Assignment #B: 图论和树算

Updated 1709 GMT+8 Apr 28, 2024

2024 spring, Complied 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

```
#
def dfs(x,y):
    global b
    if(a[x][y]=='-' or b[x][y]==1):
        return
    b[x][y]=1
    if(x>0):
        dfs(x-1,y)
    if(x<9):
        dfs(x+1,y)
    if(y>0):
```

```
        dfs(x,y-1)
    if(y<9):
        dfs(x,y+1)
a=[]
for i in range(10):
    a.append(input())
ans=0
b=[10*[0] for i in range(10)]
for i in range(10):
    for j in range(10):
        if(a[i][j]=='.' and b[i][j]==0):
            ans+=1
            dfs(i,j)
print(ans)
```

代码运行截图 ==（至少包含有"Accepted"）==

02754: 八皇后

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

思路:

代码

```
#
def solve(n):
    s = []
    sol = []
```

```
    s.append((0, [-1] * n))

    while s:
        r, q = s.pop()

        if r == n:
            sol.append(q.copy())
        else:
            for c in range(n):
                if valid(r, c, q):
                    nq = q.copy()
                    nq[r] = c
                    s.append((r + 1, nq))

    return sol


def valid(r, c, q):
    for i in range(r):
        if q[i] == c or abs(r - i) == abs(c - q[i]):
            return False
    return True


def get_string(b):
    sols = solve(8)
    if b > len(sols):
        return None
    b = len(sols) + 1 - b

    return ''.join(str(col + 1) for col in sols[b - 1])


test_cases = int(input())
for _ in range(test_cases):
    b = int(input())
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
def solve(n):
    s = []
    sol = []

    s.append((0, [-1] * n))

    while s:
        r, q = s.pop()

        if r == n:
            sol.append(q.copy())
        else:
            for c in range(n):
                if valid(r, c, q):
                    nq = q.copy()
                    nq[r] = c
                    s.append((r + 1, nq))

    return sol


def valid(r, c, q):
    for i in range(r):
        if q[i] == c or abs(r - i) == abs(c - q[i]):
            return False
    return True


def get_string(b):
    sols = solve(8)
    if b > len(sols):
        return None
    b = len(sols) + 1 - b

    return ''.join(str(col + 1) for col in sols[b - 1])


test_cases = int(input())
for _ in range(test_cases):
    b = int(input())
    print(get_string(b))
```

03151: Pots

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

思路:

代码

```
#
def bfs(A, B, C):
    s = (0, 0)
    v = set()
    v.add(s)
```

```
    q = [(s, [])]

    while q:
        (a, b), c_a = q.pop(0)

        if a == C or b == C:
            return c_a

        n_s = [(A, b), (a, B), (0, b), (a, 0), (min(a + b, A), max(0, a + b -
A)), (max(0, a + b - B), min(a + b, B))]

        for i in n_s:
            if i not in v:
                v.add(i)
                n_a = c_a + [g_a(a, b, i)]
                q.append((i, n_a))

    return ["impossible"]


def g_a(a, b, n_s):
    if n_s == (A, b):
        return "FILL(1)"
    elif n_s == (a, B):
        return "FILL(2)"
    elif n_s == (0, b):
        return "DROP(1)"
    elif n_s == (a, 0):
        return "DROP(2)"
    elif n_s == (min(a + b, A), max(0, a + b - A)):
        return "POUR(2,1)"
    else:
        return "POUR(1,2)"


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

if s == ["impossible"]:
    print(s[0])
else:
    print(len(s))
    for i in s:
        print(i)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态: Accepted

基本信

提

提交

源代码

```
def bfs(A, B, C):
    s = (0, 0)
    v = set()
    v.add(s)
    q = [(s, [])]

    while q:
        (a, b), c_a = q.pop(0)

        if a == C or b == C:
            return c_a

        n_s = [(A, b), (a, B), (0, b), (a, 0), (min(a + b, A), max(0, a

        for i in n_s:
            if i not in v:
                v.add(i)
                n_a = c_a + [g_a(a, b, i)]
                q.append((i, n_a))

    return ["impossible"]


def g_a(a, b, n_s):
    if n_s == (A, b):
        return "FILL(1)"
    elif n_s == (a, B):
        return "FILL(2)"
    elif n_s == (0, b):
        return "DROP(1)"
    elif n_s == (a, 0):
        return "DROP(2)"
    elif n_s == (min(a + b, A), max(0, a + b - A)):
        return "POUR(2,1)"
    else:
        return "POUR(1,2)"


