

# Assignment #B: 图论和树算

Updated 1709 GMT+8 Apr 28, 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

### 28170: 算鹰

dfs, <http://cs101.openjudge.cn/practice/28170/>

思路:

代码

```
d=[(0,1),(0,-1),(1,0),(-1,0)]
visited=[[0 for i in range(10)]for j in range(10)]
board=[]
ans=0
def dfs(x,y):
    global visited
    visited[x][y]=1
    for dx,dy in d:
        nx=x+dx
        ny=y+dy
        if 0<=nx<=9 and 0<=ny<=9 and visited[nx][ny]==0 and board[x][y]=='.':
```

```

        dfs(nx,ny)

for i in range(10):
    board.append(input())
for i in range(10):
    for j in range(10):
        #print(visited[i][j])
        if board[i][j]=='.' and visited[i][j]==0:
            dfs(i,j)
            ans+=1
print(ans)

```

代码运行截图

#44886506提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

d=[(0,1),(0,-1),(1,0),(-1,0)]
visited=[[0 for i in range(10)] for j in range(10)]
board=[]
ans=0
def dfs(x,y):
    global visited
    visited[x][y]=1
    for dx,dy in d:

```

基本信息

#: 44886506  
 题目: 28170  
 提交人: 23n2300012265  
 内存: 3652kB  
 时间: 19ms  
 语言: Python3  
 提交时间: 2024-05-07 11:13:06

## 02754: 八皇后

dfs, <http://cs101.openjudge.cn/practice/02754/>

思路:

代码

```

ans=[]
def queen(n,a):
    global ans
    b=a.copy()
    for i in range(1,9):
        b.append(i)
        if check(n,b):
            if n==8:
                ans.append(b)
                return
            else:
                queen(n+1,b)
        b.pop(-1)
def check(n,a):

```

```

        for i in range(n-1):
            if a[i]==a[-1] or a[i]-a[-1]==i+1-n or a[i]-a[-1]==n-i-1:
                return 0
        return 1
    queen(1, [])
    n=int(input())
    for i in range(n):
        print(*ans[int(input())-1],sep='')

```

代码运行截图

#44886719提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

ans=[]
def queen(n,a):
    global ans
    b=a.copy()
    for i in range(1,9):
        b.append(i)
        if check(n,b):
            if n==9:

```

基本信息

#: 44886719  
 题目: 02754  
 提交人: 23n2300012265  
 内存: 3636kB  
 时间: 37ms  
 语言: Python3  
 提交时间: 2024-05-07 11:36:34

## 03151: Pots

bfs, <http://cs101.openjudge.cn/practice/03151/>

思路:

代码

```

p=['','DROP(1)','DROP(2)','FILL(1)','FILL(2)','POUR(1,2)','POUR(2,1)']
class node:
    def __init__(self,x,y):
        self.a=x
        self.b=y
        self.father=None
        self.path=None
def ans(a,root):
    if root.father==None:
        print(a)
        return
    ans(a+1,root.father)
    print(p[root.path])
from collections import deque
a,b,c=map(int,input().split())
queue=deque()
in_queue=[[0 for i in range(b+1)]for j in range(a+1)]
in_queue[0][0]=1
x=node(0,0)
queue.append(x)

```

```

root=None
while queue:
    now=queue.popleft()
    x=now.a
    y=now.b
    if x==c or y==c:
        root=now
        break
    d=[(0,y,1),(x,0,2),(a,y,3),(x,b,4)]
    if x>=b-y:
        d.append((x-b+y,b,5))
    else:
        d.append((0,x+y,5))
    if y>=a-x:
        d.append((a,x+y-a,6))
    else:
        d.append((x+y,0,6))
    for i,j,k in d:

        if in_queue[i][j]==0:
            in_queue[i][j]=1
            new=node(i,j)
            new.father=now
            new.path=k
            queue.append(new)
if root==None:
    print('impossible')
else:
    ans(0,root)

