Assignment #B: 图论和树算

Updated 1709 GMT+8 Apr 28, 2024

2024 spring, Complied by 夏天明 元培学院

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

操作系统：Windows 10 | 22H2

Python编程环境：Spyder IDE 5.4.3 | Python 3.11.4 64-bit

1. 题目

28170: 算鹰

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

思路：简单的dfs

代码

```
def dfs(x, y):
    for dx, dy in direc:
        if 0<=x+dx<10 and 0<=y+dy<10 and graph[x+dx][y+dy] == '.':
            graph[x+dx][y+dy] = '-'
            dfs(x+dx, y+dy)

graph = [list(input()) for i in range(10)]
direc = [(1,0), (0,1), (-1,0), (0,-1)]
ans = 0
for i in range(10):
    for j in range(10):
        if graph[i][j] == '.':
```

```
            ans += 1
            dfs(i, j)
print(ans)
```

代码运行截图

#44837725提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
def dfs(x, y):
    for dx, dy in direc:
        if 0<=x+dx<10 and 0<=y+dy<10 and graph[x+dx][y+dy] == '.':
            graph[x+dx][y+dy] = '-'
            dfs(x+dx, y+dy)

graph = [list(input()) for i in range(10)]
direc = [(1,0), (0,1), (-1,0), (0,-1)]
ans = 0
for i in range(10):
    for j in range(10):
        if graph[i][j] == '.':
            ans += 1
            dfs(i, j)
print(ans)
```

基本信息

#: 44837725
题目: 28170
提交人: 23n2300017735(夏天明BrightSummer)
内存: 3624kB
时间: 21ms
语言: Python3
提交时间: 2024-04-30 20:22:54

English 帮助 关

02754: 八皇后

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

思路：dfs+回溯

代码

```
def dfs(row):
    if row == 8:
        ans.append(res[:])
        return
    for j in range(8):
        allow = True
        for i in range(row):
            if mat[i][j] \
               or (j-(row-i) >= 0 and mat[i][j-(row-i)]) \
               or (j+(row-i) < 8 and mat[i][j+(row-i)]):
                allow = False
                break
        if allow:
            no_solution = False
```

```
            res.append(j+1)
            mat[row][j] = 1
            dfs(row+1)
            res.pop()
            mat[row][j] = 0

mat = [[0]*8 for i in range(8)]
res = []
ans = []
dfs(0)
for o in range(int(input())):
    print(*ans[int(input())-1], sep='')
```

代码运行截图

#42810257提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
def dfs(row):
    if row == 8:
        ans.append(res[:])
        return
    for j in range(8):
        allow = True
        for i in range(row):
            if mat[i][j] \
               or (j-(row-i) >= 0 and mat[i][j-(row-i)]) \
               or (j+(row-i) < 8 and mat[i][j+(row-i)]):
                allow = False
                break
        if allow:
            no_solution = False
            res.append(j+1)
            mat[row][j] = 1
            dfs(row+1)
            res.pop()
            mat[row][j] = 0

mat = [[0]*8 for i in range(8)]
res = []
ans = []
dfs(0)
for o in range(int(input())):
    print(*ans[int(input())-1], sep='')
```

基本信息

#: 42810257
题目: 02754
提交人: 23n2300017735(夏天明 BrightSummer)
内存: 3640kB
时间: 46ms
语言: Python3
提交时间: 2023-11-28 17:58:15

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

03151: Pots

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

思路：直接bfs

代码

```python
from collections import deque

*full, C = map(int, input().split())
q = deque([((0,0), [])])
visited = set([(0,0)])
def bfs():
    while q:
        bottle, s = q.popleft()
        if C in bottle:
            print(len(s))
            for op in s:
                print(op)
            return
        for i in [0,1]:
            bo = list(bottle)
            bo[i] = full[i]
            bo = tuple(bo)
            if bo not in visited:
                q.append((bo, s + [f"FILL({i+1})"]))
                visited.add(bo)
        for i in [0,1]:
            bo = list(bottle)
            bo[i] = 0
            bo = tuple(bo)
            if bo not in visited:
                q.append((bo, s + [f"DROP({i+1})"]))
                visited.add(bo)
        for i in [0,1]:
            bo = list(bottle)
            bo[1-i] += bo[i]
            bo[i] = 0
            if bo[1-i] > full[1-i]:
                bo[i] = bo[1-i] - full[1-i]
                bo[1-i] = full[1-i]
            bo = tuple(bo)
            if bo not in visited:
                q.append((bo, s + [f"POUR({i+1},{2-i})"]))
                visited.add(bo)
    print("impossible")
bfs()
```

代码运行截图

#44838572提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
from collections import deque

*full, C = map(int, input().split())
q = deque([((0,0), [])])
visited = set([(0,0)])
def bfs():
    while q:
        bottle, s = q.popleft()
        if C in bottle:
            print(len(s))
            for op in s:
                print(op)
            return
        for i in [0,1]:
            bo = list(bottle)
            bo[i] = full[i]
            bo = tuple(bo)
            if bo not in visited:
                q.append((bo, s + [f"FILL({i+1})"]))
```

基本信息

#: 44838572
题目: 03151
提交人: 23n2300017735(夏天明BrightSummer)
内存: 3708kB
时间: 21ms
语言: Python3
提交时间: 2024-04-30 23:05:44

05907: 二叉树的操作

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

思路：建树，注意交换节点的时候搞清楚哪个变量在引用什么，否则容易出现死循环之类的

代码

