Assignment #B: 图论和树算

Updated 1709 GMT+8 Apr 28, 2024

2024 spring, Complied by ==祁轩宇、经济学院==

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：Windows 11, version 23H2

Python编程环境：VSCode 1.87.1

C/C++编程环境：

1. 题目

28170: 算鹰

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

思路：

代码

```
from collections import deque

MAX_DIRECTIONS = 4
dx = [1,0,0,-1]
dy = [0,1,-1,0]

def is_valid_move(x, y):
    return 0 <= x < n and 0 <= y < m and maze[x][y] == '.' and not in_queue[x][y]

def bfs(start_x, start_y):
    queue = deque()
    if not in_queue[start_x][start_y]:
        queue.append((start_x, start_y))
        in_queue[start_x][start_y] = True
```

```
        while queue:
            x, y = queue.popleft()
            for i in range(MAX_DIRECTIONS):
                next_x = x + dx[i]
                next_y = y + dy[i]
                if is_valid_move(next_x, next_y):
                    in_queue[next_x][next_y] = True
                    queue.append((next_x, next_y))

def size(x,y):
    if not in_queue[x][y]:
        before=after=0
        for i in range(len(in_queue)):
            before+=sum(in_queue[i])
        bfs(x,y)
        for i in range(len(in_queue)):
            after+=sum(in_queue[i])
        return after-before
    else:
        return 0

n=m=10
maze = [input() for i in range(10)]
in_queue = [[False] * m for i in range(n)]
count=[0]*m*n

for i in range(n):
    for j in range(m):
        if maze[i][j]=='-':
            in_queue[i][j]=True

for i in range(n):
    for j in range(m):
        count[i*m+j]=(size(i,j)!=0)
print(sum(count))
```

代码运行截图 ==（至少包含有"Accepted"）==

#44891415提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
from collections import deque

MAX_DIRECTIONS = 4
dx = [1,0,0,-1]
dy = [0,1,-1,0]

def is_valid_move(x, y):
    return 0 <= x < n and 0 <= y < m and maze[x][y] == '.' and not in_qu

def bfs(start_x, start_y):
    queue = deque()
    if not in_queue[start_x][start_y]:
        queue.append((start_x, start_y))
        in_queue[start_x][start_y] = True
        while queue:
            x, y = queue.popleft()
            for i in range(MAX_DIRECTIONS):
                next_x = x + dx[i]
                next_y = y + dy[i]
                if is_valid_move(next_x, next_y):
                    in_queue[next_x][next_y] = True
                    queue.append((next_x, next_y))

def size(x,y):
    if not in_queue[x][y]:
        before=after=0
        for i in range(len(in_queue)):
            before+=sum(in_queue[i])
        bfs(x,y)
        for i in range(len(in_queue)):
            after+=sum(in_queue[i])
        return after-before
    else:
        return 0

n=m=10
maze = [input() for i in range(10)]
```

基本信息

#: 44891415
题目: 28170
提交人: 2100015440
内存: 3740kB
时间: 24ms
语言: Python3
提交时间: 2024-05-07 20:55:00

02754: 八皇后

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

思路：

代码

```
#
```

代码运行截图 ==（至少包含有"Accepted"）==

03151: Pots

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

思路：

代码

```
from collections import deque

A,B,C=map(int,input().split())

def bfs(start_x, start_y):
    MAX_DIRECTIONS = 6
    action=['FILL(1)','FILL(2)','DROP(1)','DROP(2)','POUR(1,2)','POUR(2,1)']
    class_queue=[[[]]*(B+1) for _ in range(A+1)]
    queue = deque()
    if not in_queue[start_x][start_y]:
        queue.append((start_x, start_y))
        in_queue[start_x][start_y] = True
        while queue:
            x, y = queue.popleft()
            for i in range(MAX_DIRECTIONS):
                dx = [A,x,0,x,max(0,x+y-B),min(x+y,A)]
                dy = [y,B,y,0,min(y+x,B),max(0,x+y-A)]
                next_x = dx[i]
                next_y = dy[i]
                if not in_queue[next_x][next_y]:
                    in_queue[next_x][next_y] = True
                    queue.append((next_x, next_y))
                    class_queue[next_x][next_y]=class_queue[x][y].copy() # 注意要
浅复制
                    class_queue[next_x][next_y].append(action[i])
                    if (next_x==C) | (next_y==C):
                        return [str(len(class_queue[next_x]
[next_y]))]+class_queue[next_x][next_y]
        return ['impossible']

in_queue = [[False] * (B+1) for i in range(A+1)]
print('\n'.join(bfs(0,0)))
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

#44892043提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```
from collections import deque

