**云南大学数学系《算法图论实验》课程上机实验报告**

|  |  |  |
| --- | --- | --- |
| **课程名称**：算法图论实验 | **学期：**2016-2017学年第一学期 | **成绩**： |
| **指导教师**：李建平 | **学生姓名**：金洋 | **学生学号**：20131910023 |
| **实验名称**： **连通图和强连通图** | | |
| **实验编号**：No.1 | **实验日期**：**2016.9.2** | **实验学时**： **1** |
| **学院：** 数学与统计学院 | **专业： 信息与计算科学** | **年级**： **2013** |

**一、实验目的**

使用c语言实现判断一个无向图是否是连通图、设计算法求无向图的所有连通分支；

使用c语言实现判断一个有向图是否是强连通图、设计算法求有向图的所有强连通分支；

**二、实验内容**

给定一无向图G的邻接矩阵，如何判定G是连通图，设计算法求G的所有连通分支；

给定一有向图D的邻接矩阵，如何判定D是强连通图，设计算法求D的所有强连通分支；

**三、使用环境**

平台：Microsoft Visual C++ 6.0

语言：C语言

1. **算法介绍**

1. 任给图G，如何判断G是否是连通图.

Algorithm 判断连通图

Input G=(V,E);

Output G是否是连通图;

Begin

Step 1: X={};

Step 2: While (X≠V and (X)≠∅) do

1. 选取(X)中的所有边, j=1,2,...,k;
2. X=X∪{}，j=1,2,...,k;

Step 3: If (X==V) then 输出“G是连通的”

Else 输出“G不是连通的”;

End.

2. 任给图G，设计算法求G的所有连通分支.

Algorithm 求连通分支

Input G=(V,E);

Output G的所有连通分支;

Begin

Step 1: flag=0, 所有点为未标记状态

Step 2: flag=flag+1, 取一个未标记过的点, 令X={};

Step 3: 利用反圈法找到所能到达的所有点, 记为;

Step 4: X=X∪, 将X中的所有点标记为flag;

Step 5: If 仍有点未标记, Goto Step 2

Else 输出flag个连通分支 （标号相同的属于同一个连通分支）;

End.

3. 任给有向图D，如何判断D是否是强连通图.

Algorithm 判断强连通图

Input D=(V,A);

Output D是否是强连通图;

Begin

Step 1: X1={};

Step 2: While (X1≠V and (X1)≠∅) do

1.选取(X)中的所有边, j=1,2,...,k;

2.X1=X1∪{}，j=1,2,...,k;

Step 3: X2={};

Step 4: While (X2≠V and (X2)≠∅) do

1.选取(X2)中的所有边, j=1,2,...,k;

2.X2=X2∪{}，j=1,2,...,k;

Step 5: X=X1∩X2;

Step 3: If (X==V) then 输出“D是强连通的”

Else 输出“D不是强连通的”;

End.

4. 任给有向图D，设计算法求D的所有强连通分支.

Algorithm 求强连通分支

Input D=(V,A);

Output D的所有强连通分支;

Begin

Step 1: flag=0, 所有点为未标记状态

Step 2: flag=flag+1, 取一个未标记过的点, 令X={};

Step 3: 利用算法3的方法，找到包含的强连通子图的点集, 记为;

Step 4: X=X∪, 将X中的所有点标记为flag;

Step 5: If 仍有点未标记, Goto Step 2

Else输出flag个强连通分支(标号相同的属于同一个强连通分支);

End.

