

离散数学

实验指导书

合肥工业大学 计算机与信息学院
人工智能学院

王晓华，杨娟，李书杰，汪荣贵

2023 年 11 月

目 录

目 录	1
概 述	3
第一篇 数理逻辑	4
实验 1 逻辑运算表示与验证	4
1.1 实验要求和目的	4
1.2 实验课时	4
1.3 实验环境	4
1.4 实验平台简介	5
实验 2 交通信号灯模型设计	7
2.1 实验要求和目的	7
2.2 实验课时	7
2.3 实验环境	7
2.4 思考题	8
第二篇 集合与关系	9
实验 3 小型关系数据库系统实现	9
3.1 实验要求和目的	9
3.2 实验课时	10
3.3 实验环境	10
3.4 实验平台简介	11
实验 4 关系性质的判定与性质闭包求解	13
4.1 实验要求和目的	13
4.2 实验课时	13
4.3 实验环境	13
4.4 实验平台简介	14
实验 5 偏序关系哈斯图生成与任务调度	15
5.1 实验要求和目的	15
5.2 实验课时	15
5.3 实验环境	16
5.4 实验效果展示	16
第三篇 整数论	17
实验 6 数论之模运算	17
6.1 实验要求和目的	17
6.2 实验课时	17
6.3 实验环境	17

6.4	实验平台简介.....	20
第四篇	图论	21
实验 7	最短路径的 Dijkstra 与 Floyd-Warshall 算法实现	21
7.1	实验要求和目的	21
7.2	实验课时.....	21
7.3	实验环境.....	22
7.4	实验平台简介.....	22
实验 8	欧拉回路的 DFS 或 Fleury 算法求解	25
8.1	实验要求和目的	25
8.2	实验课时.....	25
8.3	实验环境.....	25
8.4	实验平台简介.....	26
实验 9	最优树的 Huffman 算法求解	27
9.1	实验要求和目的	27
9.2	实验课时.....	27
9.3	实验环境.....	27
9.4	实验平台简介.....	23
第五篇	智能应用	29
实验 10	城市道路交通信号相位设置与配时系统设计与实现	29
10.1	实验要求和目的	29
10.2	实验课时.....	29
10.3	实验环境.....	29
10.4	实验平台简介.....	29
10.5	思考题	30
附录 A:	实验报告	31

概 述

(1) 实验概述

运用某种程序设计语言选择并设计实现离散数学的数理逻辑系统、集合与关系、整数论、图论四个篇章共 8 个实验的相关内容, 以检验和巩固离散数学中的基本知识、加深对本课程定义、定理、算法的理解, 锻炼和培养学生的计算思维能力, 完成系统设计, 实现问题求解, 达到有效提升综合实践能力和创新能力的目的, 并有效缓解离散数学课程概念多、理论性强、高度抽象等特点给学生带来的畏难情绪。

(2) 实验目的和要求

实验目的是检验和巩固所学知识与方法, 通过实现相关章节内容的编码过程, 更加形象地理解并掌握离散数学的定义、定理、原理和算法, 明确其作为专业基础课的核心地位, 并灵活运用相关知识解决实际问题。

要求理解各实验相关的定义、定理、算法和实现方法, 通过这些实验的训练, 加深对课程中相关概念和算法的理解, 验证和巩固离散数学中的基本知识, 锻炼和培养学生熟练掌握和应用相关编程环境 (例如 MFC), 理解课程中基本问题的求解方法和相关改进技术, 并对结果进行充分测试。

(3) 实验环境

包括但不限于: Visual Studio、Python 等

(4) 实验内容

- 实验 1: 逻辑运算表示与验证
- 实验 2: 交通信号灯模型设计
- 实验 3: 小型关系数据库系统实现
- 实验 4: 关系性质的判定与性质闭包求解
- 实验 5: 偏序关系哈斯图生成与任务调度
- 实验 6: 数论之模运算
- 实验 7: 最短路径的 Dijkstra 与 Floyd-Warshall 算法实现
- 实验 8: 欧拉回路的 DFS 或 Fleury 算法求解
- 实验 9: 最优树的 Huffman 算法求解
- 实验 10: 城市道路交通信号相位设置与配时系统设计与实现

(5) 实验课时要求

以上实验的课时设置均为 4 学时, 可从中任选 2-3 个作为拟完成的

实验任务。

第一篇：数理逻辑

实验 1：逻辑运算表示与验证

1.1 实验要求和目的

- 实验目的：理解命题逻辑相关运算，增强逻辑推理能力
- 实验要求：
 - (1).从键盘输入命题常元的真值求他们的合取、析取、求反、异或、单条件、双条件以及命题表达式的真值。
 - (2).求任意一个给定命题表达式的真值表，并根据真值表给出主范式。

1.2 实验课时

4 学时

1.3 实验环境

- 开发环境：Visual Studio、Python 等
- 本实验可提供实验平台，从以下两种方式中任意选择：
 - (1).借助控制台实现实验要求中的相关运算
 - (2).借助 MFC 实现实验要求中的相关运算

1.4 实验平台简介

- 平台界面：如图 1-1 至 1-4 所示：

```

***** 逻辑运算 *****
1---逻辑与运算
2---逻辑或运算
3---逻辑非运算
4---逻辑异或运算
5---逻辑单条件运算
6---逻辑双条件运算
7---命题常元表达式运算
8---命题变元表达式运算
0---程序退出
*****

请选择您要进行的逻辑运算的序号: 1

-----使用说明-----
==      ture用'1'      false用'0'      ==
==      真值与真值之间使用空格键隔开      ==
==      不要输入多余的空格键      ==
==      输入最后以#键结束      ==
=====
请输入真值value1和value2: 1 0#
逻辑与结果为:0

