Assignment #B: 图论和树算

Updated 1709 GMT+8 Apr 28, 2024

2024 spring, Complied by 王业成 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：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

```
#
graph=[list(input().strip()) for i in range(10)]
def dfs(graph,x,y):
    graph[x][y]="-"
    for dx,dy in [(-1,0),(1,0),(0,1),(0,-1)]:
        if 0<=x+dx<=9 and 0<=y+dy<=9 and graph[x+dx][y+dy]==".":
            dfs(graph,x+dx,y+dy)
num=0
for i in range(10):
    for j in range(10):
        if graph[i][j]==".":
```

```
            num+=1
            dfs(graph,i,j)
print(num)
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
graph=[list(input().strip()) for i in range(10)]
def dfs(graph,x,y):
    graph[x][y]="-"
    for dx,dy in [(-1,0),(1,0),(0,1),(0,-1)]:
        if 0<=x+dx<=9 and 0<=y+dy<=9 and graph[x+dx][y+dy]==".":
            dfs(graph,x+dx,y+dy)
num=0
for i in range(10):
    for j in range(10):
        if graph[i][j]==".":
            num+=1
            dfs(graph,i,j)
print(num)
```

基本信息

#:	44887846
题目:	28170
提交人:	wangyecheng
内存:	3640kB
时间:	20ms
语言:	Python3
提交时间:	2024-05-07 15:05:56

02754: 八皇后

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

思路:

代码

```
#
def queens():
    result=[]
    queen=[-1]*8
    def dfs(row):
        if row==8:
            result.append(queen.copy())
        else:
            for a in range(8):
                if hefa(row,a):
                    queen[row]=a
                    dfs(row+1)
                    queen[row]=-1
    def hefa(row,a):
        for i in range(row):
            if queen[i]==a or abs(row-i)==abs(a-queen[i]):
                return False
        return True
    dfs(0)
    return result
n=int(input())
a=queens()
for i in range(n):
```

```
    s=int(input())
    print("".join(str(j+1) for j in a[s-1]))
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
def queens():
    result=[]
    queen=[-1]*8
    def dfs(row):
        if row==8:
            result.append(queen.copy())
        else:
            for a in range(8):
                if hefa(row,a):
                    queen[row]=a
                    dfs(row+1)
                    queen[row]=-1
    def hefa(row,a):
        for i in range(row):
            if queen[i]==a or abs(row-i)==abs(a-queen[i]):
                return False
        return True
    dfs(0)
    return result
n=int(input())
a=queens()
for i in range(n):
    s=int(input())
    print("".join(str(j+1) for j in a[s-1]))
```

基本信息

#: 44888160
题目: 02754
提交人: wangyecheng
内存: 3636kB
时间: 33ms
语言: Python3
提交时间: 2024-05-07 15:44:22

03151: Pots

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

思路:

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"）==

状态: Accepted

基本信息

#: 44888911
题目: 03151
提交人: wangyecheng
内存: 3712kB
时间: 22ms
语言: Python3
提交时间: 2024-05-07 16:34:01

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),ma
        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)
```

05907: 二叉树的操作

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

思路:

代码

```
#
class Treenode():
    def __init__(self,value):
        self.value=value
        self.left=None
        self.right=None
s=int(input())
for _ in range(s):
    n,m=map(int,input().split())
    tree=[Treenode(i) for i in range(n)]
    parent=[0 for i in range(n)]
    for _ in range(n):
        x,y,z=map(int,input().split())
        tree[x].left=tree[y] if y!=-1 else None
        tree[x].right=tree[z] if z!=-1 else None
        if y!=-1:
            parent[y]=x
        if z!=-1:
            parent[z]=x
    for _ in range(m):
        lst=list(map(int,input().split()))
        if lst[0]==1:
            a=lst[1]
            b=lst[2]
            a_fa=parent[a]
            b_fa=parent[b]
            if a_fa==b_fa:
                tree[a_fa].left,tree[a_fa].right=tree[a_fa].right,tree[a_fa].left
            else:
                if tree[a_fa].left==tree[a] and tree[b_fa].left==tree[b]:

 tree[a_fa].left,tree[b_fa].left=tree[b_fa].left,tree[a_fa].left
                if tree[a_fa].right==tree[a] and tree[b_fa].left==tree[b]:

 tree[a_fa].right,tree[b_fa].left=tree[b_fa].left,tree[a_fa].right
                if tree[a_fa].left==tree[a] and tree[b_fa].right==tree[b]:

 tree[a_fa].left,tree[b_fa].right=tree[b_fa].right,tree[a_fa].left
                if tree[a_fa].right==tree[a] and tree[b_fa].right==tree[b]:

 tree[a_fa].right,tree[b_fa].right=tree[b_fa].right,tree[a_fa].right
                parent[a]=b_fa
                parent[b]=a_fa
        if lst[0]==2:
```

```
            node=tree[lst[1]]
            while node.left:
                node=node.left
            print(node.value)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态: Accepted

