

Assignment #8: 图论：概念、遍历，及 树算

Updated 1919 GMT+8 Apr 8, 2024

2024 spring, Compiled by 数学科学学院 王镜廷 2300010724

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：Windows11 专业版

Python编程环境：VSCode 1.86.2, with extension Python and python version 3.12.222.14.1)

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

用时：约15分钟

思路：定义Vertex类，Graph类并实现。

代码

```

class Vertex :
    def __init__ (self) :
        self.adj = set()
    def addedge(self, other) :
        self.adj.add(other)
class Graph :
    def __init__ (self, vertices, edges = []) :
        self.vertices = {}
        for item in vertices :
            self.vertices[item] = Vertex()
        for edge in edges :
            self.addedge(edge[0], edge[1])
    def addedge (self, u, v) :
        self.vertices[u].addedge(v)
        self.vertices[v].addedge(u)
    def getdeg (self, v) :
        return len(self.vertices[v].adj)
    def checkedge (self, u, v) :
        if v in self.vertices[u].adj :
            return 1
        return 0
n, m = map(int, input().split())
G = Graph(list(i for i in range(n)))
for i in range(m) :
    u, v = map(int, input().split())
    G.addedge(u, v)
print("\n".join(" ".join(str(-G.checkedge(i, j) if i != j else G.getdeg(i)) for j in range(n))

```



代码运行截图（至少包含有"Accepted"）

状态: **Accepted**

源代码

```
class Vertex :
    def __init__ (self) :
        self.adj = set()
    def addedge(self, other) :
        self.adj.add(other)
class Graph :
    def __init__ (self, vertices, edges = []) :
        self.vertices = {}
        for item in vertices :
            self.vertices[item] = Vertex()
        for edge in edges :
            self.addedge(edge[0], edge[1])
    def addedge (self, u, v) :
        self.vertices[u].addedge(v)
        self.vertices[v].addedge(u)
    def getdeg (self, v) :
        return len(self.vertices[v].adj)
    def checkedge (self, u, v) :
        if v in self.vertices[u].adj :
            return 1
```

基本信息

#: 44636973
题目: 19943
提交人: 23n2300010724
内存: 3680kB
时间: 25ms
语言: Python3
提交时间: 2024-04-13 19:22:05

18160: 最大连通域面积

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

用时: 约20分钟

思路: 使用并查集记录连通块

代码

```

class DisjointSet :
    def __init__(self, item) :
        self.rep = dict(zip(item, item))
        self.size = dict(zip(item, [1] * len(item)))
    def getrep(self, x) :
        if not x in self.rep :
            return None
        if self.rep[x] == x :
            return self.rep[x]
        self.rep[x] = self.getrep(self.rep[x])
        return self.rep[x]
    def merge(self, x, y) :
        if (not x in self.rep) or (not y in self.rep) :
            return
        u = self.getrep(x)
        v = self.getrep(y)
        if u == v :
            return
        if self.size[u] >= self.size[v] :
            self.rep[v] = u
            self.size[u] += self.size[v]
            return
        v, u = u, v
        self.rep[v] = u
        self.size[u] += self.size[v]
        return
    def getsizes(self, x) :
        if not x in self.rep :
            return 0
        return self.size[self.getrep(x)]

```

```

T = int(input())
for Case in range(T) :
    s = []
    n, m = map(int, input().split())
    for i in range(n) :
        s.append(input())
    U = []
    for i in range(n) :
        U.extend((i, j) for j in range(m))
    u = DisjointSet(U)
    steps = [(-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 1), (1, 0), (1, -1), (1, 1)]
    for step in steps :
        for i in range(n) :
            for j in range(m) :

```

```

        x = i + step[0]
        y = j + step[1]
        if 0 <= x and x < n and 0 <= y and y < m and s[i][j] == s[x][y] :
            u.merge((i, j), (x, y))

res = 0
for i in range(n) :
    for j in range(m) :
        if s[i][j] == "W" :
            res = max(res, u.getsize((i, j)))
print(res)

```

代码运行截图（至少包含有"Accepted"）

#44583720提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

class DisjointSet :
    def __init__(self, item) :
        self.rep = dict(zip(item, item))
        self.size = dict(zip(item, [1] * len(item)))
    def getrep(self, x) :
        if not x in self.rep :
            return None
        if self.rep[x] == x :
            return self.rep[x]
        self.rep[x] = self.getrep(self.rep[x])
        return self.rep[x]
    def merge(self, x, y) :
        if (not x in self.rep) or (not y in self.rep) :
            return
        u = self.getrep(x)
        v = self.getrep(y)
        if u == v :
            return
        if self.size[u] >= self.size[v] :
            self.rep[v] = u

```

基本信息

#: 44583720
 题目: 18160
 提交人: 23n2300010724
 内存: 4284kB
 时间: 1313ms
 语言: Python3
 提交时间: 2024-04-09 15:47:08

sy383: 最大权值连通块

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

用时: 约10分钟

思路: 用并查集记录连通块

代码

