Assignment #8: 图论：概念、遍历，及 树算

Updated 1919 GMT+8 Apr 8, 2024

2024 spring, Complied by 武昱达 23工院

编程环境

操作系统：Windows 11

PyCharm 2023.1.4 (Professional Edition)

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：

规范标准的类实现。

代码

```
# 23工院 武昱达
class Vertex:
    def __init__(self,key):
        self.id=key
        self.connectedTo={}

    def addNeighbor(self,nbr,weight=0):
        self.connectedTo[nbr]=weight

    def getGrades(self):
        return len(self.connectedTo)

    def is_connected_to(self,other):
        return other in self.connectedTo

class Graph:
    def __init__(self):
        self.vertList={}
        self.numVertices=0

    def addVertex(self,key):
        self.numVertices=self.numVertices+1
        newVertex=Vertex(key)
        self.vertList[key]=newVertex
        return newVertex

    def addEdge(self,v1,v2,weight=0):
```

```
        if v1 not in self.vertList:
            nv=self.addVertex(v1)
        if v2 not in self.vertList:
            nv=self.addVertex(v2)
        self.vertList[v1].addNeighbor(self.vertList[v2],weight)
        self.vertList[v2].addNeighbor(self.vertList[v1],weight)

G=Graph()
n,m=map(int,input().split())
for i in range(n):
    G.addVertex(i)
for _ in range(m):
    v1,v2=map(int,input().split())
    G.addEdge(v1,v2)

matrix_1=[[0 for _ in range(n)] for _ in range(n)]
matrix_2=[[0 for _ in range(n)] for _ in range(n)]
matrix_3=[[0 for _ in range(n)] for _ in range(n)]

for i in range(n):
    matrix_1[i][i]=G.vertList[i].getGrades()

for i in range(n):
    for j in range(n):
        if G.vertList[i].is_connected_to(G.vertList[j]):
            matrix_2[i][j]=1
for i in range(n):
    for j in range(n):
        matrix_3[i][j]=matrix_1[i][j]-matrix_2[i][j]

for row in matrix_3:
    print(*row)
```

代码运行截图 （至少包含有"Accepted"）

#44617172提交状态

查看 提交 统计 提问

状态: Accepted

基本信息

#: 44617172
题目: 19943
提交人: 23n2300011119（武）
内存: 3772kB
时间: 28ms
语言: Python3
提交时间: 2024-04-12 17:19:25

源代码

```
class Vertex:
    def __init__(self,key):
        self.id=key
        self.connectedTo={}

    def addNeighbor(self,nbr,weight=0):
        self.connectedTo[nbr]=weight

    def getGrades(self):
        return len(self.connectedTo)

    def is_connected_to(self,other):
        return other in self.connectedTo

class Graph:
    def __init__(self):
        self.vertList={}
        self.numVertices=0

    def addVertex(self,key):
        self.numVertices=self.numVertices+1
        newVertex=Vertex(key)
        self.vertList[key]=newVertex
        return newVertex

    def addEdge(self,v1,v2,weight=0):
        if v1 not in self.vertList:
            nv=self.addVertex(v1)
        if v2 not in self.vertList:
            nv=self.addVertex(v2)
        self.vertList[v1].addNeighbor(self.vertList[v2],weight)
        self.vertList[v2].addNeighbor(self.vertList[v1],weight)

G=Graph()
n,m=map(int,input().split())
for i in range(n):
    G.addVertex(i)
for  in range(m):
```

18160: 最大连通域面积

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

思路:

代码

```
def dfs(matrix,x,y,visited):
    if (x<0 or x>=len(matrix) or y<0 or y>=len(matrix[0]) or matrix[x][y]!="W" or visited[x][y]):
        return 0
    visited[x][y],area=True,1
    directions = [(-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 1),(1, -1), (1, 0), (1, 1)]

    for dx, dy in directions:
        area += dfs(matrix, x + dx, y + dy, visited)
    return area

def max_adj_area(matrix):
    rows,cols,max_area=len(matrix),len(matrix[0]),0
    visited=[[False for _ in range(cols)] for _ in range(rows)]
    for row in range(rows):
        for col in range(cols):
            if matrix[row][col]=="W" and not visited[row][col]:
                area=dfs(matrix,row,col,visited)
```

```
                max_area=max(max_area,area)
    return max_area

for _ in range(T:=int(input())):
    N,M=map(int,input().split())
    matrix_1=[input() for _ in range(N)]
    print(max_adj_area(matrix_1))
```