```

图 1-1 命题常元的相关逻辑运算

```

***** 逻辑运算 *****
1---逻辑与运算
2---逻辑或运算
3---逻辑非运算
4---逻辑异或运算
5---逻辑单条件运算
6---逻辑双条件运算
7---命题常元表达式运算
8---命题变元表达式运算
0---程序退出
*****

请选择您要进行的逻辑运算的序号: 8

-----使用说明-----
==      命题变元所用字母不能重复      ==
==      且变元个数不超过8个      ==
==      每个逻辑变元用单个字母表示      ==
==      与逻辑用'&'      或逻辑用'|'      ==
==      非逻辑用'!'      异或逻辑用'^'      ==
==      单条件用'>'      双条件用'='      ==
=====

请输入要计算的逻辑表达式(以#键结束): (a+b)>c#

真值表如下:
a b c Value
0 0 0 1
0 0 1 0
0 1 0 1
0 1 1 1
1 0 0 1
1 0 1 1
1 1 0 1
1 1 1 1

析合范式为:(!a*!b*!c)+(!a*b*!c)+(!a*b*c)+(a*!b*!c)+(a*!b*c)+(a*b*!c)+(a*b*c)
合析范式为:(!a*!b+c)

```

图 1-2 命题表达式真值表及范式求解

图 1-3 命题常元的相关逻辑运算

图 1-4 命题表达式真值表及范式求解

实验 2: 交通信号灯模型设计

2.1 实验要求和目的

- 实验目的: 掌握命题逻辑在计算机中的表示。
- 实验要求:
 - (1). 能够选用合适的表示方法, 解决与命题逻辑相关的问题。
 - (2). 设计实现一个简单的交通信号灯模型, 如图 2-1 所示。要求当定时时间到的时候, 该逻辑电路即发出一个信号给交通灯, 交通灯模型则根据当前状态下的信号进行信号颜色切换 (当前是绿灯信号, 则变为黄灯亮; 若当前是黄灯信号, 则切换为红灯亮; 若当前是红灯信号, 则切换为绿灯亮)。具体要求如下:

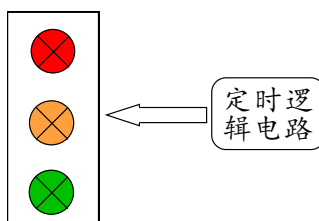


图 2-1 通过定时逻辑电路控制交通灯模型示意图

- 1) 编写函数实现, 每次接收到定时时间到消息时, 实现三种交通信号间的正确切换。
- 2) 在界面上用“定时”按钮或以系统时间展示信号灯之间的切换功能。
- 3) 切换到黄灯信号时考虑闪烁功能, 具体时长自行设置。
- 4) 考虑现实实际, 把握不同信号之间的时间间隔, 并增加设置界面, 做到自如修改。

2.2 实验课时

4 学时

2.3 实验环境

- 开发环境: Visual Studio、Python 等
- 本实验不提供实验平台, 要求同学们自行实现

2.4 思考题(选做)

- 该交通信号灯模型可根据实际情况如何进一步改进?
- 若考虑交叉路口及不同车道的实际情况, 如何设计算法进行实现?

第二篇：集合与关系

实验 3：小型关系数据库系统实现

3.1 实验要求和目的

- 实验目的：理解关系是笛卡尔积的子集，以及关系相关运算的原理；了解选择、投影运算；编程实现笛卡尔积、连接运算。同学们通过本实验更加深入直观地了解关系运算，深刻领会离散数学是数据库原理及应用的直接先行课。
- 实验要求：
 - (1). 基于已有实验平台了解关系的增加、删除、修改操作，熟悉关系的选择、投影操作。
 - (2). 在已有平台的基础上实现笛卡尔积运算和几种常用的连接运算。

3.2 实验课时

4 学时

3.3 实验环境

本试验提供实验平台

- 开发环境：Visual Studio、Python 等
- 实验平台：GridTest（自制平台）

3.4 实验平台简介

本实验提供名为 GridTest 的平台，该平台提供了对三张表添加数据和删除数据的功能，还提供了关系运算中的选择、投影、笛卡尔积、连接运算。

- 平台界面：如图 3-1 所示
- 数据添加，界面如图 3-2 所示

(1). 用户点击“添加数据”开始输入基本信息；

- 数据删除：与数据添加类似
 - (2).用户先选择一条记录
 - (3).再点击“删除数据”删除一条记录：
- 选择界面：如图 3-3 所示
- 投影界面：如图 3-4 所示
- 笛卡尔积界面：如图 3-5 所示
 - (4).实现函数：`void CGridTestDlg::Cartesian(int tab_index1, int tab_index2)`
- 连接界面：如图 3-6 所示
 - (5).实现函数：`void CGridTestDlg::Join(int table_name, CString field1, CString field2, int joinType)`