```

class DisjointSet :
    def __init__(self, item, size) :
        self.rep = dict(zip(item, item))
        self.size = dict(zip(item, size))
    def getrep(self, x) :
        if not x in self.rep :
            return None
        if self.rep[x] == x :
            return self.rep[x]
        self.rep[x] = self.getrep(self.rep[x])
        return self.rep[x]
    def merge(self, x, y) :
        if (not x in self.rep) or (not y in self.rep) :
            return
        u = self.getrep(x)
        v = self.getrep(y)
        if u == v :
            return
        if self.size[u] >= self.size[v] :
            self.rep[v] = u
            self.size[u] += self.size[v]
            return
        v, u = u, v
        self.rep[v] = u
        self.size[u] += self.size[v]
        return
    def getsize(self, x) :
        if not x in self.rep :
            return 0
        return self.size[self.getrep(x)]

n, m = map(int, input().split())
s = list(map(int, input().split()))
A = DisjointSet(list(i for i in range(n)), s)
for i in range(m) :
    u, v = map(int, input().split())
    A.merge(u, v)
print(max(A.getsize(u) for u in range(n)))

```

代码运行截图（AC代码截图，至少包含有"Accepted"）

完美通过

100% 数据通过测试

运行时长: 0 ms

语言: Python

```
1 class DisjointSet :
2     def __init__(self, item, size) :
3         self.rep = dict(zip(item, item))
4         self.size = dict(zip(item, size))
5     def getrep(self, x) :
6         if not x in self.rep :
7             return None
8         if self.rep[x] == x :
9             return self.rep[x]
10        self.rep[x] = self.getrep(self.rep[x])
11        return self.rep[x]
12    def merge(self, x, y) :
13        if (not x in self.rep) or (not y in self.rep) :
14            return
15        u = self.getrep(x)
16        v = self.getrep(y)
17        if u == v :
18            return
19        if self.size[u] < self.size[v] :
```

03441: 4 Values whose Sum is 0

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

用时: 约30分钟 (因为一开始不知道Counter怎么用)

思路: 记录A, B的和, C, D的和之后查找其中互为相反数的对, 注意使用字典会MLE, 在群里同学的提示下使用了Counter。

代码

```
from collections import Counter
from itertools import product
n = int(input())
A = []
B = []
C = []
D = []
for i in range(n) :
    u = list(map(int, input().split()))
    A.append(u[0])
    B.append(u[1])
    C.append(u[2])
    D.append(u[3])
A = sorted(A)
B = sorted(B)
C = sorted(C)
D = sorted(D)
res = 0
AB = Counter(map(lambda x : x[0] + x[1], product(A, B)))
for c in C :
    for d in D :
        res += AB.get(-c - d, 0)
print(res)
```

代码运行截图（AC代码截图，至少包含有"Accepted"）

#44584347提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
from collections import Counter
from itertools import product
n = int(input())
A = []
B = []
C = []
D = []
for i in range(n) :
    u = list(map(int, input().split()))
    A.append(u[0])
    B.append(u[1])
    C.append(u[2])
    D.append(u[3])
A = sorted(A)
B = sorted(B)
C = sorted(C)
D = sorted(D)
res = 0
AB = Counter(map(lambda x : x[0] + x[1], product(A, B)))
for c in C :
    for d in D :
        res += AB.get(-c - d, 0)
print(res)
```

基本信息

#: 44584347
题目: 03441
提交人: 23n2300010724
内存: 171724kB
时间: 3730ms
语言: Python3
提交时间: 2024-04-09 16:29:10

04089: 电话号码

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

Trie 数据结构可能需要自学下。

用时：约70分钟（中间有个bug找了好久才找到）

思路：实现Trie树，相容性条件即是说这一新词既不是所有已知词的前缀，又不以已知词作为其前缀。

代码

```

class Trienode :
    def __init__ (self) :
        self.children = {}
        self.is_end_of_word = False
class Trie :
    def __init__ (self) :
        self.root = Trienode()
    def insert(self, word) :
        cur = self.root
        for char in word :
            if not (char in cur.children) :
                cur.children[char] = Trienode()
            cur = cur.children[char]
        cur.is_end_of_word = True

    def search(self, word) :
        cur = self.root
        for char in word :
            if not (char in cur.children) :
                return False
            cur = cur.children[char]
        return cur.is_end_of_word

    def compatbl(self, word) :
        cur = self.root
        for char in word :
            #print(word, list(item for item in cur.children), cur.is_end_of_word, end = " ")
            if cur.is_end_of_word :
                #print("2")
                return False

            # DO NOT EXCHANGE THE POSITION OF TWO CONDITIONS!
            if not (char in cur.children) :
                #print("1")
                return True
            #print("3")
            cur = cur.children[char]
        return False

T = int(input())
for _ in range(T) :
    n = int(input())
    flag = True
    trie = Trie()
    for _ in range(n) :
        s = input()
        if not trie.compatbl(s) :