**五、调试过程**

1．程序代码

①判断连通图、求连通分支:

**CC.c**

#include <stdio.h>

#include <stdlib.h>

#define MAXN 999999999

#define MAXV 100

int M[MAXV][MAXV],X[MAXV],n;

int flag=0;//连通分支个数

/\* φ(X)是否为空? 是:返回0;否:返回1\*/

int Phi() {

int i,j;

for(i=1;i<=n;i++)

for (j=1;j<=n;j++)

if ((i!=j)&&(X[i]==flag)&&(X[j]==0)&&(M[i][j])) return(1);

return(0);

}

void prim() {

int i,j;

for(i=1;i<=n;i++)

for(j=1;j<=n;j++)

if ((i!=j)&&(X[i]==flag)&&(X[j]==0)&&(M[i][j])) {

X[j]=flag;

X[0]++;

}

}

void CC() {

int i=1,j;

//当仍有点未访问时

while (X[0]<n) {

while (X[i]>0) i++;

flag++;

X[i]=flag;

X[0]++;

while ( (X[0]<n) && Phi()) prim();

for (j=i;j<=n;j++)

if (X[j]==flag) printf("%d ",j);

printf("\n");

}

printf("共%d个连通分支\n",flag);

}

void main() {

int i,j;

//freopen("CC.in", "r", stdin);

printf("请输入G的顶点个数n=");

scanf("%d",&n);

printf("请输入G的邻接矩阵:\n");

for (i=1;i<=n;i++)

for (j=1;j<=n;j++)

scanf("%d",&M[i][j]);

printf("连通分支为:\n");

CC();

if (flag==1) printf("G是连通图\n");

else printf("G不是连通图\n");

}

②判断强连通图、求强连通分支:

**SCC.c**

#include <stdio.h>

#include <stdlib.h>

#define MAXN 999999999

#define MAXV 100

int M[MAXV][MAXV],X[MAXV],n;

int flag=0;//强连通分支个数

/\* φ+(X)是否为空? 是:返回0;否:返回1\*/

int Phi\_plus() {

int i,j;

for(i=1;i<=n;i++)

for (j=1;j<=n;j++)

if ((i!=j)&&(X[i]==-1)&&(X[j]==0)&&(M[i][j])) return(1);

return(0);

}

void prim1() {

int i,j;

for(i=1;i<=n;i++)

for(j=1;j<=n;j++)

if ((i!=j)&&(X[i]==-1)&&(X[j]==0)&&(M[i][j])) {

X[j]=-1;

X[0]++;

}

}

/\* φ-(X)是否为空? 是:返回0;否:返回1\*/

int Phi\_minus() {

int i,j;

for(i=1;i<=n;i++)

for (j=1;j<=n;j++)

if ((i!=j)&&(X[i]==flag)&&(X[j]==-1)&&(M[j][i])) return(1);

return(0);

}

void prim2() {

int i,j;

for(i=1;i<=n;i++)

for(j=1;j<=n;j++)

if ((i!=j)&&(X[i]==flag)&&(X[j]==-1)&&(M[j][i])) {

X[j]=flag;

X[0]++;

}

}

void SCC() {

int i=1,j;

int visited;

//当仍有点未访问时

while (X[0]<n) {

while (X[i]>0) i++;

flag++;

X[i]=-1;

X[0]++;

visited=X[0];

while ( (X[0]<n) && Phi\_plus()) prim1();

X[0]=visited;

X[i]=flag;

while ( (X[0]<n) && Phi\_minus()) prim2();

for (j=i;j<=n;j++)

if (X[j]==flag) printf("%d ",j);

else if (X[j]==-1) X[j]=0;

printf("\n");

}

printf("共%d个强连通分支\n",flag);

}

void main() {

int i,j;

//freopen("SCC.in", "r", stdin);

printf("请输入D的顶点个数n=");

scanf("%d",&n);

printf("请输入D的邻接矩阵:\n");

for (i=1;i<=n;i++)

for (j=1;j<=n;j++)

scanf("%d",&M[i][j]);

printf("强连通分支为:\n");

SCC();

