云南大学数学与统计学院  
《算法图论实验》上机实践报告

|  |  |  |
| --- | --- | --- |
| 课程名称：算法图论实验 | 年级：2015级 | 上机实践成绩： |
| 指导教师：李建平 | 姓名：刘鹏 | 专业：信息与计算科学 |
| 上机实践名称：编程实现求图的所有连通分支 | 学号：20151910042 | 上机实践日期：2018-10-19 |
| 上机实践编号：2 | 组号： |  |

# 实验目的

1. 理解图的连通分支的概念，并能写出求一个图的所有连通分支的算法；
2. 学会运用图的遍历算法来解决问题。

# 实验内容

1. 写出求一个图的所有最大连通分支的算法；
2. 用C语言编程实现上述算法。

# 实验平台

Windows 10 Pro 1803；

MacOS Mojave。

# 算法设计

对于一个图，其连通分支所构成的集合是一个图集合，在该集合里，任取一个元素都是一个连通图，经过如下（1）、（2）的操作

（1）把任何属于集合（如果该集合非空）的顶点被添加到，得到一个新图；

（2）把所有属于的、集合中顶点之间的、不在的所有边都添加到中，得到一个新图；

得到的是不连通的。换言之，每个连通子图都是最大的，再也无法往其中添加顶点了。

通过对这个“极大性”的考察，可以将广度优先遍历算法或者深度优先算法进行修改，添加条件判断语句，使新的算法可以找到所输入图的所有连通分支。

下面对该算法进行形式化描述。

|  |  |
| --- | --- |
| **Algorithm**  **Input**  **Output**  **Begin**  **Step 1**  **Step 2**  **Step 3**  **End** | 寻找图的所有极大连通分支，记此算法为-  图的邻接矩阵  图的所有连通分支 |

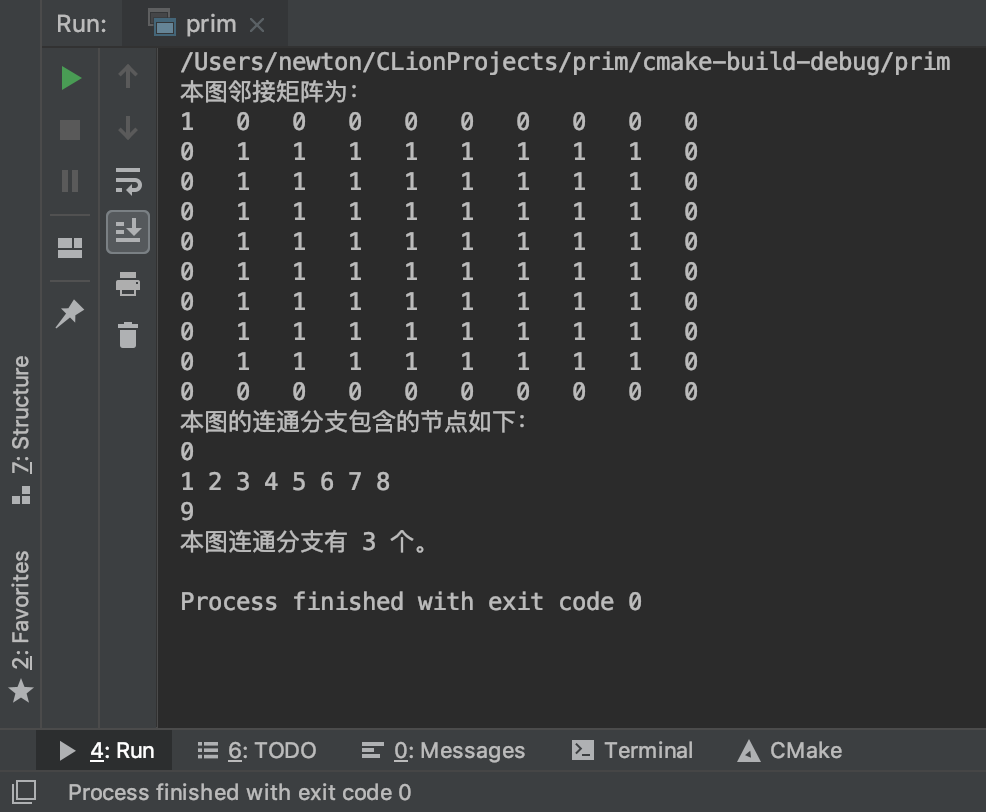
# 程序代码

利用C++语言，对上述算法进行简单实现，并且用一个拥有10顶点的图进行测试。

### 程序代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60 | #include <fstream>  #include <iostream>  #include <stdlib.h>  **using** **namespace** std**;**  void visit**(**int **\***visited**,** int num**)** **{**  visited**[**num**]** **=** 1**;**  printf**(**"%d "**,** num**);**  **}**  // 深度优先遍历算法  void DFS**(**int ph**[][**10**],** int **\***visited**,** int num**)** **{**  // visited是一个数组，表示该节点有没有被访问  **if(!**visited**[**num**])** **{**  visit**(**visited**,** num**);**  **}**  **for(**int i **=** num **+** 1**;** i **<** 10**;** i**++)** **{**  **if(**ph**[**num**][**i**]** **==** 1**)** **{**  **if(!**visited**[**i**])** **{**  visit**(**visited**,** i**);**  DFS**(**ph**,** visited**,** i**);**  **}**  **}**  **}**  **}**  int main**()** **{**  int ph**[**10**][**10**];**  int visited**[**10**]** **=** **{**0**};**  int flag **=** 0**;**  char ch**;**  ifstream OpenFile**(**"tu.txt"**);**  **for(**int j **=** 0**;** j **<** 10**;** j**++)** **{**  **for(**int k **=** 0 **;** k **<** 10**;** k**++)** **{**  OpenFile**.**get**(**ch**);**  ph**[**j**][**k**]** **=** **(**int**)**ch **-** 48**;**  **}**  **}**  cout **<<** "本图邻接矩阵为：" **<<** endl**;**  **for(**int j **=** 0**;** j **<** 10**;** j**++)** **{**  **for(**int k **=** 0**;** k **<** 10**;** k**++)** **{**  cout **<<** ph**[**j**][**k**]** **<<** " "**;**  **}**  cout **<<** endl**;**  **}**  cout **<<** "本图的连通分支包含的节点如下：" **<<** endl**;**  **for(**int j **=** 0**;** j **<** 10**;** j**++)** **{**  **if(!**visited**[**j**])** **{**  DFS**(**ph**,** visited**,** j**);**  cout **<<** endl**;**  flag**++;**  **}**  **}**  cout **<<** "本图连通分支有 " **<<** flag **<<**" 个。" **<<** endl**;**  OpenFile**.**close**();**  **}** |

### 运行结果



# 参考文献

[1] **林锐**. 高质量 C++/C 编程指南 [M]. 1.0 ed., 2001.

[2]