

Assignment #B: 图论和树算

Updated 1709 GMT+8 Apr 28, 2024

2024 spring, Compiled by 数学科学学院 张羽扬

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

```
D=[[-1,0],[0,-1],[0,1],[1,0]]
k=0
def dfs(x,y):
    global s
    global a
    if a[x][y] == '-':
        return 0
    a[x][y] = '-'
    for i in range(4):
```

```

        dfs(x+D[i][0], y+D[i][1])
    return 1
a=[['-']*12]
for i in range(10):
    t=list(input())
    t=['-']+t+['-']
    a.append(t)
a=a+ [['-']*12]
for i in range(10):
    for j in range(10):
        k+=dfs(i+1,j+1)
print(k)

```

代码运行截图

状态: Accepted

源代码

```

D=[[-1,0],[0,-1],[0,1],[1,0]]
k=0
def dfs(x,y):
    global s
    global a
    if a[x][y] == '-':
        return 0
    a[x][y] = '-'
    for i in range(4):
        dfs(x+D[i][0], y+D[i][1])
    return 1
a=[['-']*12]
for i in range(10):
    t=list(input())
    t=['-']+t+['-']
    a.append(t)
a=a+ [['-']*12]
for i in range(10):
    for j in range(10):
        k+=dfs(i+1,j+1)
print(k)

```

基本信息

#: 44890686
 题目: 28170
 提交人: 2200010796Delphinida(2200010796)
 内存: 3636kB
 时间: 22ms
 语言: Python3
 提交时间: 2024-05-07 19:54:09

02754: 八皇后

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

思路:

代码

```

a=[]
t=[]
x=[]
def dfs(k):
    global t
    global x
    for i in range(8):
        for j in range(k):
            if i==t[j] or k-j==i-t[j] or k-j==t[j]-i:
                break

```

```

        else:
            t.append(i)
            if k==7:
                u=[]
                for i in range(8):
                    u.append(str(t[i]+1))
                x.append(''.join(u))
            else:
                dfs(k+1)
            t.pop()

    return
dfs(0)
n=int(input())
for i in range(n):
    l=int(input())
    print(x[l-1])

```

代码运行截图

状态: Accepted

源代码

```

a=[]
t=[]
x=[]
def dfs(k):
    global t
    global x
    for i in range(8):
        for j in range(k):
            if i==t[j] or k-j==i-t[j] or k-j==t[j]-i:
                break
        else:
            t.append(i)
            if k==7:
                u=[]
                for i in range(8):
                    u.append(str(t[i]+1))
                x.append(''.join(u))
            else:
                dfs(k+1)
            t.pop()
    return
dfs(0)
n=int(input())
for i in range(n):
    l=int(input())
    print(x[l-1])

```

基本信息

#: 44891155
 题目: 02754
 提交人: 2200010796Delphinida(2200010796)
 内存: 3628kB
 时间: 38ms
 语言: Python3
 提交时间: 2024-05-07 20:35:29

03151: Pots

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

思路: 利用一个字典记录最短路径

代码

```

a,b,c=[int(i) for i in input().split()]

```

```

D=dict()
D[(0,0)]=-1
def f(i,j):
    global D
    S=set()
    if i!=a and (a,j) not in D.keys():
        D[(a,j)]=(i,j,'FILL(1)')
        S.add((a,j))
    if j!=b and (i,b) not in D.keys():
        D[(i,b)]=(i,j,'FILL(2)')
        S.add((i,b))
    if i+j>=a and (a,i+j-a) not in D.keys():
        D[(a,i+j-a)]=(i,j,'POUR(2,1)')
        S.add((a,i+j-a))
    elif (i+j)<=a and (i+j,0) not in D.keys():
        D[(i+j,0)]=(i,j,'POUR(2,1)')
        S.add((i+j,0))
    if i+j>=b and (i+j-b,b) not in D.keys():
        D[(i+j-b,b)]=(i,j,'POUR(1,2)')
        S.add((i+j-b,b))
    elif i+j<=b and (0,i+j) not in D.keys():
        D[(0,i+j)]=(i,j,'POUR(1,2)')
        S.add((0,i+j))
    if (0,j) not in D.keys():
        D[(0,j)]=(i,j,'DROP(1)')
        S.add((0,j))
    if (i,0) not in D.keys():
        D[(i,0)]=(i,j,'DROP(2)')
        S.add((i,0))
    return S
T={(0,0)}
k=0
s=0
while 1:
    for i in T:
        if i[0]==c or i[1]==c:
            s=1
            p=i
            print(k)
            break
    if s==1:
        break
    else:
        k=k+1
        U=set()
        for i in T:
            U=U.union(f(i[0],i[1]))
        T=U
        if T==set():
            print('impossible')
            break
if s==1:
    x=[]
    while D[p]!=-1:
        x.append(D[p][-1])

