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==

赵云天 生命科学学院

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：win10

Python编程环境：Spyder IDE 5.2.2

C/C++编程环境：

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):
    graph[x][y] = '-'
    for dx,dy in [(-1,0),(1,0),(0,1),(0,-1)]:
        if 0<=x+dx<10 and 0<=y+dy<10:
            if graph[x+dx][y+dy] == '.':
                dfs(x+dx,y+dy)

ans = 0
graph = []
for _ in range(10):
    graph.append(list(input()))
for i in range(10):
    for j in range(10):
        if graph[i][j] == '.':
```

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

代码运行截图 ==（至少包含有"Accepted"）==

#44878929提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
def dfs(x,y):
    graph[x][y] = '-'
    for dx,dy in [(-1,0),(1,0),(0,1),(0,-1)]:
        if 0<=x+dx<10 and 0<=y+dy<10:
            if graph[x+dx][y+dy] == '.':
                dfs(x+dx,y+dy)

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

基本信息

#: 44878929
题目: 28170
提交人: 23n2300012140(zyt)
内存: 3660kB
时间: 21ms
语言: Python3
提交时间: 2024-05-06 14:20:49

02754: 八皇后

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

思路：可以不停for+if解决

代码

```
A = []
for a in range(1,9):
    for b in range(1,9):
        if b not in [a, a-1, a+1]:
            for c in range(1,9):
                if c not in [a,a-2,a+2, b,b-1,b+1]:
                    for d in range(1,9):
                        if d not in [a,a-3,a+3,b,b-2,b+2,c,c-1,c+1]:
                            for e in range(1,9):
                                if e not in[a,a-4,a+4,b,b-3,b+3,c,c-2,c+2,d,d-
1,d+1]:
                                    for f in range(1,9):
                                        if f not in [a,a-5,a+5,b,b-4,b+4,c,c-
3,c+3,d,d-2,d+2,e,e-1,e+1]:
                                            for g in range(1,9):
                                                if g not in [a,a-6,a+6,b,b-
5,b+5,c,c-4,c+4,d,d-3,d+3,e,e-2,e+2,f,f-1,f+1]:
                                                    for h in range(1,9):
```

```
                                                    if h not in [a,a-
7,a+7,b,b-6,b+6,c,c-5,c+5,d,d-4,d+4,e,e-3,e+3,f,f-2,f+2,g,g-1,g+1]:
                                                        A.append(''.join(
map(str, [a,b,c,d,e,f,g,h] )))
for _ in range(int(input())):
    print(A[int(input()) - 1])
```

代码运行截图 ==（至少包含有"Accepted"）==

03151: Pots

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

思路：把不同情况考虑清楚即可

代码

```
def bfs(A, B, C):
    start = (0, 0)
    visited = set()
    visited.add(start)
    queue = [(start, [])]

    while queue:
        (a, b), actions = queue.pop(0)

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

        next_states = [(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 next_states:
            if i not in visited:
                visited.add(i)
                new_actions = actions + [get_action(a, b, i)]
                queue.append((i, new_actions))

    return ["impossible"]


def get_action(a, b, next_state):
    if next_state == (A, b):
        return "FILL(1)"
    elif next_state == (a, B):
        return "FILL(2)"
    elif next_state == (0, b):
        return "DROP(1)"
    elif next_state == (a, 0):
        return "DROP(2)"
    elif next_state == (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())
solution = bfs(A, B, C)

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

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

#44892687提交状态

查看　提交　统计　提问

状态: Accepted

源代码

```
def bfs(A, B, C):
    start = (0, 0)
    visited = set()
    visited.add(start)
    queue = [(start, [])]

    while queue:
        (a, b), actions = queue.pop(0)

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

        next_states = [(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))]
```

基本信息

#:	44892687
题目:	03151
提交人:	23n2300012140(zyt)
内存:	3684kB
时间:	22ms
语言:	Python3
提交时间:	2024-05-07 23:13:17

05907: 二叉树的操作

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

思路：复习一下二叉树

代码

```
def swap(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(op[1], op[2])
        else:
            cur = op[1]
            while tree[cur][0] != -1:
                cur = tree[cur][0]
            print(cur)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态：Accepted

源代码

