Assignment #B: 图论和树算

Updated 1709 GMT+8 Apr 28, 2024

2024 spring, Complied by 周添 物理学院

1. 题目

28170: 算鹰

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

代码

```
def count_eagles(board):
    def dfs(x, y):
        if x < 0 or x >= 10 or y < 0 or y >= 10 or board[x][y] != '.':
            return
        board[x][y] = '-'
        dfs(x + 1, y)
        dfs(x - 1, y)
        dfs(x, y + 1)
        dfs(x, y - 1)

    eagles = 0
    for i in range(10):
        for j in range(10):
            if board[i][j] == '.':
                dfs(i, j)
                eagles += 1
    return eagles


board = [list(input()) for _ in range(10)]

print(count_eagles(board))
```

状态: Accepted

源代码

```
def count_eagles(board):
    def dfs(x, y):
        if x < 0 or x >= 10 or y < 0 or y >= 10 or board[x][y] != '.':
            return
        board[x][y] = '-'
        dfs(x + 1, y)
        dfs(x - 1, y)
        dfs(x, y + 1)
        dfs(x, y - 1)

    eagles = 0
    for i in range(10):
        for j in range(10):
            if board[i][j] == '.':
                dfs(i, j)
                eagles += 1
    return eagles


board = [list(input()) for _ in range(10)]

print(count_eagles(board))
```

基本信息

#:	44831541
题目:	28170
提交人:	23n2300011538
内存:	3640kB
时间:	20ms
语言:	Python3
提交时间:	2024-04-29 17:43:3

02754: 八皇后

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

代码

```
def eq(a, b, c, n=0):
    if n == 8:
        ans.append(''.join(str(i) for i in a))
    else:
        for i in range(1, 9):
            if i not in a and (n+i) not in c and (n-i) not in b:
                a.append(i)
                b.append(n-i)
                c.append(n+i)
                eq(a, b, c, n+1)
                a.pop()
                b.pop()
                c.pop()


ans = []
eq([], [], [])

num = int(input())
for j in range(num):
    m = int(input())
    print(ans[m-1])
```

状态: Accepted

源代码

```
def eq(a, b, c, n=0):
    if n == 8:
        ans.append(''.join(str(i) for i in a))
    else:
        for i in range(1, 9):
            if i not in a and (n+i) not in c and (n-i) not in b:
                a.append(i)
                b.append(n-i)
                c.append(n+i)
                eq(a, b, c, n+1)
                a.pop()
                b.pop()
                c.pop()


ans = []
eq([], [], [])

num = int(input())
for j in range(num):
    m = int(input())
    print(ans[m-1])
```

基本信息

#:	43367227
题目:	02754
提交人:	23n2300011538
内存:	3652kB
时间:	28ms
语言:	Python3
提交时间:	2023-12-26 00:34:41

03151: Pots

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

代码

```
from collections import deque


def bfs(a, b, c):
    q = deque([(0, 0, [])])
    vis = set()
    vis.add((0, 0))

    while q:
        cur_a, cur_b, steps = q.popleft()

        if cur_a == c or cur_b == c:
            return steps

        alt = [
            (a, cur_b, steps + ['FILL(1)']),
            (cur_a, b, steps + ['FILL(2)']),
            (0, cur_b, steps + ['DROP(1)']),
            (cur_a, 0, steps + ['DROP(2)']),
            (max(0, cur_a - (b - cur_b)), min(b, cur_b + cur_a), steps +
['POUR(1,2)']),
            (min(a, cur_a + cur_b), max(0, cur_b - (a - cur_a)), steps +
['POUR(2,1)']),
        ]

        for new_a, new_b, new_steps in alt:
            if (new_a, new_b) not in vis:
                vis.add((new_a, new_b))
```

```
                q.append((new_a, new_b, new_steps))

    return "impossible"


a, b, c = map(int, input().split())
res = bfs(a, b, c)

if res == "impossible":
    print(res)
else:
    print(len(res))
    for step in res:
        print(step)
```

#44884448提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
from collections import deque


def bfs(a, b, c):
    q = deque([(0, 0, [])])
    vis = set()
    vis.add((0, 0))

    while q:
        cur_a, cur_b, steps = q.popleft()

        if cur_a == c or cur_b == c:
            return steps
```

基本信息

#: 44884448
题目: 03151
提交人: 23n2300011538
内存: 3684kB
时间: 23ms
语言: Python3
提交时间: 2024-05-06 23:54:38

05907: 二叉树的操作

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

代码

```
def swap_nodes(x, y):
    tree[loc[x][0]][loc[x][1]] = y
    tree[loc[y][0]][loc[y][1]] = x
    loc[x], loc[y] = loc[y], loc[x]


for _ in range(int(input())):
    n, m = map(int, input().split())
    tree = {}
    loc = [[] for _ in range(n)]

    for _ in range(n):
        a, b, c = map(int, input().split())
        tree[a] = [b, c]
        loc[b], loc[c] = [a, 0], [a, 1]
```

