Assignment #8: 图论：概念、遍历，及 树算

Updated 1919 GMT+8 Apr 8, 2024

2024 spring, Complied by 王业成 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

==（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：

代码

```
# class Vertex:
    def __init__(self,key):
        self.id=key
        self.connect={}
    def addneighbour(self,neighbour,weight=0):
        self.connect[neighbour]=weight
    def getid(self):
        return self.id
    def getconnect(self):
```

```
        return self.connect.keys()
class Graph:
    def __init__(self):
        self.vertlist={}
        self.numvert=0
    def addvertex(self,key):
        newvertex=Vertex(key)
        self.vertlist[key]=newvertex
        self.numvert+=1
        return newvertex
    def addedge(self,f,t):
        if f not in self.vertlist:
            a=self.addvertex(f)
        if t not in self.vertlist:
            a=self.addvertex(t)
        self.vertlist[f].addneighbour(self.vertlist[t],0)
    def __iter__(self):
        return iter(self.vertlist.values())
def construct(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)
    l=[]
    for vertex in graph:
        row=[0]*n
        row[vertex.getid()]=len(vertex.getconnect())
        for neighbour in vertex.getconnect():
            row[neighbour.getid()]=-1
        l.append(row)
    return l
n,m=map(int,input().split())
edges=[]
for i in range(m):
    a,b=map(int,input().split())
    edges.append((a,b))
l=construct(n,edges)
for row in l:
    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.connect={}
    def addneighbour(self,neighbour,weight=0):
        self.connect[neighbour]=weight
    def getid(self):
        return self.id
    def getconnect(self):
        return self.connect.keys()
class Graph:
    def __init__(self):
        self.vertlist={}
        self.numvert=0
    def addvertex(self,key):
        newvertex=Vertex(key)
        self.vertlist[key]=newvertex
        self.numvert+=1
        return newvertex
    def addedge(self,f,t):
        if f not in self.vertlist:
            a=self.addvertex(f)
        if t not in self.vertlist:
            a=self.addvertex(t)
        self.vertlist[f].addneighbour(self.vertlist[t],0)
    def __iter__(self):
        return iter(self.vertlist.values())
def construct(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)
```

基本信息

#:	44668777
题目:	19943
提交人:	wangyecheng
内存:	3732kB
时间:	29ms
语言:	Python3
提交时间:	2024-04-15 22:09:38

18160: 最大连通域面积

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

思路:

代码

```
# dire=[[-1,-1],[-1,0],[-1,1],[0,-1],[0,1],[1,-1],[1,0],[1,1]]
area=0
def dfs(x,y):
    global area
    if lst[x][y]==".":
        return
    lst[x][y]="."
    area+=1
    for i in range(len(dire)):
        dfs(x+dire[i][0],y+dire[i][1])
for _ in range(int(input())):
    n,m=map(int,input().split())
    lst=[["." for _ in range(m+2)] for _ in range(n+2)]
    for i in range(1,n+1):
        lst[i][1:-1]=input()
    sum=0
    for i in range(1,n+1):
```

```
        for j in range(1,m+1):
            if lst[i][j]=="W":
                area=0
                dfs(i,j)
                sum=max(sum,area)
    print(sum)
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
dire=[[-1,-1],[-1,0],[-1,1],[0,-1],[0,1],[1,-1],[1,0],[1,1]]
area=0
def dfs(x,y):
    global area
    if lst[x][y]==".":
        return
    lst[x][y]="."
    area+=1
    for i in range(len(dire)):
        dfs(x+dire[i][0],y+dire[i][1])
for _ in range(int(input())):
    n,m=map(int,input().split())
    lst=[["." for _ in range(m+2)] for _ in range(n+2)]
    for i in range(1,n+1):
        lst[i][1:-1]=input()
    sum=0
    for i in range(1,n+1):
        for j in range(1,m+1):
            if lst[i][j]=="W":
                area=0
                dfs(i,j)
                sum=max(sum,area)
    print(sum)
```

基本信息

#:	44669053
题目:	18160
提交人:	wangyecheng
内存:	3680kB
时间:	102ms
语言:	Python3
提交时间:	2024-04-15 22:40:02

sy383: 最大权值连通块

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

思路：采用并查集实现

代码

```
# n, m = map(int, input().split())
lst= list(map(int, input().split()))
p = [i for i in range(n)]
def find(x):
    if p[x] == x:
        return x
    else:
        y=find(p[x])
        p[x]=y
        return y
def union(x, y):
    a=find(x)
    b=find(y)
```

```
    if a!=b:
        p[a]=b
        lst[b]+=lst[a]
for _ in range(m):
    x, y = map(int, input().split())
    union(x, y)
print(max(lst))
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

完美通过 查看题解

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:

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())
dict={}
for i in range(n):
    for j in range(n):
        if not a[i]+b[j] in dict:
            dict[a[i] + b[j]] = 0
        dict[a[i] + b[j]] += 1
sum= 0
for i in range(n):
    for j in range(n):
        if -(c[i]+d[j]) in dict:
            sum+=dict[-(c[i]+d[j])]
print(sum)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态: Accepted

基本信息

#: 44669347
题目: 03441
提交人: wangyecheng
内存: 171688kB
时间: 5290ms
语言: Python3
提交时间: 2024-04-15 23:08:17

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())
dict={}
for i in range(n):
    for j in range(n):
        if not a[i]+b[j] in dict:
            dict[a[i] + b[j]] = 0
        dict[a[i] + b[j]] += 1
sum= 0
for i in range(n):
    for j in range(n):
        if -(c[i]+d[j]) in dict:
            sum+=dict[-(c[i]+d[j])]
print(sum)
```

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

```
# class Trienode:
    def __init__(self):
        self.child={}
class Trie:
    def __init__(self):
        self.root=Trienode()
    def insert(self,nums):
        node=self.root
        for x in nums:
            if x not in node.child:
                node.child[x]=Trienode()
            node=node.child[x]
    def search(self,num):
        node=self.root
        for x in num:
            if x not in node.child:
                return 0
            node=node.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 i in nums:
        s+=trie.search(i)
        trie.insert(i)
    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):
        node=self.root
        for x in nums:
            if x not in node.child:
                node.child[x]=Trienode()
            node=node.child[x]
    def search(self,num):
        node=self.root
        for x in num:
            if x not in node.child:
                return 0
            node=node.child[x]
        return 1
t = int(input())
p = []
for _ in range(t):
    n = int(input())
    nums = []
```

基本信息

#:	44669572
题目:	04089
提交人:	wangyecheng
内存:	24896kB
时间:	385ms
语言:	Python3
提交时间:	2024-04-15 23:30:08

04082: 树的镜面映射

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

思路:

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):
    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)
```

基本信息

#：44669611
题目：04082
提交人：wangyecheng
内存：3720kB
时间：26ms
语言：Python3
提交时间：2024-04-15 23:33:05

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。==

最近期中比较忙，花在数算上的时间不多，这次作业也基本都是借鉴题解先看一编理解题解的思路在写的，也算有些许收获。后续等期中过后五一期间一定要狠狠花时间补数算