Assignment #9: 图论：遍历，及 树算

Updated 1739 GMT+8 Apr 14, 2024

2024 spring, Complied by 田济维 物理学院

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

（python pycharm）

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

04081: 树的转换

http://cs101.openjudge.cn/dsapre/04081/

思路：

代码

```
#
class TreeNode:
    def __init__(self,parent=None):
        self.children = []
        self.parent = parent

def buildtree(seq):
    maxn = 0
    height = 0
```

```
    root = TreeNode()
    currentnode = root
    for x in seq:
        if x == "d":
            height +=1
            maxn = max(maxn,height)
            s = TreeNode(currentnode)
            currentnode.children.append(s)
            currentnode = s
        if x == "u":
            height-=1
            maxn = max(maxn,height)
            currentnode = currentnode.parent
    return [root,maxn]

def countheight(root):
    if root.children:
        depth = [countheight(root.children[i])+1+i for i in
range(len(root.children))]
        return max(depth)
    else:
        return 0

s = input()
l = buildtree(s)
root = l[0]
h1 = l[1]
h2 = countheight(root)
print(f"{h1} => {h2}")
```

代码运行截图 （至少包含有"Accepted"）

状态：Accepted

基本信

源代码

提

提交

```
class TreeNode:
    def __init__(self,parent=None):
        self.children = []
        self.parent = parent

def buildtree(seq):
    maxn = 0
    height = 0
    root = TreeNode()
    currentnode = root
    for x in seq:
        if x == "d":
            height +=1
            maxn = max(maxn,height)
            s = TreeNode(currentnode)
            currentnode.children.append(s)
            currentnode = s
        if x == "u":
            height-=1
            maxn = max(maxn,height)
            currentnode = currentnode.parent
```

08581: 扩展二叉树

http://cs101.openjudge.cn/dsapre/08581/

思路:

代码

```
#
class TreeNode:
    def __init__(self,key,parent=None):
        self.key = key
        self.parent = parent
        self.left = None
        self.right = None

def buildTree(seq):
    s = TreeNode(seq[0],None)
    current = s
    for x in seq[1:]:
        temp = TreeNode(x,current)
        if current.left == None:
            current.left = temp
        else:
            current.right = temp
        current = temp
        while  (current.left!=None and current.right!=None) or current.key == ".":
            current = current.parent
            if current == None:
                break
    return s

def inorder(tree):
    if tree.key !=".":
        inorder(tree.left)
        print(tree.key,end = "")
        inorder(tree.right)

def postorder(tree):
    if tree.key !=".":
        postorder(tree.left)
        postorder(tree.right)
        print(tree.key, end="")

s = input()
tree = buildTree(s)

inorder(tree)
print("")
```

```
postorder(tree)
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

基本信息
#
题目
提交人
内存
时间
语言
提交时间

源代码

```
class TreeNode:
    def __init__(self,key,parent=None):
        self.key = key
        self.parent = parent
        self.left = None
        self.right = None

def buildTree(seq):
    s = TreeNode(seq[0],None)
    current = s
    for x in seq[1:]:
        temp = TreeNode(x,current)
        if current.left == None:
            current.left = temp
        else:
            current.right = temp
        current = temp
        while  (current.left!=None and current.right!=None) or current.
            current = current.parent
            if current == None:
                break
    return s

def inorder(tree):
    if tree.key !=".":
        inorder(tree.left)
        print(tree.key,end = "")
```

22067: 快速堆猪

http://cs101.openjudge.cn/practice/22067/

思路:

代码

```
#
stack = []
minstack = [1000000]
while True:
    try:
        s = input().split()
    except EOFError:
        break
    else:
        if s[0] == "push":
            stack.append(int(s[1]))
```

```
            minstack.append(min(int(s[1]),minstack[-1]))
        elif s[0] == "pop":
            if stack:
                stack.pop()
                minstack.pop()
        else:
            if stack:
                print(minstack[-1])

```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

状态: Accepted

源代码

```
stack = []
minstack = [1000000]
while True:
    try:
        s = input().split()
    except EOFError:
        break
    else:
        if s[0] == "push":
            stack.append(int(s[1]))
            minstack.append(min(int(s[1]),minstack[-1]))
        elif s[0] == "pop":
            if stack:
                stack.pop()
                minstack.pop()
        else:
            if stack:
                print(minstack[-1])
```

04123: 马走日

dfs, http://cs101.openjudge.cn/practice/04123

思路:

代码

```
#
sx = [2,1,-1,-2,-2,-1,1,2]
sy = [1,2,2,1,-1,-2,-2,-1]
ans = [0]
def dfs(seq,x,y):

```

```
    if seq == n*m:
        ans[0]+=1
        return
    for i in range(8):
        tx = x+sx[i]
        ty = y+sy[i]
        if not (0<=tx<n and 0<=ty<m):
            continue
        elif M[tx][ty]== False:
            M[tx][ty]=True
            dfs(seq+1,tx,ty)
            M[tx][ty]=False