代码运行截图 （至少包含有"Accepted"）

#44617237提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
def dfs(matrix,x,y,visited):
    if (x<0 or x>=len(matrix) or y<0 or y>=len(matrix[0]) or matrix[x][y
        return 0
    visited[x][y],area=True,1
    directions = [(-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 1),(1, -1),

    for dx, dy in directions:
        area += dfs(matrix, x + dx, y + dy, visited)
    return area

def max_adj_area(matrix):
    rows,cols,max_area=len(matrix),len(matrix[0]),0
    visited=[[False for _ in range(cols)] for _ in range(rows)]
    for row in range(rows):
        for col in range(cols):
            if matrix[row][col]=="W" and not visited[row][col]:
                area=dfs(matrix,row,col,visited)
                max_area=max(max_area,area)
    return max_area

for _ in range(T:=int(input())):
    N,M=map(int,input().split())
    matrix_1=[input() for _ in range(N)]
    print(max_adj_area(matrix_1))
```

基本信息

#: 44617237
题目: 18160
提交人: 23n2300011119 (武)
内存: 3728kB
时间: 118ms
语言: Python3
提交时间: 2024-04-12 17:25:15

sy383: 最大权值连通块

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

思路:

和上一题完全相同，但是用类实现。

代码

```
class Vertex:
    def __init__(self,key,weight):
        self.id=key
        self.weight=weight
        self.connectedTo={}
    def addNeighbor(self,nbr,weight=0):
        self.connectedTo[nbr]=weight

    def __str__(self):
        return '*'+str(self.id)

class Graph:
    def __init__(self):
        self.vertList={}
        self.numVertices=0
```

```
    def addVertex(self,key,weight):
        self.numVertices=self.numVertices+1
        newVertex=Vertex(key,weight)
        self.vertList[key]=newVertex
    def addEdge(self,v1,v2,weight=0):
        self.vertList[v1].addNeighbor(self.vertList[v2],weight)
        self.vertList[v2].addNeighbor(self.vertList[v1],weight)


def dfs(vert_id,visited):
    vert=G.vertList[vert_id]
    if visited[vert.id]:return 0
    weight,visited[vert.id]=vert.weight,True
    if not vert.connectedTo:
        return weight

    for son_vert in vert.connectedTo:
        son_id=son_vert.id
        weight+=dfs(son_id,visited)
    return weight

def MaxAdjWeights(graph):
    weights,visited = [],[False for _ in range(n)]
    for vert_id in graph.vertList:
        if not visited[vert_id]:
            weights.append(dfs(vert_id,visited))
    return max(weights)

G=Graph()
n,m=map(int,input().split())
vert_weight=list(map(int,input().split()))

for id,weight in enumerate(vert_weight):
    G.addVertex(id,weight)

for _ in range(m):
    v1,v2=map(int,input().split())
    G.addEdge(v1,v2)

print(MaxAdjWeights(G))
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

```
        weights,visited = [],[False for _ in range(n)]
        for vert_id in graph.vertList:
            if not visited[vert_id]:
                weights.append(dfs(vert_id,visited))
        return max(weights)
```

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

03441: 4 Values whose Sum is 0

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

思路:

直接4个笛卡尔积会爆，分成两组即可。

代码

```
#
n = int(input())
a = [0]*(n+1)
b = [0]*(n+1)
c = [0]*(n+1)
d = [0]*(n+1)

for i in range(n):
    a[i],b[i],c[i],d[i] = map(int, input().split())

dict1 = {}
for i in range(n):
    for j in range(n):
        if not a[i]+b[j] in dict1:
            dict1[a[i] + b[j]] = 0
        dict1[a[i] + b[j]] += 1

ans = 0
for i in range(n):
    for j in range(n):
        if -(c[i]+d[j]) in dict1:
            ans += dict1[-(c[i]+d[j])]