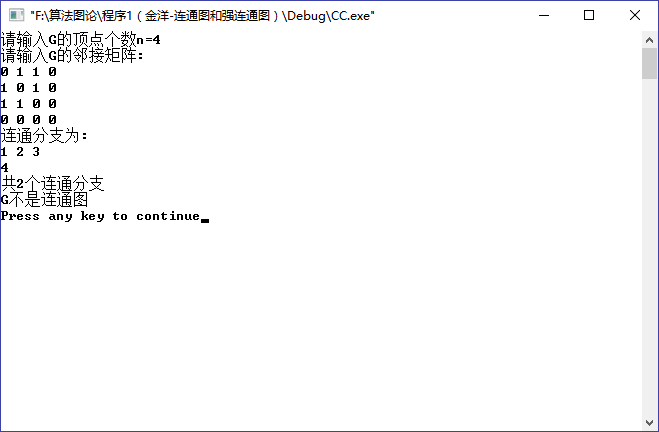
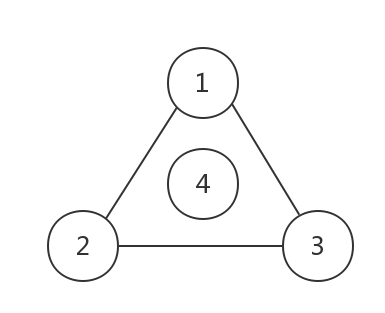
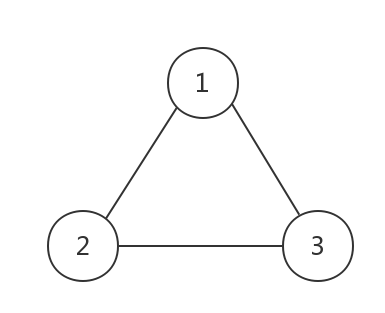
if (flag==1) printf("D是强连通图\n");

else printf("D不是强连通图\n");

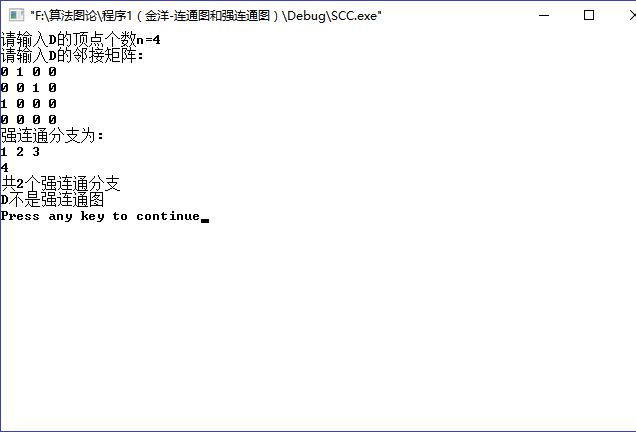
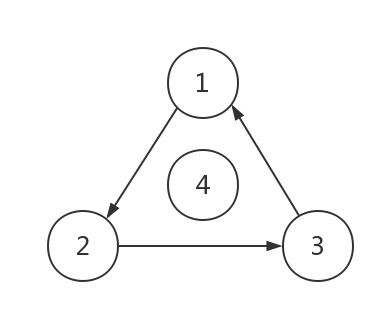
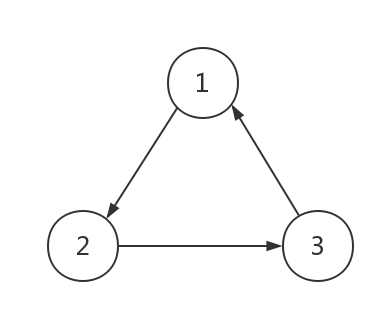
}

1. 运行窗口

①判断连通图、求连通分支:



②判断强连通图、求强连通分支:



**六、总结**

1.学会使用c语言实现判断一个无向图是否是连通图、设计算法求无向图的所有连通分支；

2. 学会使用c语言实现判断一个有向图是否是强连通图、设计算法求有向图的所有强连通分支；

**七、参考文献**

[1] 谭浩强，c程序设计（第三版），清华大学出版社，2005年7月；

[2] 《运筹学》教材编写组，运筹学（第四版），清华大学出版社，2012年9月；

**八、教师评语**