

# Assignment #8: 图论：概念、遍历，及 树算

Updated 1925 GMT+8 Apr 8, 2024

2024 spring, Compiled by 钟明衡 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 typora <https://typora.io.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

操作系统：Windows\_NT x64 10.0.19045

Python编程环境：Visual Studio Code 1.76.1

C/C++编程环境：Visual Studio Code 1.76.1

## 1. 题目

### 19943: 图的拉普拉斯矩阵

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

思路：

输入 $(i, j)$ ，则给 $L_{ii}$ 和 $L_{jj}$ 加上1，给 $L_{ij}$ 和 $L_{ji}$ 减去1

代码

```
1 n, m = map(int, input().split())
2 M = [[0]*n for i in range(n)]
3 for i in range(m):
4     x, y = map(int, input().split())
5     M[y][x] = -1
6     M[x][y] = -1
7     M[x][x] += 1
8     M[y][y] += 1
```

```

9   for i in range(n):
10      for j in range(n):
11          print(M[i][j], end='')
12          if j != n-1:
13              print(' ', end='')
14      print('')
15

```

代码运行截图

#42300045提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

n, m = map(int, input().split())
M = [[0]*n for i in range(m)]
for i in range(m):
    x, y = map(int, input().split())
    M[y][x] = -1
    M[x][y] = -1
    M[x][x] += 1
    M[y][y] += 1
for i in range(n):
    for j in range(n):
        print(M[i][j], end='')
        if j != n-1:
            print(' ', end='')
    print('')

```

基本信息

#: 42300045

题目: 19943

提交人: 23n2300011505(12号娱乐选手)

内存: 3624kB

时间: 26ms

语言: Python3

提交时间: 2023-11-06 21:45:24

## 18160: 最大连通域面积

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

思路:

找到W且未被标记的位置，然后利用dfs来寻找周围所有联通的W并计数，查找过的上标记防止重复，所有搜索中值最大的就是答案

代码

```

1   def dfs(i, j):
2       global l, used
3       count = 1
4       for x in range(i-1, i+2):
5           for y in range(j-1, j+2):
6               if l[x][y] == 'W' and not used[x][y]:
7                   used[x][y] = True
8                   count += dfs(x, y)
9       return count
10
11
12  t = int(input())

```

```

13 anss = []
14 for _ in range(t):
15     n, m = map(int, input().split())
16     used = [[False]*(m+2) for i in range(n+2)]
17     l = ['.']*(m+2)
18     ans = 0
19     for i in range(n):
20         l.append('.'+input()+'.')
21     l.append('.')*(m+2)
22     for i in range(1, n+1):
23         for j in range(1, m+1):
24             if l[i][j] == 'W' and not used[i][j]:
25                 used[i][j] = True
26                 ans = max(ans, dfs(i, j))
27     anss.append(ans)
28 for ans in anss:
29     print(ans)
30

```

代码运行截图

#42803212提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

def dfs(i, j):
    global l, used
    count = 1
    for x in range(i-1, i+2):
        for y in range(j-1, j+2):
            if l[x][y] == 'W' and not used[x][y]:
                used[x][y] = True
                count += dfs(x, y)
    return count

t = int(input())
anss = []
for _ in range(t):
    n, m = map(int, input().split())
    used = [[False]*(m+2) for i in range(n+2)]
    l = ['.']*(m+2)
    ans = 0
    for i in range(n):
        l.append('.'+input()+'.')
    l.append('.')*(m+2)
    for i in range(1, n+1):
        for j in range(1, m+1):
            if l[i][j] == 'W' and not used[i][j]:
                used[i][j] = True
                ans = max(ans, dfs(i, j))
    anss.append(ans)
for ans in anss:
    print(ans)

```

基本信息

#: 42803212  
 题目: 18160  
 提交人: 23n2300011505(12号娱乐选手)  
 内存: 3696kB  
 时间: 93ms  
 语言: Python3  
 提交时间: 2023-11-28 13:38:29

## sy383: 最大权值连通块

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

思路：

并查集，把同一个连通图里面的所有指针都指向同一个节点

注意，和冰可乐那题不同，这本身是一个无向图，可能存在环，单纯连接会死循环。解决办法是，限制指针只能由大的数指向小的数，这样绝对不会出现死循环

代码

```
1  from collections import defaultdict as D
2  p = D(lambda: -1)
3
4
5  def F(x):
6      global p
7      if p[x]+1:
8          px = F(p[x])
9          p[x] = px
10         return px
11     return x
12
13
14 n, m = map(int, input().split())
15 l = list(map(int, input().split()))
16 for _ in range(m):
17     x, y = map(int, input().split())
18     px, py = F(x), F(y)
19     if px != py:
20         p[max(px, py)] = min(px, py)
21 ans = [0]*n
22 for i in range(n):
23     ans[F(i)] += l[i]
24 print(max(ans))
25
```

代码运行截图

代码书写Python

```
1 from collections import defaultdict as D
2 p = D(lambda: -1)
3
4
5 def F(x):
6     global p
7     if p[x]+1:
8         px = F(p[x])
9         p[x] = px
10        return px
11    return x
12
```

测试输入 历史提交

| 提交时间                | 结果   | 时长(ms) | 语言     |                    |
|---------------------|------|--------|--------|--------------------|
| 2024-04-08 15:00:16 | 完美通过 | 0      | Python | <a href="#">查看</a> |

## 03441: 4 Values whose Sum is 0

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

思路:

