Assignment #9: 图论：遍历，及 树算

2024 spring, Complied by 胡豪俊 工学院

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：Windows11

Python编程环境：Visual Studio Code

1. 题目

04081: 树的转换

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

代码

```
def tree_heights(s):
    old_height=0
    max_old=0
    new_height=0
    max_new=0
    stack=[]
    for c in s:
        if c=='d':
            old_height+=1
            max_old=max(max_old,old_height)

            new_height+=1
            stack.append(new_height)
            max_new=max(max_new,new_height)
        else:
            old_height-=1

            new_height=stack[-1]
            stack.pop()
    return f"{max_old} => {max_new}"

s=input().strip()
```

```
print(tree_heights(s))
##思路比较直接
```

#44762819提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
def tree_heights(s):
    old_height=0
    max_old=0
    new_height=0
    max_new=0
    stack=[]
    for c in s:
        if c=='d':
            old_height+=1
            max_old=max(max_old,old_height)

            new_height+=1
            stack.append(new_height)
            max_new=max(max_new,new_height)
        else:
            old_height-=1

            new_height=stack[-1]
            stack.pop()
    return f"{max_old} => {max_new}"

s=input().strip()
print(tree_heights(s))
```

基本信息

#: 44762819
题目: 04081
提交人: 22n2200011800
内存: 3656kB
时间: 31ms
语言: Python3
提交时间: 2024-04-23 15:08:03

English 帮助 关于

08581: 扩展二叉树

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

代码

```
def build_tree(preorder):
    if not preorder or preorder[0]=='.':
        return None,preorder[1:]
    root=preorder[0]
    left,preorder=build_tree(preorder[1:])
    right,preorder=build_tree(preorder)
    return (root,left,right),preorder

def inorder(tree):
    if tree is None:
        return ''
    root,left,right=tree
    return inorder(left)+root+inorder(right)

def postorder(tree):
    if tree is None:
        return ''
    root,left,right=tree
    return postorder(left)+postorder(right)+root

preorder=input().strip()

tree,_=build_tree(preorder)

print(inorder(tree))
```

```
print(postorder(tree))
##错了好几次，看了群里的讨论才意识到错哪了，于是改正
```

#44762847提交状态

查看 提交 统计 提问

状态: Accepted

基本信息

#: 44762847
题目: 08581
提交人: 22n2200011800
内存: 3668kB
时间: 29ms
语言: Python3
提交时间: 2024-04-23 15:10:22

源代码

```
def build_tree(preorder):
    if not preorder or preorder[0]=='.':
        return None,preorder[1:]
    root=preorder[0]
    left,preorder=build_tree(preorder[1:])
    right,preorder=build_tree(preorder)
    return (root,left,right),preorder

def inorder(tree):
    if tree is None:
        return ''
    root,left,right=tree
    return inorder(left)+root+inorder(right)

def postorder(tree):
    if tree is None:
        return ''
    root,left,right=tree
    return postorder(left)+postorder(right)+root

preorder=input().strip()

tree,_=build_tree(preorder)

print(inorder(tree))
print(postorder(tree))
```


English 帮助 关于

22067: 快速堆猪

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

代码

```
import heapq
from collections import defaultdict

out=defaultdict(int)
pigs_heap=[]
pigs_stack=[]

while True:
    try:
        s=input()
    except EOFError:
        break

    if s=="pop":
        if pigs_stack:
            out[pigs_stack.pop()]+=1
    elif s=="min":
        if pigs_stack:
            while True:
                x=heapq.heappop(pigs_heap)
                if not out[x]:
                    heapq.heappush(pigs_heap,x)
                    print(x)
```

```
                    break
                out[x]-=1
    else:
        y=int(s.split()[1])
        pigs_stack.append(y)
        heapq.heappush(pigs_heap,y)
##heapq和defaultdict有点忘了，复习了一下
```

#44762887提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
import heapq
from collections import defaultdict

out=defaultdict(int)
pigs_heap=[]
pigs_stack=[]

while True:
    try:
        s=input()
    except EOFError:
        break

    if s=="pop":
        if pigs_stack:
            out[pigs_stack.pop()]+=1
    elif s=="min":
        if pigs_stack:
            while True:
                x=heapq.heappop(pigs_heap)
                if not out[x]:
                    heapq.heappush(pigs_heap,x)
                    print(x)
                    break
                out[x]-=1
    else:
        y=int(s.split()[1])
        pigs_stack.append(y)
        heapq.heappush(pigs_heap,y)
```

基本信息

#: 44762887
题目: 22067
提交人: 22n2200011800
内存: 8736kB
时间: 351ms
语言: Python3
提交时间: 2024-04-23 15:14:07