源代码

```
class Treenode():
    def __init__(self,value):
        self.value=value
        self.left=None
        self.right=None
s=int(input())
for _ in range(s):
    n,m=map(int,input().split())
    tree=[Treenode(i) for i in range(n)]
    parent=[0 for i in range(n)]
    for _ in range(n):
        x,y,z=map(int,input().split())
        tree[x].left=tree[y] if y!=-1 else None
        tree[x].right=tree[z] if z!=-1 else None
        if y!=-1:
            parent[y]=x
        if z!=-1:
            parent[z]=x
    for _ in range(m):
        lst=list(map(int,input().split()))
        if lst[0]==1:
            a=lst[1]
            b=lst[2]
            a_fa=parent[a]
            b_fa=parent[b]
            if a_fa==b_fa:
                tree[a_fa].left,tree[a_fa].right=tree[a_fa].right,tree[a
            else:
                if tree[a_fa].left==tree[a] and tree[b_fa].left==tree[b
                    tree[a_fa].left,tree[b_fa].left=tree[b_fa].left,tre
                if tree[a_fa].right==tree[a] and tree[b_fa].left==tree[b
                    tree[a_fa].right,tree[b_fa].left=tree[b_fa].left,tr
                if tree[a_fa].left==tree[a] and tree[b_fa].right==tree[b
                    tree[a_fa].left,tree[b_fa].right=tree[b_fa].right,t
                if tree[a_fa].right==tree[a] and tree[b_fa].right==tree
                    tree[a_fa].right,tree[b_fa].right=tree[b_fa].right,
                parent[a]=b_fa
                parent[b]=a_fa
        if lst[0]==2:
            node=tree[lst[1]]
            while node.left:
                node=node.left
            print(node.value)
```

基本信息

#:	44888803
题目:	05907
提交人:	wangyecheng
内存:	3808kB
时间:	83ms
语言:	Python3
提交时间:	2024-05-07 16:22:40

18250: 冰阔落 I

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

思路：注意：在并查集中，当一个节点的根节点更新为另一个节点时，如果该节点之后再次被更新为另一个节点的子节点，就会导致路径压缩未完全实现的情况。这可能会使得某些节点的根节点不是最深的根节点，而是更新过程中的某个中间节点。

例如：A路径压缩更新到B，但是B后来又更新为C了，导致A的根结点不是C。

代码

```
#
def find(x):
    if parent[x-1]!=x:
```

```
        parent[x-1]=find(parent[x-1])
    return parent[x-1]
def union(x,y):
    x1=find(x)
    y1=find(y)
    if x1!=y1:
        parent[y1-1]=x1
while True:
    try:
        n,m=map(int,input().split())
        parent=[i for i in range(1,n+1)]
        for _ in range(m):
            x,y=map(int,input().split())
            if find(x)==find(y):
                print("Yes")
            else:
                print("No")
                union(x,y)
        b=set(find(x) for x in range(1,n+1))
        print(len(b))
        c=sorted(b)
        print(" ".join(map(str,c)))
    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-1]!=x:
        parent[x-1]=find(parent[x-1])
    return parent[x-1]
def union(x,y):
    x1=find(x)
    y1=find(y)
    if x1!=y1:
        parent[y1-1]=x1
while True:
    try:
        n,m=map(int,input().split())
        parent=[i for i in range(1,n+1)]
        for _ in range(m):
            x,y=map(int,input().split())
            if find(x)==find(y):
                print("Yes")
            else:
                print("No")
                union(x,y)
        b=set(find(x) for x in range(1,n+1))
        print(len(b))
        c=sorted(b)
        print(" ".join(map(str,c)))
    except EOFError:
        break
```

基本信息

#:	44889622
题目:	18250
提交人:	wangyecheng
内存:	6084kB
时间:	427ms
语言:	Python3
提交时间:	2024-05-07 17:41:52

05443: 兔子与樱花

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

思路:

代码

```
#
def find(x):
    if parent[x-1]!=x:
        parent[x-1]=find(parent[x-1])
    return parent[x-1]
def union(x,y):
    x1=find(x)
    y1=find(y)
    if x1!=y1:
        parent[y1-1]=x1
while True:
    try:
        n,m=map(int,input().split())
        parent=[i for i in range(1,n+1)]
        for _ in range(m):
            x,y=map(int,input().split())
            if find(x)==find(y):
                print("Yes")
            else:
                print("No")
                union(x,y)
        b=set(find(x) for x in range(1,n+1))
        print(len(b))
        c=sorted(b)
        print(" ".join(map(str,c)))
    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

```python
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)
```

基本信息

#：	44889650
题目：	05443
提交人：	wangyecheng
内存：	3640kB
时间：	22ms
语言：	Python3
提交时间：	2024-05-07 17:45:32

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。==

dfs和bfs已经越来越熟练了，兔子与樱花图的维护一直写不对，参考了题解，还需要慢慢消化