把A和B所有可能的和算出来, 然后再用C和D去求和为0的数量

注意不能用defaultdict, 会超内存, 用普通字典就不会

代码

```
1 n = int(input())
2 L = [tuple(map(int, input().split())) for _ in range(n)]
3 D = {}
4 for a, _, _, _ in L:
5     for _, b, _, _ in L:
6         D[-a-b] = D.get(-a-b, 0)+1
7 ans = 0
8 for _, _, c, _ in L:
9     for _, _, _, d in L:
10        ans += D.get(c+d, 0)
11 print(ans)
12
```

代码运行截图

状态: Accepted

源代码

```
n = int(input())
L = [tuple(map(int, input().split())) for _ in range(n)]
D = {}
for a, _, _ in L:
    for b, _, _ in L:
        D[-a-b] = D.get(-a-b, 0)+1
ans = 0
for _, _, c, _ in L:
    for _, _, d in L:
        ans += D.get(c+d, 0)
print(ans)
```

基本信息

#: 44575644

题目: 03441

提交人: 23n2300011505(12号娱乐选手)

内存: 172164kB

时间: 5325ms

语言: Python3

提交时间: 2024-04-08 18:43:30

## 04089: 电话号码

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

思路:

将所有电话号码的字符串直接排序，之后利用字符串比大小的特性可以发现，如果出现了前缀的情况，则必定发生在两个相邻电话号码之间，且前一个不比后一个长，如此判断即可

代码

```
1  for _ in range(int(input())):
2      n = int(input())
3      l = sorted([input() for i in range(n)])
4      ans = True
5      for i in range(1, n):
6          if len(l[i]) >= len(l[i-1]):
7              if l[i][:len(l[i-1])] == l[i-1]:
8                  ans = False
9              if not ans:
10                 break
11     print('YES' if ans else 'NO')
12
```

代码运行截图

状态: Accepted

源代码

```
for _ in range(int(input())):
    n = int(input())
    l = sorted([input() for i in range(n)])
    ans = True
    for i in range(1, n):
        if len(l[i]) >= len(l[i-1]):
            if l[i][:len(l[i-1])] == l[i-1]:
                ans = False
            if not ans:
                break
    print('YES' if ans else 'NO')
```

基本信息

#: 44573030

题目: 04089

提交人: 23n2300011505(12号娱乐选手)

内存: 5016kB

时间: 87ms

语言: Python3

提交时间: 2024-04-08 16:06:03

## 04082: 树的镜面映射

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

思路:

只要能反向构造出原来的树即可。按照“左儿子右兄弟”的原则，用一个指标 $h$ 来代表深度，由于是满二叉树，当出现内部节点则下一个必定为其子节点， $h + 1$ ，出现外部节点则必定为叶节点， $h - 1$

每次将字符加到 $h$ 层，注意要反过来，新字符放到最前面，最后按顺序每层输出即可

代码

```
1 from collections import defaultdict as D
2 n = int(input())
3 d = D(str)
4 h = 0
5 ans = ''
6 for s in input().split():
7     if s[0] != '$':
8         d[h] = s[0]+d[h]
9     if s[1] == '0':
10         h += 1
11     else:
12         h -= 1
13 for a in d.values():
14     ans += a
15 print(' '.join(ans))
16
```

代码运行截图

状态: **Accepted**

源代码

```
from collections import defaultdict as D
n = int(input())
d = D(str)
h = 0
ans = ''
for s in input().split():
    if s[0] != '$':
        d[h] = s[0]+d[h]
        if s[1] == '0':
            h += 1
        else:
            h -= 1
for a in d.values():
    ans += a
print(' '.join(ans))
```

基本信息

#: 44540290  
题目: 04082  
提交人: 23n2300011505(12号娱乐选手)  
内存: 3640kB  
时间: 24ms  
语言: Python3  
提交时间: 2024-04-05 21:14:02

## 2. 学习总结和收获

01703发现它，抓住它、18250冰阔落、sy383最大权值连通块这三题对于理解并查集很有帮助

我对于并查集的理解，通俗来讲就是“找爸爸”（更严谨的说法是找root）。最基本的想法，每个节点都通过一个指针指向和它在同一个集合中的元素，通过正确构造指针（一般是新来的指向原来的），可以让每个集合最终由一个节点代表，从同一个集合的其他节点沿着指针走，走到头都是这个节点，这个节点就是root，这样的话处理许多问题都会很简化

找root用递归，这样返回时可以更新路上所有的指针，直接让它们全部指向root，后续查询就会更快，基本代码如下：

```
1 def F(x):
2     if p[x]: # 如果当前节点不是root（如果爸爸存在）
3         p[x] = F(p[x]) # 找到root同时将指针指向root（让小孩随爸爸姓）
4         return px
5     return x # 当前节点是root（找到老祖宗了）
```

之所以叫“找爸爸”，是因为这件事和现实中小孩随爸爸姓很像，如果把集合和姓氏对应起来，将连接关系视为父子关系，我们只要 $O(1)$ 的复杂度（看姓氏）就知道这个元属于哪个集合（这个小孩的祖上是谁）

在此基础上，我们可以做很多扩展，比如01703发现它、抓住它这题需要考虑到距离问题

此外还要排除一下死循环问题。毕竟指针带了个方向的意味，在有环图中就容易绕圈圈死循环，找不到root（转了一圈发现自己是自己老祖宗）。18250冰阔落并不会会有这个问题，因为“倒可乐”本身是有方向的，也不会形成环。而sy383最大权值连通块就容易出现这个问题，这个题本身是个无向图，如果统一让y指向x，样例数据就会死循环，我的解决办法是统一让大的编号指向小的编号。构造方法视具体情况而定。

还有一个发现，defaultdict虽然省事但是比dict更占内存，03441 4 Values whose Sum is 0这题我就因为一直用defaultdict导致MLE，改回dict就过了。dict用get可以避免KeyError