

Assignment #8: 图论：概念、遍历，及 树算

Updated 1919 GMT+8 Apr 8, 2024

2024 spring, Compiled 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

```
1  # 23工院 武昱达
2  class Vertex:
3      def __init__(self, key):
4          self.id = key
5          self.connectedTo = {}
6
7      def addNeighbor(self, nbr, weight=0):
8          self.connectedTo[nbr] = weight
9
10     def getGrades(self):
11         return len(self.connectedTo)
12
13     def is_connected_to(self, other):
14         return other in self.connectedTo
15
16 class Graph:
17     def __init__(self):
18         self.vertList = {}
19         self.numVertices = 0
20
21     def addVertex(self, key):
22         self.numVertices = self.numVertices + 1
23         newVertex = Vertex(key)
24         self.vertList[key] = newVertex
25         return newVertex
26
27     def addEdge(self, v1, v2, weight=0):
```

```

28         if v1 not in self.vertList:
29             nv=self.addVertex(v1)
30         if v2 not in self.vertList:
31             nv=self.addVertex(v2)
32         self.vertList[v1].addNeighbor(self.vertList[v2],weight)
33         self.vertList[v2].addNeighbor(self.vertList[v1],weight)
34
35     G=Graph()
36     n,m=map(int,input().split())
37     for i in range(n):
38         G.addVertex(i)
39     for _ in range(m):
40         v1,v2=map(int,input().split())
41         G.addEdge(v1,v2)
42
43     matrix_1=[[0 for _ in range(n)] for _ in range(n)]
44     matrix_2=[[0 for _ in range(n)] for _ in range(n)]
45     matrix_3=[[0 for _ in range(n)] for _ in range(n)]
46
47     for i in range(n):
48         matrix_1[i][i]=G.vertList[i].getGrades()
49
50     for i in range(n):
51         for j in range(n):
52             if G.vertList[i].is_connected_to(G.vertList[j]):
53                 matrix_2[i][j]=1
54     for i in range(n):
55         for j in range(n):
56             matrix_3[i][j]=matrix_1[i][j]-matrix_2[i][j]
57
58     for row in matrix_3:
59         print(*row)

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

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 i in range(m):
```

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

18160: 最大连通域面积

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

思路:

代码

```
1 def dfs(matrix,x,y,visited):
2     if (x<0 or x>=len(matrix) or y<0 or y>=len(matrix[0]) or matrix[x]
3     [y]!="w" or visited[x][y]):
4         return 0
5     visited[x][y],area=True,1
6     directions = [(-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 1),(1, -1), (1,
7     0), (1, 1)]
8     for dx, dy in directions:
9         area += dfs(matrix, x + dx, y + dy, visited)
10    return area
11 def max_adj_area(matrix):
12    rows,cols,max_area=len(matrix),len(matrix[0]),0
13    visited=[[False for _ in range(cols)] for _ in range(rows)]
14    for row in range(rows):
15        for col in range(cols):
16            if matrix[row][col]=="w" and not visited[row][col]:
17                area=dfs(matrix,row,col,visited)
```

```

18         max_area=max(max_area,area)
19     return max_area
20
21 for _ in range(T:=int(input())):
22     N,M=map(int,input().split())
23     matrix_1=[input() for _ in range(N)]
24     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] in visited):
        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))

```

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

```

1 class Vertex:
2     def __init__(self,key,weight):
3         self.id=key
4         self.weight=weight
5         self.connectedTo={}
6     def addNeighbor(self,nbr,weight=0):
7         self.connectedTo[nbr]=weight
8
9     def __str__(self):
10        return '*' + str(self.id)
11
12 class Graph:
13     def __init__(self):
14         self.vertList={}
15         self.numVertices=0

```

```

16     def addVertex(self,key,weight):
17         self.numVertices=self.numVertices+1
18         newVertex=Vertex(key,weight)
19         self.vertList[key]=newVertex
20     def addEdge(self,v1,v2,weight=0):
21         self.vertList[v1].addNeighbor(self.vertList[v2],weight)
22         self.vertList[v2].addNeighbor(self.vertList[v1],weight)
23
24
25     def dfs(vert_id,visited):
26         vert=G.vertList[vert_id]
27         if visited[vert.id]:return 0
28         weight,visited[vert.id]=vert.weight,True
29         if not vert.connectedTo:
30             return weight
31
32         for son_vert in vert.connectedTo:
33             son_id=son_vert.id
34             weight+=dfs(son_id,visited)
35         return weight
36
37     def MaxAdjWeights(graph):
38         weights,visited = [],[False for _ in range(n)]
39         for vert_id in graph.vertList:
40             if not visited[vert_id]:
41                 weights.append(dfs(vert_id,visited))
42         return max(weights)
43
44     G=Graph()
45     n,m=map(int,input().split())
46     vert_weight=list(map(int,input().split()))
47
48     for id,weight in enumerate(vert_weight):
49         G.addVertex(id,weight)
50
51     for _ in range(m):
52         v1,v2=map(int,input().split())
53         G.addEdge(v1,v2)
54
55     print(MaxAdjWeights(G))

```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

```

38     ... weights, visited = [], [False for _ in range(n)]
39     ... for vert_id in graph.vertList:
40     ...     if not visited[vert_id]:
41     ...         weights.append(dfs(vert_id, visited))
42     ... 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

```

1  #
2  n = int(input())
3  a = [0]*(n+1)
4  b = [0]*(n+1)
5  c = [0]*(n+1)
6  d = [0]*(n+1)
7
8  for i in range(n):
9      a[i], b[i], c[i], d[i] = map(int, input().split())
10
11 dict1 = {}
12 for i in range(n):
13     for j in range(n):
14         if not a[i]+b[j] in dict1:
15             dict1[a[i] + b[j]] = 0
16             dict1[a[i] + b[j]] += 1
17
18 ans = 0
19 for i in range(n):
20     for j in range(n):
21         if -(c[i]+d[j]) in dict1:
22             ans += dict1[-(c[i]+d[j])]
23

```

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

```
1 # 23工院 武昱达
2 for _ in range(t:=int(input())):
3     flag,n=True,int(input())
4     numbers=[input() for _ in range(n)]
5     numbers.sort()
6     for i in range(1,n):
7         if numbers[i].startswith(numbers[i-1]):
8             flag=False
9             break
10    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')
```

©2002-2022 POJ 京ICP备20010980号-1

04082: 树的镜面映射

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

思路:

代码

```
1  from collections import deque
2  class GenericTreeNode:
3      def __init__(self,val):
4          self.val=val
5          self.children=[]
6
7  def build_tree(tempList,index):
8      node=GenericTreeNode(tempList[index][0])
9
10     if tempList[index][1]=='0' and node.val!='$':
11         index+=1
12         child,index=build_tree(tempList,index)
13         node.children.append(child)
14         index+=1
15         child,index=build_tree(tempList,index)
16         node.children.append(child)
17
18     return node,index
19
20 def print_tree(p):
21     Q,S=deque(),deque()
22     while p!=None:
23         if p.val!='$':
24             S.append(p)
25         p=p.children[1] if len(p.children)>1 else None
26
27     while S:
```



```

28         Q.append(S.pop())
29     while Q:
30         p=Q.popleft()
31         print(p.val,end=' ')
32
33         if p.children:
34             p=p.children[0]
35             while p!=None:
36                 if p.val!="$":
37                     S.append(p)
38                 p=p.children[1] if len(p.children)>1 else None
39
40         while S:
41             Q.append(S.pop())
42
43 n=int(input())
44 tempList=input().split()
45 root,_=build_tree(tempList,0)
46 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!