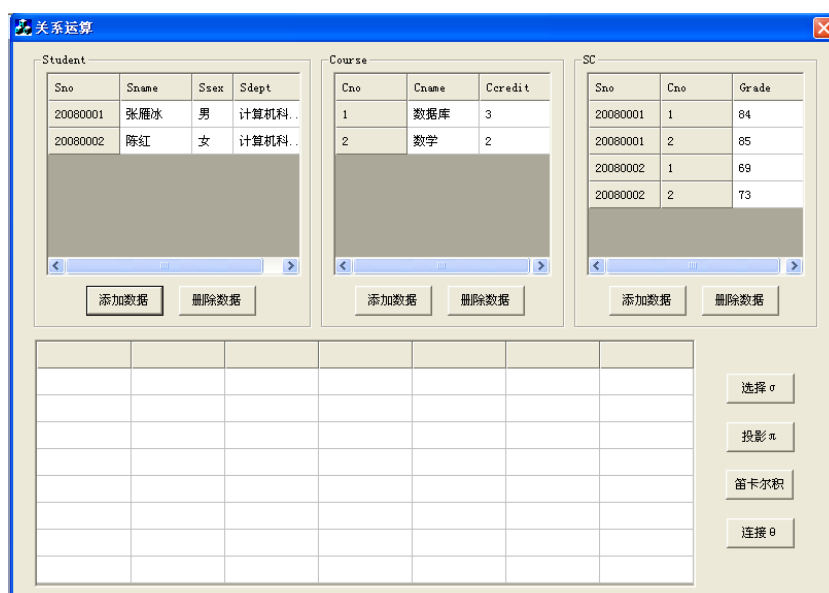


图 3-1 实验平台界面

Figure 3-2 shows the '关系运算' (Relation Operation) application window. It contains three tables: 'Student', 'Course', and 'SC'. The 'Student' table has columns Sno, Sname, Ssex, and Sdept. The 'Course' table has columns Cno, Cname, and Ccredit. The 'SC' table has columns Sno and Cno. A modal dialog titled '学生信息' (Student Information) is open, allowing the user to input student details. The dialog fields are: 学号 (Student ID) with value 20080003, 姓名 (Name) with value 李丽, 性别 (Gender) with value 女, and 系别 (Department) with value 应用数学. There are '确定' (OK) and '取消' (Cancel) buttons at the bottom of the dialog.

Sno	Sname	Ssex	Sdept
20080001	张雁冰	男	计算机科..
20080002	陈红	女	计算机科..

Cno	Cname	Ccredit
1	数据库	3
2	数学	2

Sno	Cno
2008	
2008	

学生信息

学号: 20080003 (如 20080001)
姓名: 李丽 (如 张三)
性别: 女
系别: 应用数学

确定 取消

图 3-2 输入学生信息界面

Figure 3-3 shows the '选择' (Select) dialog. It provides an example: '选择属于计算机科学与技术的学生的学号和姓名' (Select student IDs and names belonging to Computer Science and Technology). The interface includes sections for 'select' (选择任意多个), 'from' (选择一个表项), and 'where' (选择一个字段). The 'select' section has checkboxes for Sno, Sname, Ssex, and Sdept. The 'from' section has radio buttons for Student, Course, and SC. The 'where' section has a dropdown for 'where' and a text box for '条件值'. There are also radio buttons for 'and' and 'or' between conditions. At the bottom, there are '确定' (OK) and '取消' (Cancel) buttons.

选择

举例: 选择属于计算机科学与技术的学生的学号和姓名

select 选择任意多个

☐ Sno
☐ Sname
☐ Ssex
☐ Sdept

from 选择一个表项

☒ Student
☐ Course
☐ SC

where 选择一个字段

条件值

☐ and
☐ or

字段

条件值

确定 取消

图 3-3 选择界面

Figure 3-4 shows the '投影' (Projection) dialog. It features a large Greek letter π on the left. The title is '选择任意多项' (Select any multiple items). There is a list of checkboxes for Sno, Sname, Ssex, and Sdept. To the right, the text '查询表中某些列的信息' (Query information of some columns in the table) is followed by a dropdown menu and parentheses, with an example '(如 Student)' (e.g., Student). At the bottom, there are '确定' (OK) and '取消' (Cancel) buttons.

投影

选择任意多项

☐ Sno
☐ Sname
☐ Ssex
☐ Sdept

π

查询表中某些列的信息

()
(如 Student)

确定 取消

图 3-4 投影界面

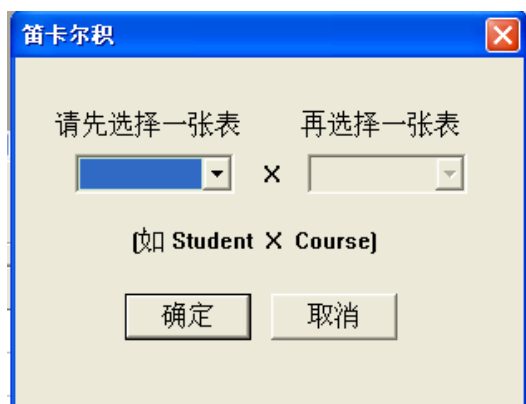


图 3-5 笛卡尔积界面

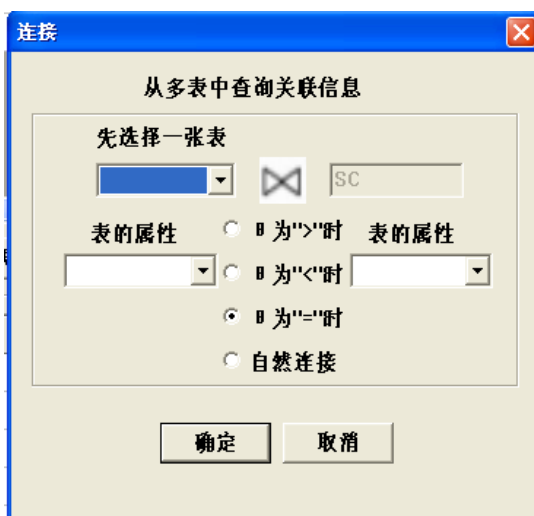


图 3-6 连接界

面

实验 4: 关系性质的判定与性质闭包求解

4.1 实验要求和目的

- 实验目的:

对关系性质进行研究可以更深入地了解关系的本质以及关系之间的联系。不同性质的组合可以构成不同的特殊关系。特殊关系比如等价关系是集合结点间的连通关系、图论最小生成树等内容的直接应用。

关系的闭包运算是对其某一不满足某种特性的关系进行最“经济”(即增加尽可能少的序对)扩充的运算,使之具有某一特性。关系 R 的闭包运算主要包括自反、对称和传递闭包,分别记为 $r(R)$ 、 $s(R)$ 和 $t(R)$ 。在熟悉计算原理的基础上,编程实现关系性质的判定及三种性质闭包的求解。