```
    for _ in range(m):
        op = list(map(int, input().split()))
        if op[0] == 1:
            swap_nodes(op[1], op[2])
        else:
            cur = op[1]
            while tree[cur][0] != -1:
                cur = tree[cur][0]
            print(cur)
```

状态: Accepted

源代码

```
def swap_nodes(x, y):
    tree[loc[x][0]][loc[x][1]] = y
    tree[loc[y][0]][loc[y][1]] = x
    loc[x], loc[y] = loc[y], loc[x]

for _ in range(int(input())):
    n, m = map(int, input().split())
    tree = {}
    loc = [[] for _ in range(n)]

    for _ in range(n):
        a, b, c = map(int, input().split())
        tree[a] = [b, c]
        loc[b], loc[c] = [a, 0], [a, 1]

    for _ in range(m):
        op = list(map(int, input().split()))
        if op[0] == 1:
            swap_nodes(op[1], op[2])
        else:
            cur = op[1]
            while tree[cur][0] != -1:
```

基本信息

#:	44884469
题目:	05907
提交人:	23n2300011538
内存:	3704kB
时间:	75ms
语言:	Python3
提交时间:	2024-05-06 23:58:46

18250: 冰阔落 I

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

代码

```
class ChiNingMeng:
    def __init__(self, n):
        self.p = list(range(n))

    def find(self, x):
        if self.p[x] != x:
            self.p[x] = self.find(self.p[x])
        return self.p[x]

    def union(self, x, y):
        rootx = self.find(x)
```

```
        rooty = self.find(y)
        self.p[rooty] = self.p[rootx]

while True:
    try:
        n, m = map(int, input().split())
        c = ChiNingMeng(n)

        for _ in range(m):
            x, y = map(int, input().split())
            x, y = x - 1, y - 1
            if c.find(x) == c.find(y):
                print('Yes')
            else:
                print('No')
                c.union(x, y)

        a = set([c.find(x) + 1 for x in range(n)])
        print(len(a))
        print(*sorted(a))

    except EOFError:
        break
```

#44884500提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
class ChiNingMeng:
    def __init__(self, n):
        self.p = list(range(n))

    def find(self, x):
        if self.p[x] != x:
            self.p[x] = self.find(self.p[x])
        return self.p[x]

    def union(self, x, y):
        rootx = self.find(x)
        rooty = self.find(y)
        self.p[rooty] = self.p[rootx]

while True:
    try:
        n, m = map(int, input().split())
        c = ChiNingMeng(n)

        for _ in range(m):
            x, y = map(int, input().split())
```

基本信息

#: 44884500
题目: 18250
提交人: 23n2300011538
内存: 5740kB
时间: 466ms
语言: Python3
提交时间: 2024-05-07 00:07:55

05443: 兔子与樱花

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

代码

```
from heapq import heappop, heappush
from collections import defaultdict
```

```python
def dijkstra(start, end):
    heap = [(0, start, [start])]
    vis = set()

    while heap:
        (cost, u, path) = heappop(heap)
        if u in vis:
            continue
        vis.add(u)
        if u == end:
            return (cost, path)
        for v in graph[u]:
            if v not in vis:
                heappush(heap, (cost + graph[u][v], v, path + [v]))

_ = [input() for _ in range(int(input()))]
graph = defaultdict(dict)

for _ in range(int(input())):
    u, v, w = input().split()
    graph[u][v] = graph[v][u] = int(w)

r = int(input())

for _ in range(r):
    start, end = input().split()
    cost, path = dijkstra(start, end)
    for i in range(len(path) - 1):
        print(f'{path[i]}->({graph[path[i]][path[i+1]]})->', end='')
    print(path[-1])
```

状态: Accepted

源代码

```python
from heapq import heappop, heappush
from collections import defaultdict

def dijkstra(start, end):
    heap = [(0, start, [start])]
    vis = set()

    while heap:
        (cost, u, path) = heappop(heap)
        if u in vis:
            continue
        vis.add(u)
        if u == end:
            return (cost, path)
        for v in graph[u]:
            if v not in vis:
                heappush(heap, (cost + graph[u][v], v, path + [v]))

_ = [input() for _ in range(int(input()))]
graph = defaultdict(dict)

for _ in range(int(input())):
    u, v, w = input().split()
    graph[u][v] = graph[v][u] = int(w)

r = int(input())
```

基本信息

#:	44884509
题目:	05443
提交人:	23n2300011538
内存:	3600kB
时间:	22ms
语言:	Python3
提交时间:	2024-05-07 00:12:14

2. 学习总结和收获

上学期学的八皇后差点忘了咋写

这周周末参加的比赛要结题presentation了，事情太多了，一遍没写对的代码我就搬了jzx同学的（求放过），但是他写得确实条理清晰，比我写的可读性强很多，值得学习

yysy真的很难平衡各种事情的时间分配，事情真的太多了，又都觉得好重要。。。事情多了就有点没心气做了，我上的体育课老师说没心气的时候再高强度运动容易受伤，但是我现在不得不一件一件把事情搞定。希望结局是好的