1.

class list\_node:

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

self.val = 0

self.next = None

head = [list\_node]\*5

new\_node = list\_node()

data = [[1, 2], [2, 1], [2, 3], [2, 4], [4, 3], [4, 1]]

print('圖形的鄰接串列內容:')

print('------------------------')

for i in range(1, 5):

head[i].val = i

head[i].next = None

print('頂點 %d =>' %i, end = '')

ptr = head[i]

for j in range(6):

if data[j][0] == i:

new\_node.val = data[j][1]

new\_node.next = None

while ptr != None:

ptr = ptr.next

ptr = new\_node

print('[%d] ' %new\_node.val, end = '')

print()

2.

graph = {

"A": ["D", "C", "B"],

"B": ["A", "E"],

"C": ["A", "F"],

"D": ["A", "G", "H"],

"E": ["B"],

"F": ["C", "I", "J"],

"G": ["D"],

"H": ["D"],

"I": ["F"],

"J": ["F"]

}

def DFS(graph, s):

stack = []

stack.append(s)

seen = set()

seen.add(s)

while (len(stack) > 0):

vertex = stack.pop()

nodes = graph[vertex]

for w in nodes:

if w not in seen:

stack.append(w)

seen.add(w)

print(vertex)

DFS(graph, "F")