- 实验要求:

- (1).结合关系性质判定定理对给定关系进行五种性质判定,给出明确输出结果。
- (2).在完成对关系自反、对称、传递性质判定的基础上利用关系矩阵求解有限集上给定关系的自反、对称和传递闭包。
- (3).自反和对称闭包的求解相对简单。自反闭包只需将关系矩阵的主对角线元素全部置 1,对称闭包则是在求出当前关系矩阵的转置矩阵的基础上和原关系矩阵进行求和。
- (4).传递闭包的求解相对复杂,要求编码实现关系幂运算和 Warshall 算法两种求解方法,并做到相互验证及完成算法效率对比。

4.2 实验课时

4 学时

4.3 实验环境

开发环境: 不限, Visual Studio、Python 等

4.4 部分界面展示

- 界面：MFC 界面如图 4-1 所示，Python 页面如图 4-3 所示。
 - 数据输入：可以在 Visual Studio MFC/控制台界面或 Python 页面输入关系矩阵的大小，以及按行输入整个关系矩阵，每行之间用空格或换行符隔开。
- 矩阵验证：通过验证确认输入是否正确，并重新修正。
- 运行算法判定关系性质并实现闭包求解，扩展代码实现相关算法效率对比。
- 借助 Warshall 算法求解传递闭包的测试样例（如图 4-2）。

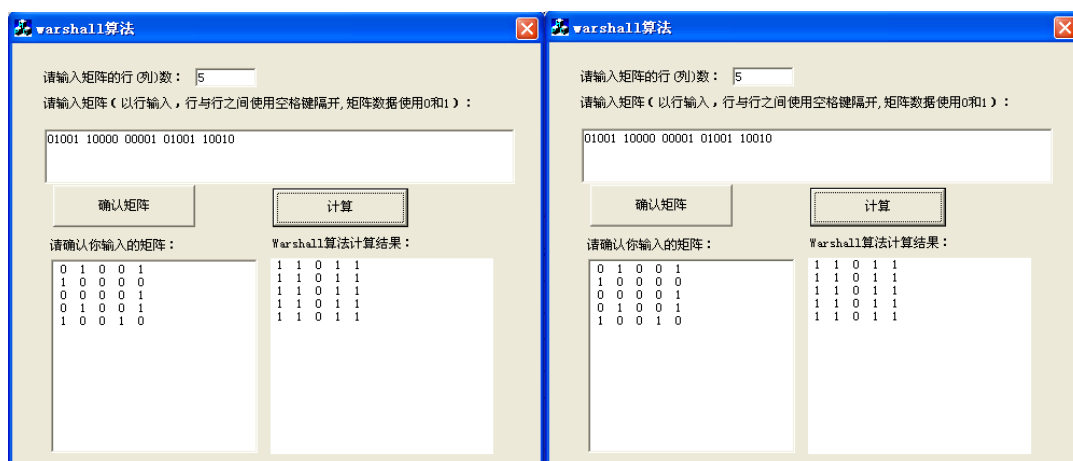


图 4-1 输入关系矩阵界面

图 4-2 传递闭包的求解

```

25     print(n)
26     relation = [[0] * n] * n
27     for i in range(n):
28         relation[i] = list(map(int, (input("请输入关系矩阵: (同行数字用空格分隔,
29         for i in range(n):
30             print(relation[i])
31     # Code 1: 使用flag标志位判断是否具备自反性, 反自反性质无从判断
32     i=0
33     flag=reflexive(i,n)
34     if flag:
35
if __name__ == '__main__': for i in range(n)

```

test ×

请输入集合中元素个数: 3

请输入关系矩阵: (同行数字用空格分隔, 不同行则用回车换行) 1 0 0

请输入关系矩阵: (同行数字用空格分隔, 不同行则用回车换行) 0 1 0

请输入关系矩阵: (同行数字用空格分隔, 不同行则用回车换行) 0 0 1

[1, 0, 0]

[0, 1, 0]

[0, 0, 1]

Code 1: 该关系是自反的

Code 2: 该关系是自反的

Code 3: 该关系是自反的

图 4-3 关系性质判定

实验 5: 偏序关系哈斯图生成与任务调度

5.1 实验要求和目的

- 实验目的:

- (1).在实验 4 对关系性质判定的基础上, 掌握不同关系性质的组合可获得不同的特殊关系; 例如等价关系、相容关系、偏序关系。
- (2).掌握关系结构在计算机中的表示方法。
- (3).解决特殊关系中偏序关系的哈斯图生成问题, 集合极小元素的求解问题、能够根据输入完成任务调度, 并进行适当的效率分析。

- 实验要求:

- (1).在项目安排何工作调度中, 经常会遇到一些流程图对应的拓扑排序问题。比如一个项目由若干个工作任务构成, 一些任务的完成需要在其他任务完成的基础上才能开始, 即串行执行; 而一些任务可以与另一些任务并行执行。
- (2).构造一个任务集合上的偏序关系, 比如可以借助自然数集合上的整除关系进行。首先输入一个自然数集合的子集, 结合整除运算编写程序, 生成哈斯图。
- (3).把哈斯图中的自然数结点看作任务名称, 求出任务的完成顺序 (根据以下算法 5.4)。
- (4).输出对应哈斯图中的反链, 并标识反链对应的可并行执行任务结点。

5.2 实验课时

4 学时

5.3 实验环境

开发环境: 不限, Visual Studio、Python 等

5.4 任务调度算法(参考)

Step 1: 根据哈斯图结点对应任务集合 S ;

Step 2: $k=1$;