```
class Node:
    def __init__(self, name):
        self.name = name
        self.child = [None, None]
        self.parent = None

    def findLef(self):
        curr = self
        while (lef := nodes[curr.child[0]]):
            curr = lef
        return curr.name

for o in range(int(input())):
    n, m = map(int, input().split())
    nodes = [Node(i) for i in range(n)] + [None]
    for o in range(n):
        x, *idx = map(int, input().split())
        nodes[x].child = idx
        for i in [0,1]:
            if idx[i] != -1:
                nodes[idx[i]].parent = (nodes[x],i)
```

```
    for o in range(m):
        token, *idx = map(int, input().split())
        if token == 1:
            p = [nodes[i].parent for i in idx]
            for i in [0,1]:
                p[i][0].child[p[i][1]] = idx[1-i]
                nodes[idx[i]].parent = p[1-i]
        else:
            print(nodes[idx[0]].findLef())
```

代码运行截图

状态: Accepted

源代码

```
class Node:
    def __init__(self, name):
        self.name = name
        self.child = [None, None]
        self.parent = None

    def findLef(self):
        curr = self
        while (lef := nodes[curr.child[0]]):
            curr = lef
        return curr.name

for o in range(int(input())):
    n, m = map(int, input().split())
    nodes = [Node(i) for i in range(n)] + [None]
    for o in range(n):
        x, *idx = map(int, input().split())
        nodes[x].child = idx
        for i in [0,1]:
            if idx[i] != -1:
                nodes[idx[i]].parent = (nodes[x],i)
    for o in range(m):
        token, *idx = map(int, input().split())
        if token == 1:
            p = [nodes[i].parent for i in idx]
            for i in [0,1]:
```

基本信息

#: 43776812
题目: 05907
提交人: 23n2300017735(夏天明 BrightSummer)
内存: 3688kB
时间: 82ms
语言: Python3
提交时间: 2024-01-29 20:10:59

18250: 冰阔落 I

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

思路：直接使用并查集实现

代码

```
class Dsu:
    def __init__(self, size):
        self.pa = list(range(size))
```

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

    def union(self, x, y):
        self.pa[self.find(x)] = self.find(y)

while True:
    try:
        n, m = map(int, input().split())
    except EOFError:
        break
    dsu = Dsu(n)
    for o in range(m):
        x, y = map(int, input().split())
        print('Yes' if dsu.find(x-1) == dsu.find(y-1) else 'No')
        dsu.union(y-1, x-1)
    ans = [i+1 for i in range(n) if dsu.pa[i] == i]
    print(len(ans))
    print(*ans)
```

代码运行截图

05443: 兔子与樱花

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

思路：dijkstra

代码

```
from copy import deepcopy
from heapq import heappop, heappush

class Place:
    def __init__(self):
        self.name = None
        self.nex = {}
        self.pre = None
        self.dist = float('inf')
        self.unknown = True

P = int(input())
graph = {input():Place() for i in range(P)}
for name, p in graph.items():
    p.name = name
Q = int(input())
for i in range(Q):
    begin, end, w = input().split()
    graph[begin].nex[end] = graph[end].nex[begin] = int(w)
R = int(input())
for o in range(R):
    begin, end = input().split()
    g = deepcopy(graph)
    q = [(0, begin)]
    while q:
        d, loc = heappop(q)
        if loc == end:
            break
        if not g[loc].unknown:
            continue
        g[loc].unknown = False
        for new, w in g[loc].nex.items():
            if g[new].unknown and d+w < g[new].dist:
                g[new].dist = d+w
                g[new].pre = loc
                heappush(q, (d+w, new))
    path = [end]
    while loc := g[path[-1]].pre:
        path.append(loc)
    ans = []
    while path:
        loc = path.pop()
        ans.append(loc)
        if path:
            ans.append(f"({g[loc].nex[path[-1]]})")
    print(*ans, sep='->')
```

代码运行截图

#43854181提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
from copy import deepcopy
from heapq import heappop, heappush

class Place:
    def __init__(self):
        self.name = None
        self.nex = {}
        self.pre = None
        self.dist = float('inf')
        self.unknown = True

P = int(input())
graph = {input():Place() for i in range(P)}
for name, p in graph.items():
    p.name = name
Q = int(input())
for i in range(Q):
    begin, end, w = input().split()
    graph[begin].nex[end] = graph[end].nex[begin] = int(w)
```

基本信息

#: 43854181
题目: 05443
提交人: 23n2300017735(夏天明 BrightSummer)
内存: 4056kB
时间: 27ms
语言: Python3
提交时间: 2024-02-04 14:11:43

2. 学习总结和收获

本次作业是上几周的图、树、并查集的内容的复习，较为基础