

Assignment #8: 图论：概念、遍历，及树算

Updated 1919 GMT+8 Apr 8, 2024

2024 spring, Compiled by 赵语涵 生命科学学院

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

操作系统：windows 11

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：对于题目中给出的3个矩阵，实际上用一个矩阵表示即可（用字典表示，初始值为0）。D矩阵只有对角有数值，且L矩阵对角线值与之相同，记录为m组给出顶点重复的数目（即度）；而A矩阵只有非对角有数值，与L相同位置数值恰好相反，记录为m组边的连接顶点。最后输出矩阵用join函数依次输出每一行矩阵的值。

代码

```
1 #赵语涵2300012254
2 n,m = map(int,input().split())
3 matrix = {x:dict(zip(range(n),[0]*n)) for x in range(n)}
4 for _ in range(m):
5     a,b = map(int,input().split())
6     matrix[a][a]+=1;matrix[b][b]+=1
7     matrix[a][b]-=1;matrix[b][a]-=1
8 for i in range(n):
9     print(' '.join(map(str,matrix[i].values())))
```

状态: Accepted

源代码

```
#赵语涵2300012254
n,m = map(int,input().split())
matrix = {x:dict(zip(range(n),[0]*n)) for x in range(n)}
```

基本信息

#: 44830151
题目: 19943
提交人: 23n2300012254
内存: 3628kB

18160: 最大连通域面积

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

思路：用到递归的方式（dfs），函数对当前位置增加1面积，并寻找周围为'W'的位置进行递归（用mark变量记录已经计数过的位置），直到找不到周围'W'位置为止。

代码

```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Wed Nov 29 17:00:37 2023
4
5  @author: 赵语涵2300012254
6  """
7  def count(data,mark,pos,s):
8      global m,n
9      for x in range(max(pos[0]-1,0),min(pos[0]+2,n)):
10         for y in range(max(pos[1]-1,0),min(pos[1]+2,m)):
11             if data[x][y] == 'W' and mark[x][y] == 0:
12                 mark[x][y] = 1
13                 s += 1
14                 if [x,y] != pos:
15                     s = count(data,mark,[x,y],s)
16         return s
17
18  for _ in range(int(input())):
19      n,m = map(int,input().split())
20      table,mark,big = {},{},0
21      for i in range(n):
22          table[i] = input()
23          mark[i] = dict(zip(range(m),[0]*m))
24      for x in range(n):
25          for y in range(m):
26              if table[x][y] == 'W' and mark[x][y] == 0:
27                  s = count(table, mark, [x,y], 0)
28                  big = [big,s][s>big]
29      print(big)
```

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Wed Nov 29 17:00:37 2023

@author: 赵语涵2300012254
"""
```

基本信息

#: 42847032
题目: 18160
提交人: 23n2300012254
内存: 3824kB
时间: 140ms
语言: Python3
提交时间: 2023-11-30 17:42:41

sy383: 最大权值连通块

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

思路: 建立无向图, 关于每个节点有序号num, 权重weight, 边edge 3个参数。输入数据时边互相指向相应的节点。在计数权重时有mark标记已经记过的节点, 对每一个节点考虑其通向的未标记节点并加上其权重即可。

代码

```
1  #赵语涵2300012254
2  class Node():
3      def __init__(self,num,weight):
4          self.num = num
5          self.weight = weight
6          self.edge = []
7
8  n,m = map(int,input().split())
9  mark = [True]*n
10 maps = [Node(x,y) for (x,y) in zip(range(n),map(int,input().split()))]
11 for _ in range(m):
12     a,b = map(int,input().split())
13     maps[a].edge.append(maps[b])
14     maps[b].edge.append(maps[a])
15
16 def find(i,count):
17     mark[i] = False
18     for j in maps[i].edge:
19         if mark[j.num]:
20             count = find(j.num,count+j.weight)
21     return count
22
23 big = 0
24 for i in range(n):
25     big = max(big,find(i,maps[i].weight))
26 print(big)
```

代码运行截图

```
1  #赵语涵2300012254
2  class Node():
3      def __init__(self,num,weight):
4          self.num = num
5          self.weight = weight
6          self.edge = []
7
8  n,m = map(int,input().split())
9  mark = [True]*n
10 maps = [Node(x,y) for (x,y) in zip(range(n),map(int,input
11 for in range(m):
```

测试输入

提交结果

历史提交

提交时间

结果

时长(ms)

语言

2024-05-05 23:30:52

完美通过

0

Python

查看

03441: 4 Values whose Sum is 0

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

思路：因为要遍历各种组合，联想到以前做的八皇后就直接写了dfs然后超时了。有时候思维不能一下想得太复杂了...就用for循环进行组合就行。然后参考题解，在题目限定的空间时间范围内将数据分为a+b和c+d两组。存储a+b的值为key，数量为value，在c+d的组合中查找其相反数是否存在于keys中即可

代码