Step 3: 若 $S \neq \emptyset$, 则 $a_k = S$ 的极小元素, $S = S - \{a_k\}$, $k = k + 1$, Go to Step

3, Else Go to Step4;

Step 4: $a_1, a_2, a_3, \dots, a_n$ 即为所求任务执行次序。

5.5 部分节点提示

- 项目执行任务图：可参考教材 P222 图 6-22, 6-23。
- 数据输入：可以在执行页面输入拟参与整除运算的自然数集合的子集, 及该子集的基数大小。
- 生成整除关系哈斯图(如图 5-1), 最佳结果是最小元在下端, 若展示有难度, 也可以相反方向显示。

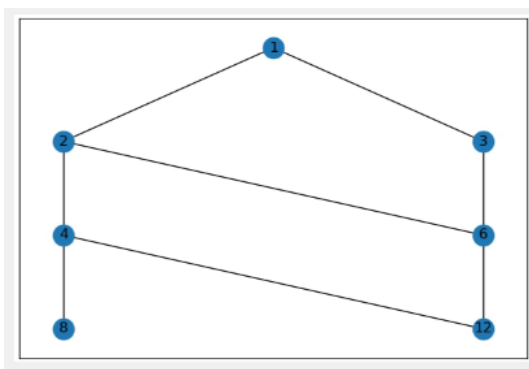


图 5-1 整除关系的哈斯图示意

- 输出可并行执行的任务(以反链展示, 如图 5-2)。

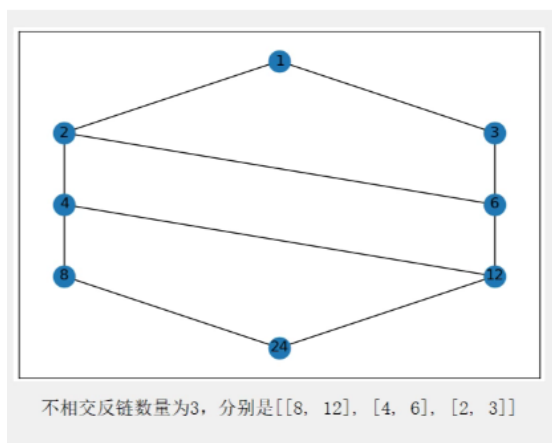


图 5-2 可并行执行的任务输出

第三篇：整数论

实验 6：数论之模运算

6.1 实验要求和目的

- 实验目的：掌握用欧几里德算法实现最大公约数、最小公倍数；掌握 a 模 n 的逆的求解方法；掌握用中国剩余定理求解线性同余方程组。
- 实验要求：
 - (1). 利用欧几里德算法求解两个数的最大公约数。
 - (2). 利用欧几里德算法求解两个数的最小公倍数。
 - (3). 利用中国剩余定理求解线性同余方程组

6.2 实验课时

4 学时

6.3 实验环境

- 开发环境：Visual Studio、Python 等
- 借助控制台或 MFC 实现实验要求中的相关运算（自己搭建）
或借助实验平台：ModularEquations（自制平台）完成

6.4 实验平台简介

本实验提供名为 ModularEquations 的平台，该平台提供基本的选择、输入功能，学生在此基础上实现欧几里德算法实现最大公约数、最小公倍数，扩展的欧几里德算法， a 模 n 的逆的算法，中国余数定理求解线性同余方程组，并进行分析。

- 平台界面：如图 6-1 所示。

- 选 择：通过在“项目”中选择一个项，如图 6-2 所示。
- 输 入：在选择“求线性方程组的解”后，在“线性方程数目”的编辑框里输入方程数目，然后在表格里面输入相应的系数，必须为正整数，最后点击“计算”，如图 6-3，图 6-4 所示。
- 实现算法：

- ◆ 欧几里德算法实现最小公倍数：

```
int CAlgorithm::Least_Common_Multiple ( int a,int b )
```

- ◆ 欧几里德算法实现最大公约数：

```
int CAlgorithm::Greatest_Common_Divisor( int a,int  
b )
```

- ◆ 扩展欧几里德算法：

```
void CAlgorithm::Extended_GCD(__int64 a, __int64 b,  
__int64 &d, __int64t &x, __int64 &y )
```

- 实现函数：

- ◆ a 模 n 的逆：

```
__int64 CAlgorithm::inv(__int64 a, __int64 n )
```

- ◆ 用中国余数定理求解线性同余方程组：

```
void CAlgorithm::solModularEquations( int z,int  
b[],int n[] )
```