```
def swap(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(op[1], op[2])
        else:
            cur = op[1]
            while tree[cur][0] != -1:
                cur = tree[cur][0]
            print(cur)
```

基本信息

#: 44892712
题目: 05907
提交人: 23n2300012140(zyt)
内存: 3684kB
时间: 74ms
语言: Python3
提交时间: 2024-05-07 23:16:31

18250: 冰阔落 I

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

思路：看的题解做法

代码

```
def find(x):
    if parent[x] != x:
        parent[x] = find(parent[x])
    return parent[x]

def union(x, y):
    root_x = find(x)
    root_y = find(y)
    if root_x != root_y:
        parent[root_y] = root_x

while True:
    try:
        n, m = map(int, input().split())
        parent = list(range(n + 1))

        for _ in range(m):
            a, b = map(int, input().split())
            if find(a) == find(b):
                print('Yes')
            else:
                print('No')
                union(a, b)
```

```
        unique_parents = set(find(x) for x in range(1, n + 1))
        ans = sorted(unique_parents)
        print(len(ans))
        print(*ans)

    except EOFError:
        break
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态: Accepted

源代码

```
def find(x):
    if parent[x] != x:
        parent[x] = find(parent[x])
    return parent[x]

def union(x, y):
    root_x = find(x)
    root_y = find(y)
    if root_x != root_y:
        parent[root_y] = root_x

while True:
    try:
        n, m = map(int, input().split())
        parent = list(range(n + 1))

        for _ in range(m):
            a, b = map(int, input().split())
            if find(a) == find(b):
                print('Yes')
            else:
                print('No')
                union(a, b)

        unique_parents = set(find(x) for x in range(1, n + 1))
```

基本信息

#:	44892748
题目:	18250
提交人:	23n2300012140(zyt)
内存:	5504kB
时间:	374ms
语言:	Python3
提交时间:	2024-05-07 23:20:57

05443: 兔子与樱花

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

思路：看的题解做法

代码

```
import heapq
import math
def dijkstra(graph,start,end,P):
    if start == end: return []
    dist = {i:(math.inf,[]) for i in graph}
    dist[start] = (0,[start])
    pos = []
    heapq.heappush(pos,(0,start,[]))
    while pos:
```

```
        dist1,current,path = heapq.heappop(pos)
        for (next,dist2) in graph[current].items():
            if dist2+dist1 < dist[next][0]:
                dist[next] = (dist2+dist1,path+[next])
                heapq.heappush(pos,(dist1+dist2,next,path+[next]))
    return dist[end][1]

P = int(input())
graph = {input():{} for _ in range(P)}
for _ in range(int(input())):
    place1,place2,dist = input().split()
    graph[place1][place2] = graph[place2][place1] = int(dist)

for _ in range(int(input())):
    start,end = input().split()
    path = dijkstra(graph,start,end,P)
    s = start
    current = start
    for i in path:
        s += f'->({graph[current][i]})->{i}'
        current = i
    print(s)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

#44892767提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
import heapq
import math
def dijkstra(graph,start,end,P):
    if start == end: return []
    dist = {i:(math.inf,[]) for i in graph}
    dist[start] = (0,[start])
    pos = []
    heapq.heappush(pos,(0,start,[]))
    while pos:
        dist1,current,path = heapq.heappop(pos)
        for (next,dist2) in graph[current].items():
            if dist2+dist1 < dist[next][0]:
                dist[next] = (dist2+dist1,path+[next])
                heapq.heappush(pos,(dist1+dist2,next,path+[next]))
    return dist[end][1]

P = int(input())
graph = {input():{} for _ in range(P)}
for _ in range(int(input())):
    place1,place2,dist = input().split()
    graph[place1][place2] = graph[place2][place1] = int(dist)
```

基本信息

#: 44892767
题目: 05443
提交人: 23n2300012140(zyt)
内存: 3696kB
时间: 20ms
语言: Python3
提交时间: 2024-05-07 23:22:48

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。==

图论的东西在上学期学过，还算好处理。