```

代码运行截图

#44886885提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

p=['','DROP(1)','DROP(2)','FILL(1)','FILL(2)','POUR(1,2)','POUR(2,1)']
class node:
    def __init__(self,x,y):
        self.a=x
        self.b=y
        self.father=None
        self.path=None

```

基本信息

#: 44886885  
 题目: 03151  
 提交人: 23n2300012265  
 内存: 3724kB  
 时间: 22ms  
 语言: Python3  
 提交时间: 2024-05-07 12:12:02

## 05907: 二叉树的操作

<http://cs101.openjudge.cn/practice/05907/>

思路:

代码

```

class treeNode:
    def __init__(self,x):
        self.key=x
        self.left=None
        self.right=None
        self.father=None

def swap(x,y):
    a=x.father
    al=a.left
    ar=a.right
    b=y.father
    bl=b.left
    br=b.right
    x.father=b
    y.father=a
    if al==x:
        a.left=y
    else:
        a.right=y
    if bl==y:
        b.left=x
    else:
        b.right=x

def left(a):
    if a.left!=None:
        left(a.left)
    else:
        print(a.key)

t=int(input())
for _ in range(t):
    tree={}
    n,m=map(int,input().split())
    for i in range(n):
        node=treeNode(i)
        tree[i]=node
    for i in range(n):
        x,y,z=map(int,input().split())
        if y!=-1:
            tree[x].left=tree[y]
            tree[y].father=tree[x]
        if z!=-1:
            tree[x].right=tree[z]
            tree[z].father=tree[x]
    for i in range(m):
        s=[int(i) for i in input().split()]
        if s[0]==1:
            swap(tree[s[1]],tree[s[2]])
        if s[0]==2:
            left(tree[s[1]])

```

状态: Accepted

源代码

```
class treenode:
    def __init__(self,x):
        self.key=x
        self.left=None
        self.right=None
        self.father=None

def swab(x,v):
```

基本信息

#: 44886937  
题目: 05907  
提交人: 23n2300012265  
内存: 3936kB  
时间: 79ms  
语言: Python3  
提交时间: 2024-05-07 12:44:21

## 18250: 冰阔落 I

Disjoint set, <http://cs101.openjudge.cn/practice/18250/>

思路:

代码

```
class UnionFindSet():
    def __init__(self,data_list):
        self.father_dict={}
        self.size_dict={}
        for node in data_list:
            self.father_dict[node]=node
            self.size_dict[node]=1
    def find(self,node):
        father=self.father_dict[node]
        if father!=node:
            if father!=self.father_dict[father]:
                self.size_dict[father]-=1
            father=self.find(father)
        self.father_dict[node]=father
        return father
    def is_same_set(self,node_a,node_b):
        return self.find(node_a)==self.find(node_b)
    def union(self,node_a,node_b):
        a_head=self.find(node_a)
        b_head=self.find(node_b)
        self.father_dict[b_head]=a_head
        self.size_dict[a_head]+=self.size_dict[b_head]
    def ans(self,data_list):
        a=[]
        for i in data_list:
            if self.father_dict[i]==i:
                a.append(i)
        print(len(a))
```

```

        print(*a,sep=' ')

while 1:
    try:
        n,m=map(int,input().split())
        kuolo=UnionFindSet([i for i in range(1,n+1)])
        for i in range(m):
            a,b=map(int,input().split())
            if kuolo.is_same_set(a,b):
                print("Yes")
            else:
                print("No")
                kuolo.union(a,b)
        kuolo.ans([i for i in range(1,n+1)])
    except:
        break

```

代码运行截图

#44892345提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

class UnionFindSet():
    def __init__(self,data_list):
        self.father_dict={}
        self.size_dict={}
        for node in data_list:
            self.father_dict[node]=node
            self.size_dict[node]=1
    def find(self,node):

```

基本信息

#: 44892345  
 题目: 18250  
 提交人: 23n2300012265  
 内存: 9392kB  
 时间: 477ms  
 语言: Python3  
 提交时间: 2024-05-07 22:28:11

## 05443: 兔子与樱花

<http://cs101.openjudge.cn/practice/05443/>

思路:

代码

```
#
```

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

## 2. 学习总结和收获

---

bfs和dfs越来越熟练了，阔落那道题又练习了一遍并查集，还没来得及写兔子和樱花，先把作业交了，后面一定补上