图 6-1 主界面

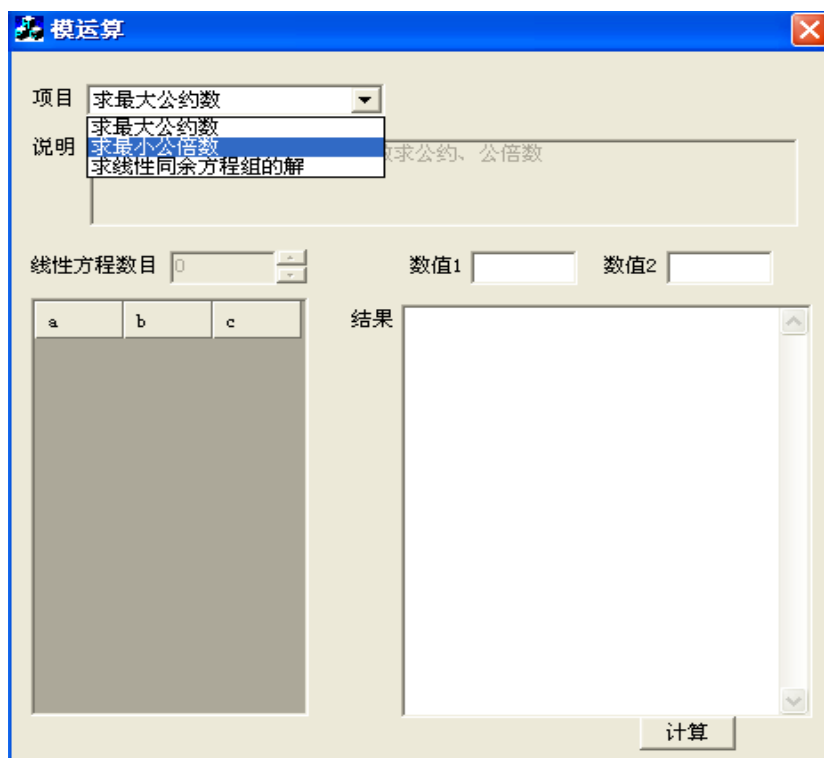


图 6-2 题目选择界面

模运算

项目 求线性同余方程组的解

说明 线性同余方程格式如下：
 $ax \equiv b \pmod{c}$
请先选择线性方程数目 (1~10)，再输入方程的三个系数 a、b、c

线性方程数目 2 数值1 数值2

a	b	c
1	3	4
1	1	9

结果

计算

图 6-3 数据输入界面

模运算

项目 求线性同余方程组的解

说明 线性同余方程格式如下：
 $ax \equiv b \pmod{c}$
请先选择线性方程数目 (1~10)，再输入方程的三个系数 a、b、c

线性方程数目 2 数值1 数值2

a	b	c
1	3	4
1	1	9

结果 您输入的第1个线性同余方程是：
 $1x \equiv 3 \pmod{4}$
您输入的第2个线性同余方程是：
 $1x \equiv 1 \pmod{9}$
线性同余方程组的解是：
 $x = 19 \pmod{36}$

计算

图 6-4 结果输出界面

第四篇：图论

实验 7：最短路径 Dijkstra 与 Floyd-Warshall 算法实现

7.1 实验要求和目的

- 实验目的：理解掌握常用的单源和多源最短路径算法的基本原理和方法。
- 实验要求：
 - (1).借助已有平台基于 Dijkstra 算法求解最短路径，要求图形化显示并文本输出相关路径。
 - (2).借助已有平台基于 Floyd-Warshall 算法求解最短路径，要求图形化显示并文本输出相关路径。

7.2 实验课时

4 学时

7.3 实验环境

本试验提供实验平台

- 开发环境：Visual Studio、Python 等
- 实验平台：ShortestPath（自制平台）

7.4 实验平台简介

本实验提供名为 ShortestPath 的平台，该平台提供添加、移动、删除顶点，添加边，选择源点和终点，提供 Dijkstra 算法或 Floyd-Warshall 算法计算最短路径，并可以输出最短的路径。

- 平台界面：如图 7-1 所示
- 添加顶点：先点击“添加”按钮，然后在左下方空白处添加顶点，如图 7-2 所示

- 添加边：先点击“边”按钮，然后选中边的起点顶点和终点顶点，输入正整数权值，如图 7-3 所示。
- 移动顶点：先点击“移动”按钮，再选中一个顶点，单击左键，然后移动到你想要调整到的位置，单击左键。
- 删除顶点：先点击“删除”按钮，再选中一个顶点，单击左键。
- 选择源点：先点击“源点”按钮，再选中一个顶点作为源点，被选中的源点颜色变为红色，如图 7-4 所示。
- 选择终点，先点击“终点”按钮，再选中一个顶点作为终点，被选中的源点颜色变为蓝色，如图 7-4 所示
- 计算最短路径：先选择一种算法（Dijkstra 算法或者 Floyd 算法），再点击“计算最短路径”，结果如图 7-5、图 7-6 所示
- 实现算法：

- 用 Dijkstra 算法求解最短路径的长度：

```
template <typename Type>  
  
void Graph<Type>::ShortestPath_DIJ()
```

- 用 Dijkstra 算法显示出最短路径：

```
template <typename Type>  
  
void Graph<Type>::Show_ShortestPath_DIJ()
```

- 用 Floyd 算法求解最短路径的长度：

```
template <typename Type>  
  
void Graph<Type>::ShortestPath_FLOYD()
```

- 用 Floyd 算法显示出最短路径

```
template <typename Type>  
  
void Graph<Type>::Show_ShortestPath_Floyd()
```