English 帮助 关于

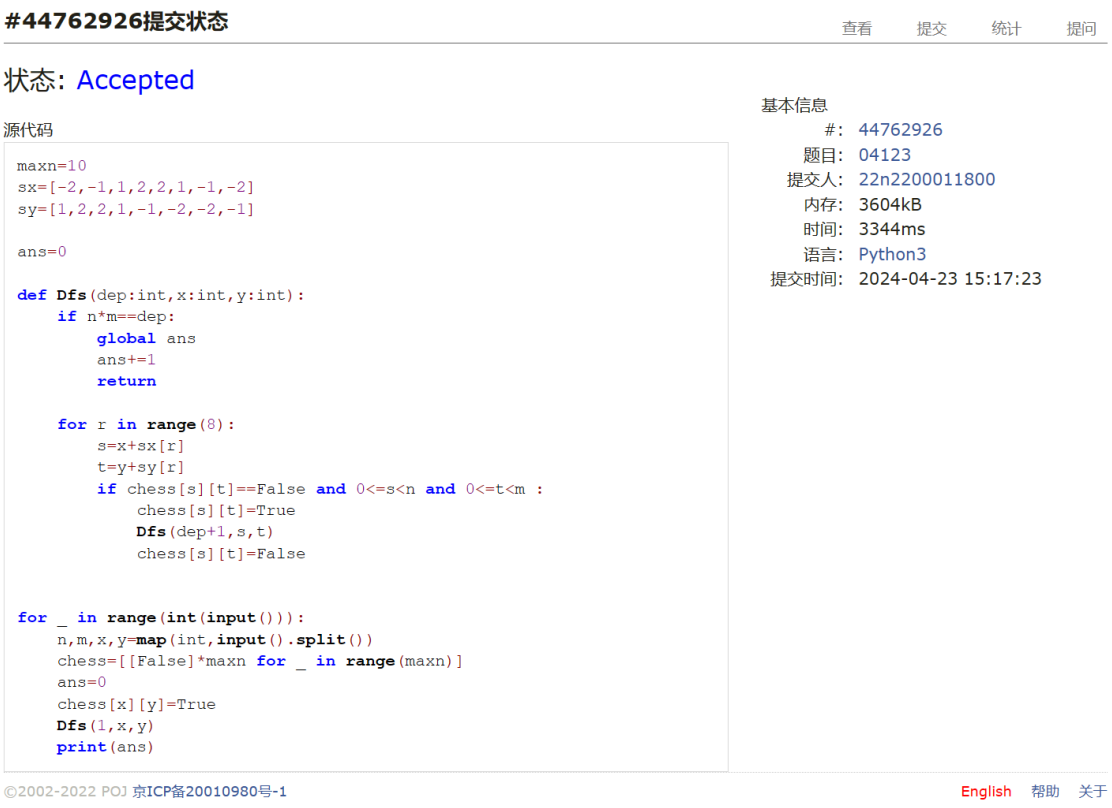
04123: 马走日

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

代码

```
maxn=10
sx=[-2,-1,1,2,2,1,-1,-2]
sy=[1,2,2,1,-1,-2,-2,-1]

ans=0

def Dfs(dep:int,x:int,y:int):
    if n*m==dep:
        global ans
        ans+=1
        return

    for r in range(8):
        s=x+sx[r]
        t=y+sy[r]
        if chess[s][t]==False and 0<=s<n and 0<=t<m :
            chess[s][t]=True
```

```
            Dfs(dep+1,s,t)
            chess[s][t]=False


for _ in range(int(input())):
    n,m,x,y=map(int,input().split())
    chess=[[False]*maxn for _ in range(maxn)]
    ans=0
    chess[x][y]=True
    Dfs(1,x,y)
    print(ans)
##复用了以前写过的代码
```



#44762926提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
maxn=10
sx=[-2,-1,1,2,2,1,-1,-2]
sy=[1,2,2,1,-1,-2,-2,-1]

ans=0

def Dfs(dep:int,x:int,y:int):
    if n*m==dep:
        global ans
        ans+=1
        return

    for r in range(8):
        s=x+sx[r]
        t=y+sy[r]
        if chess[s][t]==False and 0<=s<n and 0<=t<m :
            chess[s][t]=True
            Dfs(dep+1,s,t)
            chess[s][t]=False


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

基本信息

#: 44762926
题目: 04123
提交人: 22n2200011800
内存: 3604kB
时间: 3344ms
语言: Python3
提交时间: 2024-04-23 15:17:23

