Python实现Primepath步骤:

1.判断路径中是否有节点重复算法实现

2.深度优先遍历算法实现

3.求出图的simplepath集合

4.对simplepath中的path进行判断,如果此条路径不在其余路径之中，即为primepath

5.在主函数中输入图

附算法源码:

#coding:utf-8

import copy

def Repeat(path,node):

#判断路径中是否有节点重复

pathtemp = copy.deepcopy(path)

if len(pathtemp) > 0:

pathtemp.pop(0)

while len(pathtemp) > 0:

if pathtemp[0] == node:

return True

pathtemp.pop(0)

return False

def DFS(path, key):#深度优先遍历

value = graph[key]

for i in value:

while Repeat(path,i) == False:

path.append(i)

simplePath.append(copy.deepcopy(path))

if path[0]!=i:

DFS(copy.deepcopy(path),i)

path.pop()

return

simplePath = []

def SimplePath():#用DFS生成simplepath

for key in graph.keys():

path = []

path.append(key)

simplePath.append(copy.deepcopy(path))

DFS(copy.deepcopy(path),key)

path.pop()

return

primePath = []

def PrimePath():#求primepath

for path1 in simplePath:#对simplepath中的path进行判断

flag = False#初始化信号值

for path2 in simplePath:#如果不相等的话就是primepath

if path1 != path2:

if ','.join(path1) in ','.join(path2):

flag = True

break

if flag == False:

primePath.append(copy.deepcopy(path1))

graph = dict()#写入图

graph['0'] = ['1','2']

graph['1'] = ['3']

graph['2'] = ['3']

graph['3'] = ['0']

SimplePath()

PrimePath()

print primePath