```

```
flag = False
trie.insert(s)
print("YES" if flag else "NO")
```

代码运行截图（AC代码截图，至少包含有"Accepted"）

#44635584提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
class TrieNode :
    def __init__ (self) :
        self.children = {}
        self.is_end_of_word = False
class Trie :
    def __init__ (self) :
        self.root = TrieNode()
    def insert(self, word) :
        cur = self.root
        for char in word :
            if not (char in cur.children) :
                cur.children[char] = TrieNode()
            cur = cur.children[char]
        cur.is_end_of_word = True

    def search(self, word) :
        cur = self.root
        for char in word :
            if not (char in cur.children) :
                return False
            cur = cur.children[char]
        return cur.is_end_of_word

    def compatbl(self, word) :
        cur = self.root
```

基本信息

#: 44635584
题目: 04089
提交人: 23n2300010724
内存: 24280kB
时间: 410ms
语言: Python3
提交时间: 2024-04-13 18:09:27

04082: 树的镜面映射

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

用时: 约40分钟

思路: 直接模拟题意

代码

```

from collections import deque

class BinTreeNode :
    def __init__ (self, val, left = None, right = None) :
        self.left = left
        self.right = right
        self.val = val
    def toGenTree (self) :
        #print("togentree", self.Traversal("pre"), "building:", self.val)
        c = self.val
        cur = self.left
        res = []
        while cur != None and cur.val != "$":
            res.append(cur.toGenTree())
            cur = cur.right
        #print(self.val, "Done")
        return GenTreeNode(c, res)
    def Traversal (self, mode) :
        u = "" if self.left == None else self.left.Traversal(mode)
        v = "" if self.right == None else self.right.Traversal(mode)
        if mode == "pre" :
            return self.val + " " + u + " " + v
        if mode == "mid" :
            return u + " " + self.val + " " + v
        if mode == "post" :
            return u + " " + v + " " + self.val
def _getnode (s) :
    if s[0][1] == "1" :
        return BinTreeNode(s[0][0]), 1
    else :
        c = s[0][0]
        u, lu = _getnode(s[1:])
        v, lv = _getnode(s[(lu + 1):])
        return BinTreeNode(c, u, v), lu + lv + 1
def getnode (s) :
    return _getnode(s)[0]
class BinTree :
    def __init__ (self, s) :
        self.root = getnode(s)
    def Traversal (self, mode) :
        return self.root.Traversal(mode)
    def toGenTree (self) :
        return self.root.toGenTree()
class GenTreeNode :
    def __init__ (self, val, children = []) :
        self.children = children

```

```

        self.val = val
class GenTree :
    def __init__ (self, bintree) :
        self.root = bintree.toGenTree()
    def _mirror (self, pos) :
        pos.children.reverse()
        for child in pos.children :
            self._mirror(child)
    def mirror (self) :
        self._mirror(self.root)
    def level_traversal(self) :
        q = deque()
        q.append(self.root)
        res = []
        while len(q) > 0 :
            res.append(q[0].val)
            q.extend(q[0].children)
            q.popleft()
        return " ".join(item for item in res)

_ = int(input())
B = BinTree(input().split())
#print(B.Traversal("pre"))
#print(B.Traversal("mid"))
G = GenTree(B)
G.mirror()
print(G.level_traversal())

```

代码运行截图（AC代码截图，至少包含有"Accepted"）

#44636491提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

from collections import deque

class BinTreeNode :
    def __init__ (self, val, left = None, right = None) :
        self.left = left
        self.right = right
        self.val = val
    def toGenTree (self) :
        #print("togentree", self.Traversal("pre"), "building:", self.val)
        c = self.val
        cur = self.left
        res = []
        while cur != None and cur.val != "$":
            res.append(cur.toGenTree())
            cur = cur.right
        #print(self.val, "Done")
        return GenTreeNode(c, res)
    def Traversal (self, mode) :
        u = "" if self.left == None else self.left.Traversal(mode)
        v = "" if self.right == None else self.right.Traversal(mode)
        if mode == "pre" :

```

基本信息

#: 44636491
 题目: 04082
 提交人: 23n2300010724
 内存: 3784kB
 时间: 28ms
 语言: Python3
 提交时间: 2024-04-13 19:10:15

2. 学习总结和收获

这周看到群里同学使用Counter，试了一下，感觉很好用，又发现python里面可以写lambda函数，这个功能很有意思，但还不太会用，问了一下GPT，介绍的Y Combinator技巧很巧妙。debug水平感觉还有待提高，对于一般树的递归处理熟练程度仍需加强。最近忙着期中考试，每日一练暂时还没跟上。