```
1  #赵语涵2300012254
2  n = int(input())
3  a,b,c,d = [0]*n,[0]*n,[0]*n,[0]*n
4  for i in range(n):
5      a[i],b[i],c[i],d[i] = map(int,input().split())
6
7  groups = {}
8  for i in range(n):
9      for j in range(n):
10         try:
11             groups[a[i]+b[j]]+=1
12         except:
13             groups[a[i]+b[j]] = 1
```

```

14
15 result = 0
16 for i in range(n):
17     for j in range(n):
18         if -(c[i]+d[j]) in groups.keys():
19             result += groups[-c[i]-d[j]]
20 print(result)

```

代码运行截图

#44875454提交状态

查看 提交 统计

状态: Accepted

源代码

```

#赵语涵2300012254
n = int(input())
a,b,c,d = [0]*n,[0]*n,[0]*n,[0]*n
for i in range(n):
    a[i],b[i],c[i],d[i] = map(int,input().split())

```

基本信息

#: 44875454
 题目: 03441
 提交人: 23n2300012254
 内存: 171600kB
 时间: 4965ms
 语言: Python3
 提交时间: 2024-05-05 21:22:56

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：通过class建立前缀树，其中设置了一个end参数，当某个节点为号码末尾时end设置为True。对于输入号码，每个数字指向下一个数字：child参数存储其后面的节点，当下一个数字已存在于子节点中时进入该子节点，否则向子节点列表添加新的子节点，最后一个子节点设置end为True。将所有电话号码按位数从小到大排，处理某个电话号码时如果遇到某个节点end=True，则说明有重叠，输出'NO'，否则输出'YES'

代码

```

1 #赵语涵2300012254
2 class Tree():
3     def __init__(self,value):
4         self.value = value
5         self.child = []
6         self.end = False
7
8 for _ in range(int(input())):
9     start = Tree(None)
10    t = start
11    answer = 'YES'
12    phones = []
13    for n in range(int(input())):
14        phones.append(list(input()))

```

```

15     for tele in sorted(phones):
16         for x in tele:
17             for y in t.child:
18                 if x == y.value:
19                     if y.end:
20                         answer = 'NO'
21                         t = y
22                         break
23             else:
24                 m = Tree(x)
25                 t.child.append(m)
26                 t = m
27         t.end = True
28         t = start
29     print(answer)

```

代码运行截图

#44876220提交状态

[查看](#) [提交](#) [统计](#)

状态: Accepted

源代码

```

#赵语涵2300012254
class Tree():
    def __init__(self,value):
        self.value = value
        self.child = []
        self.end = False

```

基本信息

#: 44876220
 题目: 04089
 提交人: 23n2300012254
 内存: 18968kB
 时间: 475ms
 语言: Python3
 提交时间: 2024-05-05 22:24:23

04082: 树的镜面映射

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

思路：学习了左儿子右兄弟这类树的转换方法，向左进入节点时层数增加，而到达叶节点进入兄弟节点时层数减少

代码

```

1     from collections import defaultdict
2     n=int(input())
3     nodes=[(x,int(i)) for x,i in input().split()]
4     output=defaultdict(list)
5     t=0
6     last=0
7     ma=0
8     for x,i in nodes:
9         #print(last,x,i)
10        if last == 1:
11            t-=1

```

```
12     else:
13         t+=1
14         if x != '$':
15             output[t].append(x)
16             ma=max(ma,t)
17         last=i
18 print(' '.join([' '.join(output[i][::-1])for i in range(1,ma+1)]))
```

代码运行截图

#45194666提交状态

[查看](#) [提交](#) [统计](#)

状态: **Accepted**

源代码

```
from collections import defaultdict
n=int(input())
nodes=[(x,int(i)) for x,i in input().split()]
output=defaultdict(list)
t=0
last=0
-- ^
```

基本信息

#: 45194666
题目: 04082
提交人: 23n2300012254
内存: 3636kB
时间: 20ms
语言: Python3
提交时间: 2024-06-04 10:18:36

2. 学习总结和收获

练习了以前计概用的一些dfs等算法以及树的用法，学习了二叉树转换的模版写法。