44604853

题目: 18160

提交人: 23n2300013289

内存: 3816kB

时间: 84ms

语言: Python3

提交时间: 2024-04-11 16:22:54

sy383: 最大权值连通块

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

思路: dfs

代码

```
#
n,m=map(int,input().split())
class vertex:
    def __init__(self,name,value):
        self.connect=[]
        self.value=value
        self.name=name

valuelis=[int(x) for x in input().split()]
groups={}
for i in range(n):
    groups[i]=vertex(i,valuelis[i])
for m in range(m):
    x,y=map(int,input().split())
    groups[x].connect.append(groups[y])
    groups[y].connect.append(groups[x])

name=list(groups)
count=[]
def dfs(vertex,visited):
    visited.append(vertex.name)
    cur_va=vertex.value
    for i in vertex.connect:
        if i.name not in visited:
            cur_va+=dfs(i,visited)
```

```
        return cur_va

    for m in name:
        visited=[]
        count.append(dfs(groups[m],visited))

    print(max(count))
```

代码运行截图 == (AC代码截图，至少包含有"Accepted") ==

「2025考研初试：跨考数据结构与算法改造训练营」现已发布，[点击查看](#)

您还未设置昵称，可点击[此处](#)进入个人中心修改

篇 (4) —— 图算法专题

图的遍历

无向图的连通块

无向图判环

有向图判环

最大权值连通块

无向图的顶点层号

受限层号的顶点数

题目

题解

最大权值连通块

通过数 641 提交数 1179 难度 中等 显示标签

题目描述

现有一个共 n 个顶点、 m 条边的无向图（假设顶点编号为从 0 到 $n-1$ ），每个顶点有各自的权值。我们把一个连通块中所有顶点的权值之和称为这个连通块的权值。求图中所有连通块的最大权值。

输入描述

第一行两个整数 n 、 m ($1 \leq n \leq 100, 0 \leq m \leq \frac{n(n-1)}{2}$)，分别表示顶点数和边数；
第二行 n 个用空格隔开的正整数（每个正整数不超过 100 ），表示 n 个顶点的权值。
接下来 m 行，每行两个整数 u 、 v ($0 \leq u \leq n-1, 0 \leq v \leq n-1, u \neq v$)，表示一条边的两个端点的编号。数据保证不会有重边。

输出描述

输出一个整数，表示连通块的最大权值。

样例1

代码书写

```
3 class vertex:
4     def __init__(self, name, value):
5         self.connect=[]
6         self.value=value
7         self.name=name
8
9     valuelis=[int(x) for x in input().split()]
10    groups={}
11    for i in range(n):
12        groups[i]=vertex(i, valuelis[i])
13    for m in range(m):
14        x,y=map(int, input().split())
15        groups[x].connect.append(groups[y])
```

测试输入

历史提交

提交时间	结果	时长(ms)	语言	
2024-04-11 17:27:34	完美通过	0	Python	查看

[收起面板](#)[运行](#)[提交](#)

03441: 4 Values whose Sum is 0

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

思路：两两分组，a+b与c+d，统计a+b各个结构出现次数，放在一个字典里，计算c+d的时候直接相反数查询即可

代码

```
# n=int(input())
A=[]
B=[]
C=[]
D=[]
for i in range(n):
    a=[int(x) for x in input().split()]
    A.append(a[0])
    B.append(a[1])
    C.append(a[2])
    D.append(a[3])
```


```

abdic={}

for i in A:
    for m in B:
        if i+m not in abdic:
            abdic[i+m]=0
        abdic[i+m]+=1#目的是为了统计所有a+b各种"和"的数量,保证各种不同的和只出现一次
cnt=0
for i in C:
    for m in D:
        tempo=i+m
        if -(i+m) in abdic:
            cnt+=abdic[-(i+m)]
print(cnt)

```

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


CS101 / 题库

[题目](#)
[排名](#)
[状态](#)
[提问](#)

#44665125提交状态

[查看](#)
[提交](#)
[统计](#)
[提问](#)

状态: **Accepted**

源代码

```

#03441:4 Values whose Sum is 0
n=int(input())
A=[]
B=[]
C=[]
D=[]
for i in range(n):
    a=[int(x) for x in input().split()]
    A.append(a[0])
    B.append(a[1])
    C.append(a[2])
    D.append(a[3])
abdic={}

for i in A:
    for m in B:
        if i+m not in abdic:
            abdic[i+m]=0
        abdic[i+m]+=1#目的是为了统计所有a+b各种"和"的数量,保证各种不同的和只出现
cnt=0
for i in C:
    for m in D:
        tempo=i+m
        if -(i+m) in abdic:
            cnt+=abdic[-(i+m)]
print(cnt)

```

基本信息

: 44665125
题目: 03441
提交人: 23n2300013289
内存: 171736kB
时间: 4413ms
语言: Python3
提交时间: 2024-04-15 17:29:00

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: trie的数据结构, 字典套字典

代码

```
#
```

```

class trienode:
    def __init__(self):
        self.child={}
class trie:
    def __init__(self):
        self.root=trienode()
    def insert(self,str):#插入一段电话号码
        root1=self.root
        for i in str:#电话号码中的每个数字
            if i not in root1.child:
                root1.child[i]=trienode()#字典的value值对应一个新字典
            root1=root1.child[i]
    def search(self,num):#num是目标电话号码
        root2=self.root
        for i in num:
            if i not in root2.child:
                return 0
            root2=root2.child[i]
        return 1
t=trie()

for g in range(t):
    datas=[]
    n=int(input())
    for m in range(n):
        tempo=input()
        datas.append(tempo)
    trie1=trie()
    datas.sort(key=lambda x : -len(x))#?
    cnt=0
    for g in datas:
        cnt+=trie1.search(g)
        trie1.insert(g)
    if cnt!=0:
        print("NO")
    else:
        print("YES")

```

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

#44674926提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
class trienode:
    def __init__(self):
        self.child={}
class trie:
    def __init__(self):
        self.root=trienode()
    def insert(self, str): #插入一段电话号码
        root1=self.root
        for i in str: #电话号码中的每个数字
            if i not in root1.child:
                root1.child[i]=trienode() #字典的value值对应一个新字典
            root1=root1.child[i]
    def search(self, num): #num是目标电话号码
        root2=self.root
        for i in num:
            if i not in root2.child:
                return 0
            root2=root2.child[i]
        return 1
t=trie()
t.insert(input())

for g in range(t):
    datas=[]
    n=int(input())
    for m in range(n):
        tempo=input()
        datas.append(tempo)
    triel=trie()
```

基本信息

#: 44674926
题目: 04089
提交人: 23n2300013289
内存: 24736kB
时间: 400ms
语言: Python3
提交时间: 2024-04-16 17:21:56

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。==

由于本周有两门共8学分的期中考试, 不太顾得上数算, 先写5道, 打算考完以后的时间几乎都给数算补上, 还请见谅~