```

```

p=(D[p][0],D[p][1])
x.reverse()
for i in x:
    print(i)#

```

代码运行截图

状态: Accepted

源代码

```

a,b,c=[int(i) for i in input().split()]
D=dict()
D[(0,0)]=-1
def f(i,j):
    global D
    S=set()
    if i!=a and (a,j) not in D.keys():
        D[(a,j)]=(i,j,'FILL(1)')
        S.add((a,j))
    if j!=b and (i,b) not in D.keys():
        D[(i,b)]=(i,j,'FILL(2)')
        S.add((i,b))
    if i+j>=a and (a,i+j-a) not in D.keys():
        D[(a,i+j-a)]=(i,j,'POUR(2,1)')
        S.add((a,i+j-a))
    elif (i+j)<=a and (i+j,0) not in D.keys():
        D[(i+j,0)]=(i,j,'POUR(2,1)')
        S.add((i+j,0))
    if i+j>=b and (i+j-b,b) not in D.keys():
        D[(i+j-b,b)]=(i,j,'POUR(1,2)')
        S.add((i+j-b,b))
    elif i+j<=b and (0,i+j) not in D.keys():
        D[(0,i+j)]=(i,j,'POUR(1,2)')
        S.add((0,i+j))
    if (0,j) not in D.keys():
        D[(0,j)]=(i,j,'DROP(1)')

```

基本信息

#: 44891830
 题目: 03151
 提交人: 2200010796Delphinida(2200010796)
 内存: 3824kB
 时间: 22ms
 语言: Python3
 提交时间: 2024-05-07 21:30:51

05907: 二叉树的操作

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

思路:

代码

```

class TreeNode:
    def __init__(self, val=0):
        self.val=val
        self.left=None
        self.right=None
    def build_tree(nodes_info):
        nodes=[TreeNode(i) for i in range(n)]
        for val, left, right in nodes_info:
            if left!=-1:
                nodes[val].left=nodes[left]
            if right!=-1:
                nodes[val].right=nodes[right]
        return nodes
    def swap_nodes(nodes, x, y):

```

```

for node in nodes:
    if node.left and node.left.val in [x,y]:
        node.left=nodes[y] if node.left.val==x else nodes[x]
    if node.right and node.right.val in [x,y]:
        node.right=nodes[y] if node.right.val==x else nodes[x]
def find_leftmost(node):
    while node and node.left:
        node=node.left
    return node.val if node else -1
for _ in range(int(input())):
    n,m=map(int,input().split())
    nodes_info=[tuple(map(int,input().split())) for _ in range(n)]
    ops=[tuple(map(int,input().split())) for _ in range(m)]
    nodes=build_tree(nodes_info)
    for op in ops:
        if op[0]==1:
            swap_nodes(nodes,op[1],op[2])
        elif op[0]==2:
            print(find_leftmost(nodes[op[1]]))#

```

代码运行截图

状态: Accepted

源代码

```

class TreeNode:
    def __init__(self, val=0):
        self.val=val
        self.left=None
        self.right=None
def build_tree(nodes_info):
    nodes=[TreeNode(i) for i in range(n)]
    for val, left, right in nodes_info:
        if left!=-1:
            nodes[val].left=nodes[left]
        if right!=-1:
            nodes[val].right=nodes[right]
    return nodes
def swap_nodes(nodes, x, y):
    for node in nodes:
        if node.left and node.left.val in [x,y]:
            node.left=nodes[y] if node.left.val==x else nodes[x]
        if node.right and node.right.val in [x,y]:
            node.right=nodes[y] if node.right.val==x else nodes[x]
def find_leftmost(node):
    while node and node.left:
        node=node.left
    return node.val if node else -1
for _ in range(int(input())):
    n,m=map(int,input().split())
    nodes_info=[tuple(map(int,input().split())) for _ in range(n)]

```

基本信息

#: 44891983
 题目: 05907
 提交人: 2200010796Delphinida(2200010796)
 内存: 3960kB
 时间: 156ms
 语言: Python3
 提交时间: 2024-05-07 21:42:18

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

```
import heapq

def dijkstra(adjacency, start):
    distances = {vertex: float('infinity') for vertex in adjacency}
    previous = {vertex: None for vertex in adjacency}
    distances[start] = 0
    pq = [(0, start)]

    while pq:
        current_distance, current_vertex = heapq.heappop(pq)
        if current_distance > distances[current_vertex]:
            continue

        for neighbor, weight in adjacency[current_vertex].items():
            distance = current_distance + weight
            if distance < distances[neighbor]:
                distances[neighbor] = distance
                previous[neighbor] = current_vertex
                heapq.heappush(pq, (distance, neighbor))

    return distances, previous

def shortest_path_to(adjacency, start, end):
    distances, previous = dijkstra(adjacency, start)
    path = []
    current = end
    while previous[current] is not None:
        path.insert(0, current)
        current = previous[current]
    path.insert(0, start)
    return path, distances[end]

# Read the input data
P = int(input())
```

```

places = {input().strip() for _ in range(P)}

Q = int(input())
graph = {place: {} for place in places}
for _ in range(Q):
    src, dest, dist = input().split()
    dist = int(dist)
    graph[src][dest] = dist
    graph[dest][src] = dist # Assuming the graph is bidirectional

R = int(input())
requests = [input().split() for _ in range(R)]

# Process each request
for start, end in requests:
    if start == end:
        print(start)
        continue

    path, total_dist = shortest_path_to(graph, start, end)
    output = ""
    for i in range(len(path) - 1):
        output += f"{path[i]}->({graph[path[i]][path[i+1]]})->"
    output += f"{end}"
    print(output)

```

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

代码量对我来说有点太大了, 很消耗体力, 做了三题没精力继续写了, 后几道题抄了。感觉dijkstra还是没有完全学会, 之后再看一下