A,B,C=map(int,input().split())

def bfs(start_x, start_y):
    MAX_DIRECTIONS = 6
    action=['FILL(1)','FILL(2)','DROP(1)','DROP(2)','POUR(1,2)','POUR(2,1)']
    class_queue=[[[]]*(B+1) for _ in range(A+1)]
    queue = deque()
    if not in_queue[start_x][start_y]:
        queue.append((start_x, start_y))
        in_queue[start_x][start_y] = True
        while queue:
            x, y = queue.popleft()
            for i in range(MAX_DIRECTIONS):
                dx = [A,x,0,x,max(0,x+y-B),min(x+y,A)]
                dy = [y,B,y,0,min(y+x,B),max(0,x+y-A)]
                next_x = dx[i]
                next_y = dy[i]
                if not in_queue[next_x][next_y]:
                    in_queue[next_x][next_y] = True
                    queue.append((next_x, next_y))
                    class_queue[next_x][next_y]=class_queue[x][y].copy()
                    class_queue[next_x][next_y].append(action[i])
                    if (next_x==C) | (next_y==C):
                        return [str(len(class_queue[next_x][next_y]))]+
        return ['impossible']

in_queue = [[False] * (B+1) for i in range(A+1)]
print('\n'.join(bfs(0,0)))
```

基本信息

#:	44892043
题目:	03151
提交人:	2100015440
内存:	3712kB
时间:	20ms
语言:	Python3
提交时间:	2024-05-07 21:48:01

05907: 二叉树的操作

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

思路：

代码

```
def find_leftmost_node(son, u):
    while son[u][0] != -1:
        u = son[u][0]
    return u

def main():
    t = int(input())
    for _ in range(t):
        n, m = map(int, input().split())

        son = [-1] * (n + 1)  # 存储每个节点的子节点
        parent = {}  # 存储每个节点的父节点和方向，{节点: (父节点, 方向)}

        for _ in range(n):
            i, u, v = map(int, input().split())
```

```
            son[i] = [u, v]
            parent[u] = (i, 0)  # 左子节点
            parent[v] = (i, 1)  # 右子节点

        for _ in range(m):
            s = input().split()
            if s[0] == "1":
                u, v = map(int, s[1:])
                fu, diru = parent[u]
                fv, dirv = parent[v]
                son[fu][diru] = v
                son[fv][dirv] = u
                parent[v] = (fu, diru)
                parent[u] = (fv, dirv)
            elif s[0] == "2":
                u = int(s[1])
                root = find_leftmost_node(son, u)
                print(root)

if __name__ == "__main__":
    main()
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

#44892888提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
def find_leftmost_node(son, u):
    while son[u][0] != -1:
        u = son[u][0]
    return u

def main():
    t = int(input())
    for _ in range(t):
        n, m = map(int, input().split())

        son = [-1] * (n + 1)  # 存储每个节点的子节点
        parent = {}  # 存储每个节点的父节点和方向，{节点: (父节点, 方向)}

        for _ in range(n):
            i, u, v = map(int, input().split())
            son[i] = [u, v]
            parent[u] = (i, 0)  # 左子节点
            parent[v] = (i, 1)  # 右子节点

        for _ in range(m):
            s = input().split()
            if s[0] == "1":
                u, v = map(int, s[1:])
                fu, diru = parent[u]
                fv, dirv = parent[v]
                son[fu][diru] = v
                son[fv][dirv] = u
                parent[v] = (fu, diru)
                parent[u] = (fv, dirv)
            elif s[0] == "2":
                u = int(s[1])
                root = find_leftmost_node(son, u)
                print(root)

if __name__ == "__main__":
    main()
```

基本信息

#:	44892888
题目:	05907
提交人:	2100015440
内存:	3700kB
时间:	66ms
语言:	Python3
提交时间:	2024-05-07 23:35:58

18250: 冰阔落 I

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

思路：

代码

```
def find(x):
    if parent[x] != x: # 如果不是根结点，继续循环
        parent[x] = find(parent[x])
    return parent[x]

def union(x, y):
    parent[find(y)] = find(x)

while True:
    try:
        n,m=map(int,input().split())
        parent = list(range(n + 1)) # parent[i] == i，则说明元素i是该集合的根结点
        for _ in range(m):
            x,y=map(int, input().split())
            if find(x)==find(y):
                print('Yes')
            else:
                print('No')
                union(x,y)
        classes = set(find(x) for x in range(1, n + 1))
        classes = sorted(classes)
        print(len(classes))
        print(*classes)
    except:
        break
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

#44892740提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
def find(x):
    if parent[x] != x: # 如果不是根结点，继续循环
        parent[x] = find(parent[x])
    return parent[x]

def union(x, y):
    parent[find(y)] = find(x)

while True:
    try:
        n,m=map(int,input().split())
        parent = list(range(n + 1))      # parent[i] == i, 则说明元素i是该集
        for _ in range(m):
            x,y=map(int, input().split())
            if find(x)==find(y):
                print('Yes')
            else:
                print('No')
                union(x,y)
        classes = set(find(x) for x in range(1, n + 1))
        classes = sorted(classes)
        print(len(classes))
        print(*classes)
    except:
        break
```

基本信息

#: 44892740
题目: 18250
提交人: 2100015440
内存: 5364kB
时间: 371ms
语言: Python3
提交时间: 2024-05-07 23:19:56

©2002-2022 POJ 京ICP备20010980号-1 English 帮助 关于

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

==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。==

作业有一定难度，但有不少之前类似做过的题目。比如18250: 冰阔落 I和宗教信仰比较接近，03151: Pots和其他bfs的题目做法相似，可以套用之前的模板完成。