

Assignment #8: 图论：概念、遍历，及 树算

Updated 1919 GMT+8 Apr 8, 2024

2024 spring, Compiled by ==同学的姓名、院系==

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：macOS Ventura 13.4.1 (c)

Python编程环境：Spyder IDE 5.2.2, PyCharm 2023.1.4 (Professional Edition)

C/C++编程环境：Mac terminal vi (version 9.0.1424), g++/gcc (Apple clang version 14.0.3, clang-1403.0.22.14.1)

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：把两个类里的各种方法都用了一下

代码

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

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

    def getConnections(self):
```

```

        return self.connectedTo.keys()

    def getId(self):
        return self.id

    def getweight(self, nbr):
        return self.connectedTo[nbr]

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 getVertex(self, n):
        if n in self.vertList:
            return self.vertList[n]
        else:
            return None

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

    def getVertices(self):
        return self.vertList.keys()

    def __iter__(self):
        return iter(self.vertList.values())

def constructLaplacianMatrix(n, edges):
    graph = Graph()
    for i in range(n):
        graph.addVertex(i)

    for edge in edges:
        a, b = edge
        graph.addEdge(a, b)
        graph.addEdge(b, a)

    laplacianMatrix = []
    for vertex in graph:
        row = [0] * n
        row[vertex.getId()] = len(vertex.getConnections())
        for neighbor in vertex.getConnections():
            row[neighbor.getId()] = -1
        laplacianMatrix.append(row)

```

```

        return laplacianMatrix

n, m = map(int, input().split())
edges = []
for i in range(m):
    a, b = map(int, input().split())
    edges.append((a, b))
laplacianMatrix = constructLaplacianMatrix(n, edges)
for row in laplacianMatrix:
    print(' '.join(map(str, 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 getConnections(self):
        return self.connectedTo.keys()

    def getId(self):
        return self.id

    def getWeight(self, nbr):
        return self.connectedTo[nbr]

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

```

基本信息

#: 44668392
 题目: 19943
 提交人: 23n2300012140(zyt)
 内存: 3752kB
 时间: 25ms
 语言: Python3
 提交时间: 2024-04-15 21:43:09

18160: 最大连通域面积

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

思路: 用上学期提高班的做法处理

代码

```

a = [[-1, -1], [-1, 0], [-1, 1], [0, -1], [0, 1], [1, -1], [1, 0], [1, 1]]

b = 0
def lianjie(x, y):
    global b
    if juzhen[x][y] == ".":
        return
    else:
        b += 1
        juzhen[x][y] = '.'

```

```

        for i in range(8):
            lianjie(x+a[i][0],y+a[i][1])

cishu = int(input())

for _ in range(cishu):
    hang , lie = map(int,input().split())
    juzhen = [["." for _ in range(lie+2)] for _ in range(hang+2)]
    for w in range(1,hang+1):
        juzhen[w][1:-1] = input()
    daan = 0
    for s in range(1,hang+1):
        for t in range(1,lie+1):
            if juzhen[s][t] == 'W':
                b = 0
                lianjie(s,t)
                daan = max(b,daan)
    print(daan)

```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```

a = [[-1,-1],[-1,0],[-1,1],[0,-1],[0,1],[1,-1],[1,0],[1,1]]

b = 0
def lianjie(x,y):
    global b
    if juzhen[x][y] == ".":
        return
    else:
        b += 1
        juzhen[x][y] = '.'
        for i in range(8):
            lianjie(x+a[i][0],y+a[i][1])

cishu = int(input())

for _ in range(cishu):
    hang , lie = map(int,input().split())
    juzhen = [["." for _ in range(lie+2)] for _ in range(hang+2)]
    for w in range(1,hang+1):
        juzhen[w][1:-1] = input()
    daan = 0
    for s in range(1,hang+1):
        for t in range(1,lie+1):
            if juzhen[s][t] == 'W':
                b = 0