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

if s == ["impossible"]:
    print(s[0])
else:
    print(len(s))
    for i in s:
        print(i)
```

05907: 二叉树的操作

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

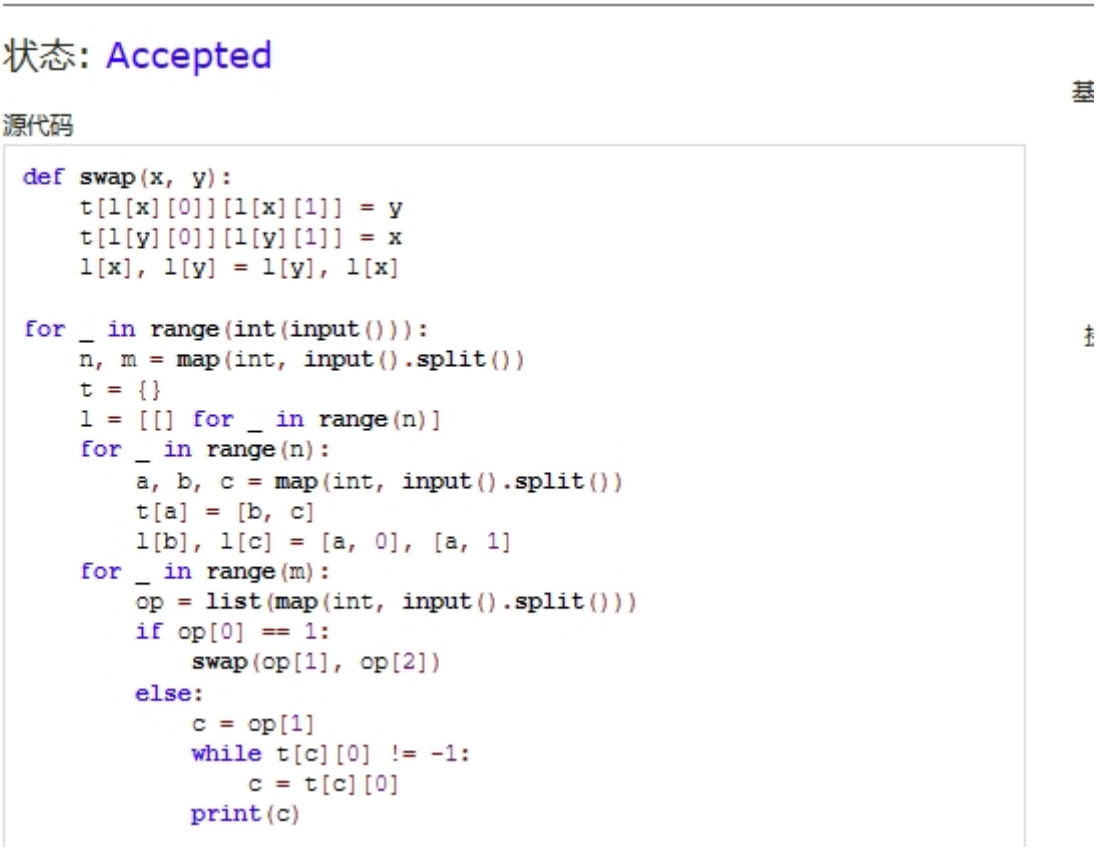
思路:

代码

```
#
def swap(x, y):
    t[l[x][0]][l[x][1]] = y
    t[l[y][0]][l[y][1]] = x
    l[x], l[y] = l[y], l[x]

for _ in range(int(input())):
    n, m = map(int, input().split())
    t = {}
    l = [[] for _ in range(n)]
    for _ in range(n):
        a, b, c = map(int, input().split())
        t[a] = [b, c]
        l[b], l[c] = [a, 0], [a, 1]
    for _ in range(m):
        op = list(map(int, input().split()))
        if op[0] == 1:
            swap(op[1], op[2])
        else:
            c = op[1]
            while t[c][0] != -1:
                c = t[c][0]
            print(c)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



18250: 冰阔落 I

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

思路：

代码

```
#
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

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。==

这周目前只做了四道题，因为又要考试啦...奇怪的期中季

这次的题目总体来说都是比较模版的，比较好做。