English 帮助 关于

28046: 词梯

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

代码

```
##时间不太够了，词梯看老师的ppt一直没彻底搞懂，申请先放着等期中全部考完再来解决
```

28050: 骑士周游

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

代码

```
import sys

class Graph:
```

```
    def __init__(self):
        self.vertices={}
        self.num_vertices=0

    def add_vertex(self,key):
        self.num_vertices=self.num_vertices+1
        new_ertex=Vertex(key)
        self.vertices[key]=new_ertex
        return new_ertex

    def get_vertex(self,n):
        if n in self.vertices:
            return self.vertices[n]
        else:
            return None

    def __len__(self):
        return self.num_vertices

    def __contains__(self,n):
        return n in self.vertices

    def add_edge(self,f,t,cost=0):
        if f not in self.vertices:
            nv=self.add_vertex(f)
        if t not in self.vertices:
            nv=self.add_vertex(t)
        self.vertices[f].add_neighbor(self.vertices[t],cost)

    def getVertices(self):
        return list(self.vertices.keys())

    def __iter__(self):
        return iter(self.vertices.values())


class Vertex:
    def __init__(self,num):
        self.key=num
        self.connectedTo={}
        self.color='white'
        self.distance=sys.maxsize
        self.previous=None
        self.disc=0
        self.fin=0

    def __lt__(self,o):
        return self.key<o.key

    def add_neighbor(self,nbr,weight=0):
        self.connectedTo[nbr]=weight


    def get_neighbors(self):
        return self.connectedTo.keys()
```

```
    def __str__(self):
        return str(self.key)+":color "+self.color+":disc "+str(self.disc)+":fin
"+ str(self.fin)+":dist "+str(self.distance)+":pred
\n\t["+str(self.previous)+"]\n"



def knight_graph(board_size):
    kt_graph=Graph()
    for row in range(board_size):
        for col in range(board_size):
            node_id=pos_to_node_id(row,col,board_size)
            new_positions=gen_legal_moves(row,col,board_size)
            for row2,col2 in new_positions:
                other_node_id=pos_to_node_id(row2, col2,board_size)
                kt_graph.add_edge(node_id,other_node_id)
    return kt_graph

def pos_to_node_id(x,y,bdSize):
    return x*bdSize+y

def gen_legal_moves(row,col,board_size):
    new_moves=[]
    move_offsets=[(-1,-2),(-1,2),(-2,-1),(-2,1),(1,-2),(1,2),(2,-1),(2,1)]
    for r_off, c_off in move_offsets:
        if 0<=row+r_off<board_size and 0<=col+c_off<board_size:
            new_moves.append((row+r_off,col+c_off))
    return new_moves

def knight_tour(n,path,u,limit):
    u.color="gray"
    path.append(u)
    if n<limit:
        neighbors=ordered_by_avail(u)
        i=0

        for nbr in neighbors:
            if nbr.color=="white" and \
                knight_tour(n+1,path,nbr,limit):
                return True
        else:
            path.pop()
            u.color="white"
            return False
    else:
        return True

def ordered_by_avail(n):
    res_list=[]
    for v in n.get_neighbors():
        if v.color=="white":
            c=0
            for w in v.get_neighbors():
                if w.color=="white":
                    c+=1
            res_list.append((c,v))
```

```
    res_list.sort(key=lambda x:x[0])
    return [y[1] for y in res_list]

def main():
    def NodeToPos(id):
       return ((id//8,id%8))

    bdSize=int(input())
    *start_pos,=map(int, input().split())
    g=knight_graph(bdSize)
    start_vertex=g.get_vertex(pos_to_node_id(start_pos[0],start_pos[1],bdSize))
    if start_vertex is None:
        print("fail")
        exit(0)

    tour_path=[]
    done=knight_tour(0,tour_path,start_vertex,bdSize*bdSize-1)
    if done:
        print("success")
    else:
        print("fail")

    exit(0)
    cnt=0
    for vertex in tour_path:
        cnt+=1
        if cnt%bdSize==0:
            print()
        else:
            print(vertex.key,end=" ")

if __name__=='__main__':
    main()

##学的老师给的解答，复杂
```

#44763534提交状态

查看 提交 统计 提问

状态: Accepted

基本信息

#: 44763534
题目: 28050
提交人: 22n2200011800
内存: 4064kB
时间: 31ms
语言: Python3
提交时间: 2024-04-23 16:05:55

源代码

```
import sys

class Graph:
    def __init__(self):
        self.vertices={}
        self.num_vertices=0

    def add_vertex(self,key):
        self.num_vertices=self.num_vertices+1
        new_ertex=Vertex(key)
        self.vertices[key]=new_ertex
        return new_ertex

    def get_vertex(self,n):
        if n in self.vertices:
            return self.vertices[n]
        else:
            return None

    def __len__(self):
        return self.num_vertices

    def __contains__(self,n):
        return n in self.vertices

    def add_edge(self,f,t,cost=0):
        if f not in self.vertices:
            nv=self.add_vertex(f)
        if t not in self.vertices:
            nv=self.add_vertex(t)
        self.vertices[f].add_neighbor(self.vertices[t],cost)

    def getVertices(self):
        return list(self.vertices.keys())

    def __iter__(self):
        return iter(self.vertices.values())


class Vertex:
```

2. 学习总结和收获

已经忙不过来了，明天和大后天还有两门期中考试，这周的词梯感觉没时间仔细思考只能先放着等五一假期补上了。这几周数算落下的有点多，急急急。