```

基本信息

#: 44678547
 题目: 18160
 提交人: 23n2300012140(zyt)
 内存: 3740kB
 时间: 99ms
 语言: Python3
 提交时间: 2024-04-16 21:45:24

sy383: 最大权值连通块

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

思路: 比较平和的dfs

代码

```
class Vertex:
```

```

def __init__(self, key, weight):
    self.key = key
    self.weight = weight
    self.nbrs = []

def addnbr(self, nbr):
    self.nbrs.append(nbr)

class Graph:
    def __init__(self):
        self.vertices = {}

    def addvertex(self, key, weight):
        cur = Vertex(key, weight)
        self.vertices[key] = cur
        return cur

    def addedge(self, k1, k2):
        self.vertices[k1].nbrs.append(self.vertices[k2])
        self.vertices[k2].nbrs.append(self.vertices[k1])

def dfs(vertex):
    ans = vertex.weight
    check[vertex.key] = False
    for k in vertex.nbrs:
        if check[k.key]:
            ans += dfs(k)
            check[k.key] = False
    return ans

n, m = map(int, input().split())
check = [True] * n
weights = list(map(int, input().split()))
p = Graph()
for i in range(n):
    p.addvertex(i, weights[i])
for j in range(m):
    k1, k2 = map(int, input().split())
    p.addedge(k1, k2)
ans = 0
for vertex in p.vertices.values():
    if check[vertex.key]:
        ans = max(ans, dfs(vertex))
print(ans)

```

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

代码书写 Python

```
13
14 def addvertex(self, key, weight):
15     cur = Vertex(key, weight)
16     self.vertexs[key] = cur
17     return cur
18
19 def addedge(self, k1, k2):
20     ...
```

测试输入 提交结果 历史提交

完美通过

查看题解

100% 数据通过测试

运行时长: 0 ms

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：纯用列表就会超时，看到答案的字典加列表感受到巧妙

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") ==

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

#: 44679173
 题目: 03441
 提交人: 23n2300012140(zyt)
 内存: 171728kB
 时间: 5612ms
 语言: Python3
 提交时间: 2024-04-16 22:30:21

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: 学习了题解的方法

代码

```

class TrieNode:
    def __init__(self):
        self.child={}

class Trie:
    def __init__(self):
        self.root = TrieNode()

    def insert(self, nums):
        curnode = self.root
        for x in nums:

```

```

        if x not in curnode.child:
            curnode.child[x] = TrieNode()
            curnode=curnode.child[x]

    def search(self, num):
        curnode = self.root
        for x in num:
            if x not in curnode.child:
                return 0
            curnode = curnode.child[x]
        return 1

t = int(input())
p = []
for _ in range(t):
    n = int(input())
    nums = []
    for _ in range(n):
        nums.append(str(input()))
    nums.sort(reverse=True)
    s = 0
    trie = Trie()
    for num in nums:
        s += trie.search(num)
        trie.insert(num)
    if s > 0:
        print('NO')
    else:
        print('YES')

```

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

状态: **Accepted**

源代码

```

class TrieNode:
    def __init__(self):
        self.child={}

class Trie:
    def __init__(self):
        self.root = TrieNode()

    def insert(self, nums):
        curnode = self.root
        for x in nums:
            if x not in curnode.child:
                curnode.child[x] = TrieNode()
            curnode=curnode.child[x]

    def search(self, num):
        curnode = self.root
        for x in num:
            if x not in curnode.child:
                return 0
            curnode = curnode.child[x]

```

基本信息

#: 44676670
 题目: 04089
 提交人: 23n2300012140(zyt)
 内存: 24668kB
 时间: 392ms
 语言: Python3
 提交时间: 2024-04-16 19:42:30

04082: 树的镜面映射

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

思路：感觉还是很有难度的，一点一点看题解

代码

```
from collections import deque

class TreeNode:
    def __init__(self, x):
        self.x = x
        self.children = []

def create_node():
    return TreeNode('')

def build_tree(tempList, index):
    node = create_node()
    node.x = tempList[index][0]
    if tempList[index][1] == '0':
        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 = deque()
    s = deque()

    while p is not None:
        if p.x != '$':
            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.x, end=' ')

        if p.children:
            p = p.children[0]
            while p is not None:
                if p.x != '$':
                    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") ==

状态: **Accepted**

源代码

```
from collections import deque

class TreeNode:
    def __init__(self, x):
        self.x = x
        self.children = []

def create_node():
    return TreeNode('')

def build_tree(tempList, index):
    node = create_node()
    node.x = tempList[index][0]
    if tempList[index][1] == '0':
        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):
```

基本信息

#: 44676785
题目: 04082
提交人: 23n2300012140(zyt)
内存: 3712kB
时间: 26ms
语言: Python3
提交时间: 2024-04-16 19:50:01

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。==

本周期中任务繁重, 幸好在上学期学习了搜索和图的相关知识。