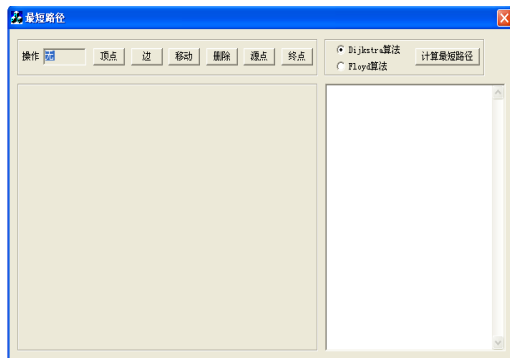


图 7-1 主界面

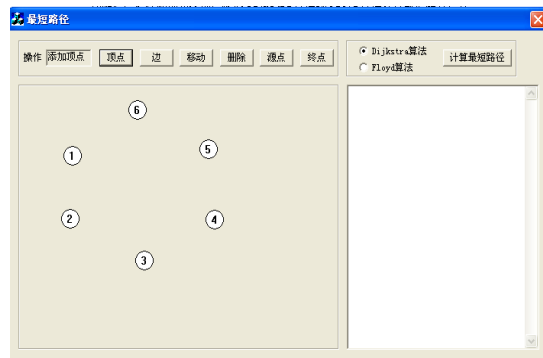


图 7-2 添加顶点界面

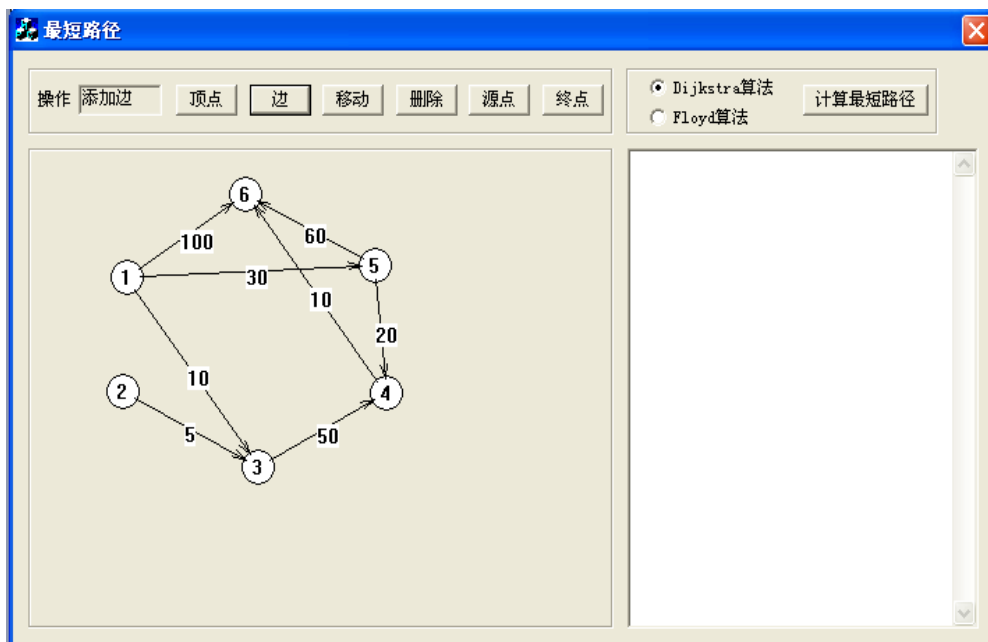


图 7-3 添加边界面

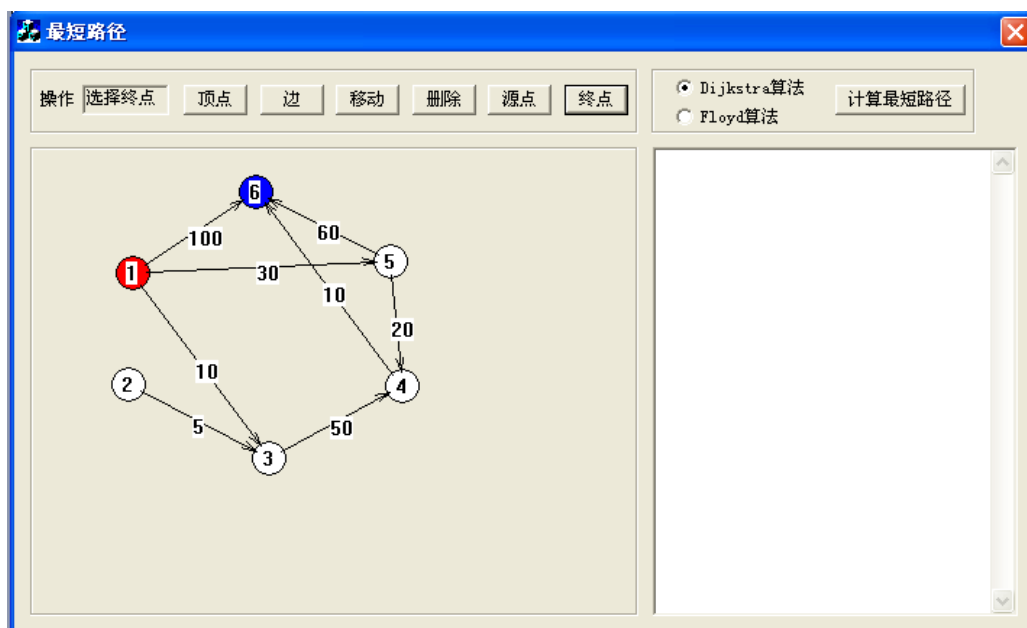


图 7-4 选择源点和终点的界面



图 7-5 Dijkstra 算法计算出最短路径的界面



图 7-6 Floyd-warshall 算法计算出最短路径的界面

实验 8: 欧拉回路的 DFS 或 Fleury 算法求解

8.1 实验要求和目的

- 实验目的:

为求解柯尼斯堡七桥遍历问题, 伟大数学家欧拉创造性地将具体问题抽象为图模型, 创立了图论这一重要数学理论。一笔画问题、中国邮路问题都需要基于图模型进行欧拉回路的求解。

理解欧拉图的定义和求解原理, 并借助 DFS 或 Fleury 算法实现欧拉回(通)路求解。

- 实验要求:

