****示例****：使用字典的方式构建有向图，并搜索图中的路径。

图很容易通过列表和词典来构造。比如说，这有一张简单的图：

　A -> B  
　A -> C  
　A -> D  
　B -> E  
　C -> D  
　C -> F  
　D -> B  
　D -> E  
　E ->    
　F -> D  
　F -> G  
　G -> E  
这个图有6个节点(A-G)和8个弧。它可以通过下面的Python数据结构来表示：  
graph = {'A': ['B', 'C','D'],  
              'B': [ 'E'],  
              'C': ['D','F'],  
              'D': ['B','E','G'],  
              'E': [],  
              'F': ['D','G']  
             'G': ['E']}

代码

# 找到一条从start到end的路径

def findPath(graph,start,end,path=[]):

path = path + [start]

if start == end:

return path

for node in graph[start]:

if node not in path:

newpath = findPath(graph,node,end,path)

if newpath:

return newpath

return None

# 找到所有从start到end的路径

def findAllPath(graph,start,end,path=[]):

path = path +[start]

if start == end:

return [path]

paths = [] #存储所有路径

for node in graph[start]:

if node not in path:

newpaths = findAllPath(graph,node,end,path)

for newpath in newpaths:

paths.append(newpath)

return paths

# 查找最短路径

def findShortestPath(graph,start,end,path=[]):

path = path +[start]

if start == end:

return path

shortestPath = []

for node in graph[start]:

if node not in path:

newpath = findShortestPath(graph,node,end,path)

if newpath:

if not shortestPath or len(newpath)<len(shortestPath):

shortestPath = newpath

return shortestPath

'''

主程序

'''

graph = {'A': ['B', 'C','D'],

'B': [ 'E'],

'C': ['D','F'],

'D': ['B','E','G'],

'E': [],

'F': ['D','G'],

'G': ['E']}

onepath = findPath(graph,'A','G')

print('一条路径:',onepath)

allpath = findAllPath(graph,'A','G')

print('\n所有路径：',allpath)

shortpath = findShortestPath(graph,'A','G')

print('\n最短路径：',shortpath)