```

```
print(ans)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

#44622038提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
n = int(input())
a = [0]*(n+1)
b = [0]*(n+1)
c = [0]*(n+1)
d = [0]*(n+1)

for i in range(n):
    a[i],b[i],c[i],d[i] = map(int, input().split())

dict1 = {}
for i in range(n):
    for j in range(n):
        if not a[i]+b[j] in dict1:
            dict1[a[i] + b[j]] = 0
        dict1[a[i] + b[j]] += 1

ans = 0
for i in range(n):
    for j in range(n):
        if -(c[i]+d[j]) in dict1:
            ans += dict1[-(c[i]+d[j])]

print(ans)
```

基本信息

#: 44622038
题目: 03441
提交人: 23n2300011119 (武)
内存: 171656kB
时间: 5407ms
语言: Python3
提交时间: 2024-04-12 22:01:10

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

```
# 23工院 武昱达
for _ in range(t:=int(input())):
    flag,n=True,int(input())
    numbers=[input() for _ in range(n)]
    numbers.sort()
    for i in range(1,n):
        if numbers[i].startswith(numbers[i-1]):
            flag=False
            break
    print('YES' if flag else 'NO')
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

#44639974提交状态

状态: Accepted

源代码

```
for _ in range(t:=int(input())):
    flag,n=True,int(input())
    numbers=[input() for _ in range(n)]
    numbers.sort()
    for i in range(1,n):
        if numbers[i].startswith(numbers[i-1]):
            flag=False
            break
    print('YES' if flag else 'NO')
```


04082: 树的镜面映射

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

思路:

代码

```
from collections import deque
class GenericTreeNode:
    def __init__(self,val):
        self.val=val
        self.children=[]

def build_tree(tempList,index):
    node=GenericTreeNode(tempList[index][0])

    if tempList[index][1]=='0' and node.val!='$':
        index+=1
        child,index=build_tree(tempList,index)
        node.children.append(child)
        index+=1
        child,index=build_tree(tempList,index)
        node.children.append(child)

    return node,index

def print_tree(p):
    Q,S=deque(),deque()
    while p!=None:
        if p.val!='$':
            S.append(p)
        p=p.children[1] if len(p.children)>1 else None

    while S:
```

```
        Q.append(S.pop())
    while Q:
        p=Q.popleft()
        print(p.val,end=' ')

        if p.children:
            p=p.children[0]
            while p!=None:
                if p.val!="$":
                    S.append(p)
                p=p.children[1] if len(p.children)>1 else None

            while S:
                Q.append(S.pop())

n=int(input())
tempList=input().split()
root,_=build_tree(tempList,0)
print_tree(root)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

#44640397提交状态

查看 提交 统计 提问

状态：Accepted

源代码

```
from collections import deque
class GenericTreeNode:
    def __init__(self,val):
        self.val=val
        self.children=[]

def build_tree(tempList,index):
    node=GenericTreeNode(tempList[index][0])

    if tempList[index][1]=='0' and node.val!='$':
        index+=1
        child,index=build_tree(tempList,index)
        node.children.append(child)
        index+=1
        child,index=build_tree(tempList,index)
        node.children.append(child)

    return node,index

def print_tree(p):
    Q,S=deque(),deque()
    while p!=None:
        if p.val!='$':
            S.append(p)
        p=p.children[1] if len(p.children)>1 else None
```

基本信息

#: 44640397
题目: 04082
提交人: 23n2300011119 (武)
内存: 3720kB
时间: 27ms
语言: Python3
提交时间: 2024-04-13 21:22:32

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。

近两周赶上期中季，忙不过来了，每日选做落下了10道题，以后尽量补上。

树的镜面映射自己搓代码比较耗时，参考github上的题解完成。

期中季何！时！结！束！