

Assignment #8: 图论：概念、遍历，及 树算

Updated 1919 GMT+8 Apr 8, 2024

2024 spring, Compiled by ==祁轩宇、经济学院==

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 11, version 23H2

Python编程环境：VSCode 1.87.1

C/C++编程环境：

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):
    u,v=map(int,input().split())
    D[u][u]+=1
    D[v][v]+=1
    A[u][v]=1
    A[v][u]=1
L=[[str(D[i][j]-A[i][j]) for j in range(n)] for i in range(n)]
for i in range(n):
    print(' '.join(L[i]))
```

代码运行截图 == （至少包含有"Accepted"） ==

#44676984提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
n,m=map(int,input().split())
D=[ [0]*n for _ in range(n)]
A=[ [0]*n for _ in range(n)]
for _ in range(m):
    u,v=map(int,input().split())
    D[u][u]+=1
    D[v][v]+=1
    A[u][v]=1
    A[v][u]=1
L=[ [str(D[i][j]-A[i][j]) for j in range(n)] for i in range(n)]
for i in range(n):
    print(' '.join(L[i]))
```

基本信息

#: 44676984
题目: 19943
提交人: 2100015440
内存: 3668kB
时间: 29ms
语言: Python3
提交时间: 2024-04-16 20:02:38

18160: 最大连通域面积

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

思路 :

代码

```
from collections import deque

MAX_DIRECTIONS = 8
dx = [0, 0, 1, -1,1,1,-1,-1]
dy = [1, -1, 0, 0,1,-1,1,-1]

def is_valid_move(x, y):
    return 0 <= x < n and 0 <= y < m and maze[x][y] == 'W' and not in_queue[x][y]

def bfs(start_x, start_y):
    queue = deque()
    if not in_queue[start_x][start_y]:
        queue.append((start_x, start_y))
        in_queue[start_x][start_y] = True
    while queue:
        x, y = queue.popleft()
        for i in range(MAX_DIRECTIONS):
            next_x = x + dx[i]
            next_y = y + dy[i]
            if is_valid_move(next_x, next_y):
                in_queue[next_x][next_y] = True
                queue.append((next_x, next_y))

def size(x,y):
    if not in_queue[x][y]:
        before=after=0
```

```
        for i in range(len(in_queue)):
            before+=sum(in_queue[i])
        bfs(x,y)
        for i in range(len(in_queue)):
            after+=sum(in_queue[i])
        return after-before
    else:
        return 0

N=int(input())
for _ in range(N):
    n, m = map(int, input().split())
    maze = [input() for i in range(n)]
    in_queue = [[False] * m for i in range(n)]
    size_dif = [0]*m*n

    for i in range(n):
        for j in range(m):
            if maze[i][j]=='.':
                in_queue[i][j]=True

    for i in range(n):
        for j in range(m):
            size_dif[i*m+j]=size(i,j)
    print(max(size_dif))
```

代码运行截图 ==（至少包含有"Accepted"）==

#44678420提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
from collections import deque

MAX_DIRECTIONS = 8
dx = [0, 0, 1, -1, 1, 1, -1, -1]
dy = [1, -1, 0, 0, 1, -1, 1, -1]

def is_valid_move(x, y):
    return 0 <= x < n and 0 <= y < m and maze[x][y] == 'W' and not in_queue[x][y]

def bfs(start_x, start_y):
    queue = deque()
    if not in_queue[start_x][start_y]:
        queue.append((start_x, start_y))
        in_queue[start_x][start_y] = True
    while queue:
        x, y = queue.popleft()
        for i in range(MAX_DIRECTIONS):
            next_x = x + dx[i]
            next_y = y + dy[i]
            if is_valid_move(next_x, next_y):
                in_queue[next_x][next_y] = True
                queue.append((next_x, next_y))

def size(x, y):
    if not in_queue[x][y]:
        before = after = 0
        for i in range(len(in_queue)):
            before += sum(in_queue[i])
        bfs(x, y)
        for i in range(len(in_queue)):
            after += sum(in_queue[i])
        return after - before
    else:
        return 0

N = int(input())
```

基本信息

#: 44678420
题目: 18160
提交人: 2100015440
内存: 3840kB
时间: 347ms
语言: Python3
提交时间: 2024-04-16 21:37:54

sy383: 最大权值连通块

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

思路：

代码

```
def dfs(node, visited, value, adjacency_list):
    visited[node] = True
    for neighbor in adjacency_list[node]:
        if not visited[neighbor]:
            dfs(neighbor, visited, value, adjacency_list)

n, m = map(int, input().split())
value = list(map(int, input().split()))
adjacency_list = [[] for _ in range(n)]
for _ in range(m):
    u, v = map(int, input().split())
    adjacency_list[u].append(v)
    adjacency_list[v].append(u)
```

```
visited = [False] * n
connected_components = [0]*n
for i in range(n):
    if not visited[i]:
        before=after=0
        for x in range(n):
            before+=visited[x]*value[x]
        dfs(i, visited, value, adjacency_list)
        for x in range(n):
            after+=visited[x]*value[x]
        connected_components[i]=after-before

print(max(connected_components))
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"） ==

题目题解

最大权值连通块

通过数 660 提交数 1203 难度 中等 显示标签

题目描述

现有一个共 n 个点、 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

输入复制

```
6 5
2 3 4 1 3 2
0 1
0 3
3 5
2 4
1 5
```

代码书写

Python

```
12 adjacency_list[u].append(v)
13 adjacency_list[v].append(u)
14
15 visited = [False] * n
16 connected_components = [0]*n
17 for i in range(n):
18     if not visited[i]:
19         before=after=0
20         for x in range(n):
21             before+=visited[x]*value[x]
22         dfs(i, visited, value, adjacency_list)
23         for x in range(n):
24             after+=visited[x]*value[x]
25         connected_components[i]=after-before
26
27 print(max(connected_components))
```

测试输入

提交结果

历史提交

完美通过

100% 数据通过测试

运行时长: 0 ms

查看题解

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：

代码

```
#
```

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：

代码

```
t=int(input())
for _ in range(t):
    n=int(input())
    phones=[]
    for __ in range(n):
        phones.append(input())
    phones.sort()

    flag=1
    for i in range(n-1):
        if phones[i + 1].startswith(phones[i]):
            flag=0
            break
    if flag:
        print('YES')
    else:
        print('NO')
```

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

#44679018提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
t=int(input())
for _ in range(t):
    n=int(input())
    phones=[]
    for __ in range(n):
        phones.append(input())
    phones.sort()

    flag=1
    for i in range(n-1):
        if phones[i + 1].startswith(phones[i]):
            flag=0
            break
    if flag:
        print('YES')
    else:
        print('NO')
```

基本信息

#: 44679018
题目: 04089
提交人: 2100015440
内存: 4320kB
时间: 87ms
语言: Python3
提交时间: 2024-04-16 22:19:52

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。==

对我来说这两周作业难度比之前稍微低一点，不依靠题解也可以AC4，不过想AC5、AC6的话还得再练。

19943: 图的拉普拉斯矩阵比较简单，考了图的基础概念。

18160: 最大连通域面积和sy383: 最大权值连通块比较像，思路都是使用bfs/dfs找到连通块之后计算面积/权值。不过二者图的结构略有区别，最大连通域面积类似于sy321: 迷宫最短路径，使用八个方向的步子来模拟节点之间的相邻关系；最大权值连通块就是典型的节点和边，使用邻接表即可。

04089: 电话号码使用Trie 数据结构更符合数算思维，不过排序后判断相邻号码是否包含前缀的方法也比较方便。