T = int(input())
for _ in range(T):
    n, m, x, y = map(int,input().split())
    ans[0] = 0
    M = [[False]*m for i in range(n)]
    M[x][y] = True
    dfs(1,x,y)
    print(ans[0])
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

状态：Accepted

源代码

```
sx = [2,1,-1,-2,-2,-1,1,2]
sy = [1,2,2,1,-1,-2,-2,-1]
ans = [0]
def dfs(seq,x,y):

    if seq == n*m:
        ans[0]+=1
        return
    for i in range(8):
        tx = x+sx[i]
        ty = y+sy[i]
        if not (0<=tx<n and 0<=ty<m):
            continue
        elif M[tx][ty]== False:
            M[tx][ty]=True
            dfs(seq+1,tx,ty)
            M[tx][ty]=False



T = int(input())
for _ in range(T):
```

28046: 词梯

bfs, http://cs101.openjudge.cn/practice/28046/

思路:

代码

```
#
from collections import deque
n = int(input())
buckets = {}
for _ in range(n):
    s = input()
    for i in range(4):
        bucket = f"{s[:i]}_{s[i+1:]}"
        if bucket not in buckets:
            buckets[bucket]=[s]
        else:
            buckets[bucket].append(s)

def bfs(s,e):
    que = deque([[s]])
    visited = {s:1}
    while que:
        t = que.popleft()
        word = t[-1]
        for i in range(4):
            bucket = f"{word[:i]}_{word[i+1:]}"
            for x in buckets[bucket]:
                if x == e:
                    return t
                elif x not in visited:
                    que.append(t+[x])
                    visited[x]=1
    return "NO"
s,e = input().split()
road = bfs(s,e)
if road == "NO":
    print(road)
    exit()
road.append(e)
print(" ".join(road))


```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

状态: Accepted

基本信息

源代码

```
from collections import deque
n = int(input())
buckets = {}
for _ in range(n):
    s = input()
    for i in range(4):
        bucket = f"{s[:i]}_{s[i+1:]}"
        if bucket not in buckets:
            buckets[bucket]=[s]
        else:
            buckets[bucket].append(s)

def bfs(s,e):
    que = deque([[s]])
    visited = {s:1}
    while que:
        t = que.popleft()
        word = t[-1]
        for i in range(4):
            bucket = f"{word[:i]}_{word[i+1:]}"
```

28050: 骑士周游

dfs, http://cs101.openjudge.cn/practice/28050/

思路:

代码

```
#
n = int(input())
sx = [2,1,-1,-2,-2,-1,1,2]
sy = [1,2,2,1,-1,-2,-2,-1]
def postoid(i,j,n):
    return i*n+j

def buildgraph(n):
    for x in range(n):
        for y in range(n):
            node_id = postoid(x,y,n)
            for i in range(8):
                tx = x+sx[i]
                ty = y+sy[i]
                if 0<=tx<n and 0<=ty<n:
                    anotherid = postoid(tx,ty,n)
                    graph[node_id].append(anotherid)


def sortnode(u):
    s = graph[u]
    eff = []
    for x in s:
```

```
        if Color[x]:
            w = 0
            for y in graph[x]:
                if Color[y]:
                    w+=1
            eff.append((x,w))
    temp = sorted(eff,key = lambda x:x[1])
    return temp

def dfs(depth,u,n):
    Color[u]=False
    if depth == n*n:
        Color[u]=True
        return True

    g = sortnode(u)
    for x in g:
        if Color[x[0]] and dfs(depth+1,x[0],n):
            Color[u]=True
            return True
    Color[u]=True
    return False


graph = {}
x,y = map(int,input().split())
for i in range(n*n):
    graph[i]=[]
buildgraph(n)
Color = [True for i in range(n * n)]
result = dfs(1,postoid(x,y,n),n)
if result:
    print("success")
else:
    print("fail")

```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

#44760863提交状态

查看 提交 统计 提问

状态: Accepted

基本信息

#: 44760863
题目: 28050
提交人: 23n2300011503
内存: 3996kB
时间: 28ms
语言: Python3
提交时间: 2024-04-23 11:23:13

源代码

```
n = int(input())
sx = [2,1,-1,-2,-2,-1,1,2]
sy = [1,2,2,1,-1,-2,-2,-1]
def postoid(i,j,n):
    return i*n+j

def buildgraph(n):
    for x in range(n):
        for y in range(n):
            node_id = postoid(x,y,n)
            for i in range(8):
                tx = x+sx[i]
                ty = y+sy[i]
                if 0<=tx<n and 0<=ty<n:
                    anotherid = postoid(tx,ty,n)
                    graph[node_id].append(anotherid)


def sortnode(u):
    s = graph[u]
    eff = []
```

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。

感觉做作业时被第一题搞心态了（一般觉得作业第一题很简单），整体来说本次作业难度不高。花了3h