1带时间的DFS

Python

# 定义一个图或树结构的类

class Node:

def \_\_init\_\_(self, value):

self.value = value

self.children = []

self.visited = False

self.timestamp = None

# 带时间的DFS函数

def timed\_dfs(node, timestamp):

# 标记节点为已访问

node.visited = True

# 设置时间戳

node.timestamp = timestamp

# 处理当前节点

print(f"访问节点 {node.value}，时间戳为 {node.timestamp}")

# 遍历当前节点的子节点

for child in node.children:

# 如果子节点未被访问，则递归调用DFS函数

if not child.visited:

timestamp += 1

timestamp = timed\_dfs(child, timestamp)

timestamp += 1

return timestamp

# 创建一个示例图/树

# A

# / \

# B C

# / \ \

# D E F

# /

# G

# 创建节点

A = Node('A')

B = Node('B')

C = Node('C')

D = Node('D')

E = Node('E')

F = Node('F')

G = Node('G')

# 构建关系

A.children = [B, C]

B.children = [D, E]

C.children = [F]

F.children = [G]

# 使用带时间的DFS进行遍历

timed\_dfs(A, 1)