(1). 自行编码实现借助 DFS 算法的欧拉回(通)路求解。

(2). 借助已有平台基于 Fleury 算法求解并展示欧拉路径。

8.2 实验课时

4 学时

8.3 实验环境

本实验提供实验平台

- 开发环境: Visual Studio、Python 等

8.4 实验平台简介

- 平台界面: 如图 8-1 所示。

数据输入: 可以在该界面上借助鼠标, 进行结点和边的输入, 平台实现结点的自动编号和输入结束后图的完整展示。

图的修改: 可在界面上实现结点、边的增加、删除, 以及图的位置调整。

运行 Fleury 算法求解欧拉回(通)路并在界面上用不同颜色标注, 同时在右侧文本框输出路径。

- 借助 Fleury 算法求解欧拉回(通)路的测试样例(如图 8-2)。

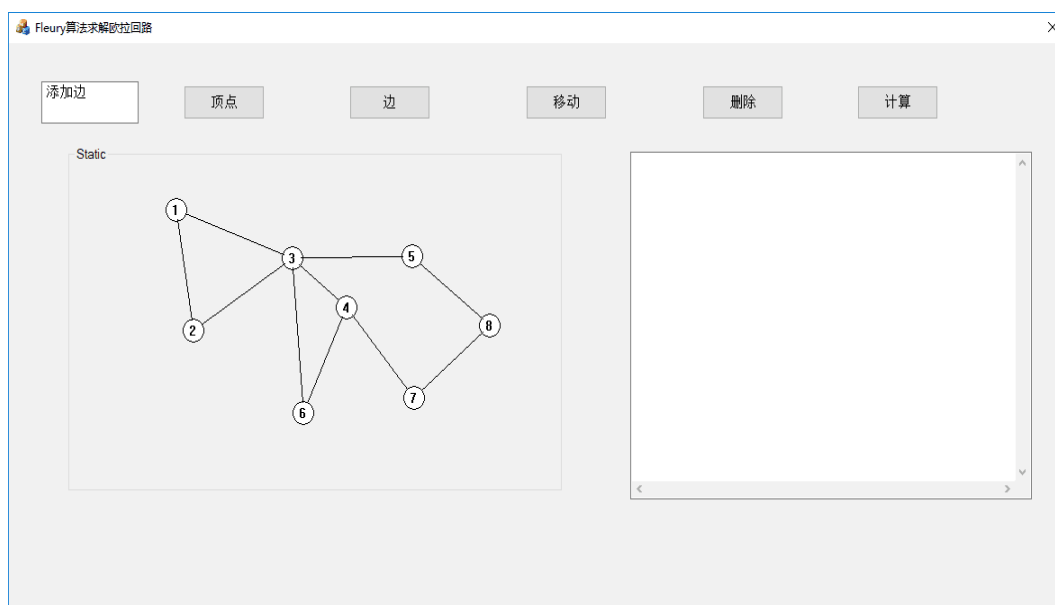


图 8-1 欧拉回（通）路求解主界面

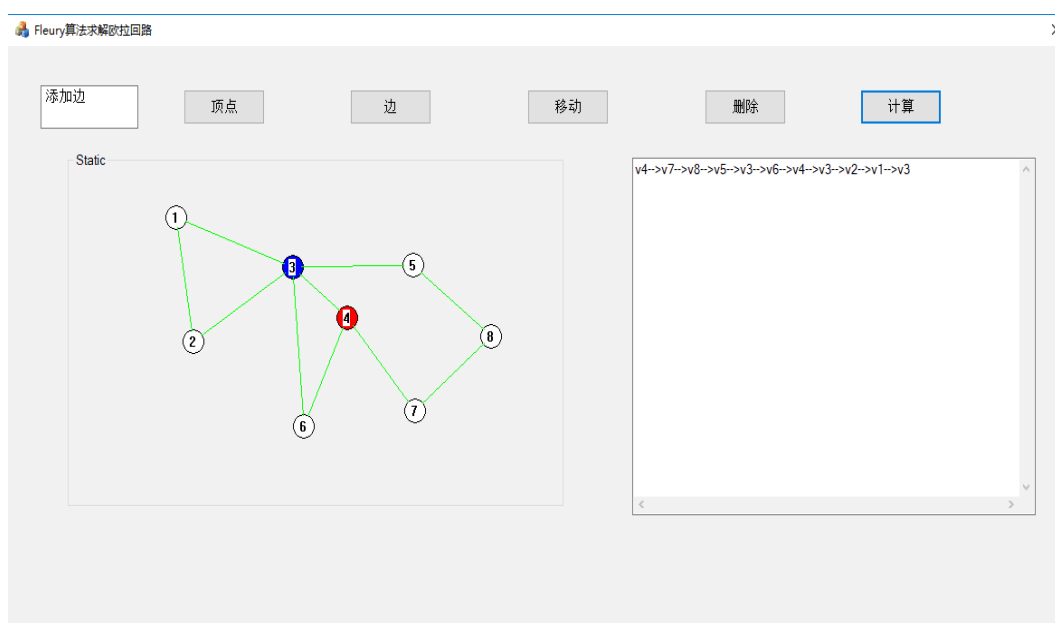


图 8-2 欧拉回（通）路求解测试样例展示

实验 9: 最优树的 Huffman 算法求解

9.1 实验要求和目的

- 实验目的:

Huffman 算法的一个经典应用就是 Huffman 编码, 结合在图论部分对二元树这种图结构的学习, 了解该算法在数据通信编码、图像压缩编码中的最直接应用。主要描述为: 数据通信中经常需要将传送的内容转换成二进制编码, Huffman 编码是一种变长的编码方案, 其核心是使频率越高的码采用越短的编码方式, 同时所有码在解码时不会出现二义性。

理解最优二元树的编码原理, 并借助 Huffman 算法实现最优树求解。

- 实验要求:

(1). 借助已有平台基于 Huffman 算法实现最优二元树的求解。

(2). 或自行编码实现最优二元树的求解及结果展示。

9.2 实验课时

4 学时

9.3 实验环境

本实验提供实验平台

- 开发环境: Visual Studio、Python 等

9.4 实验平台简介

- 平台界面: 如图 9-1 所示。

数据输入: 可以在该界面上进行树的结点个数, 树中结点的按序权值输入, 输入完成时需要进行确认。

图的修改: 可在界面上实现结点个数、按序结点对应权值的修改。

运行 Huffman 算法求解最优二元树并在界面上给出树的图形化展

示, 同时列出每个叶子结点对应的二元前缀码。

- 借助 Huffman 算法求解最优二元树的测试样例 (如图 9-2)。



图 9-1 最优树求解主界面

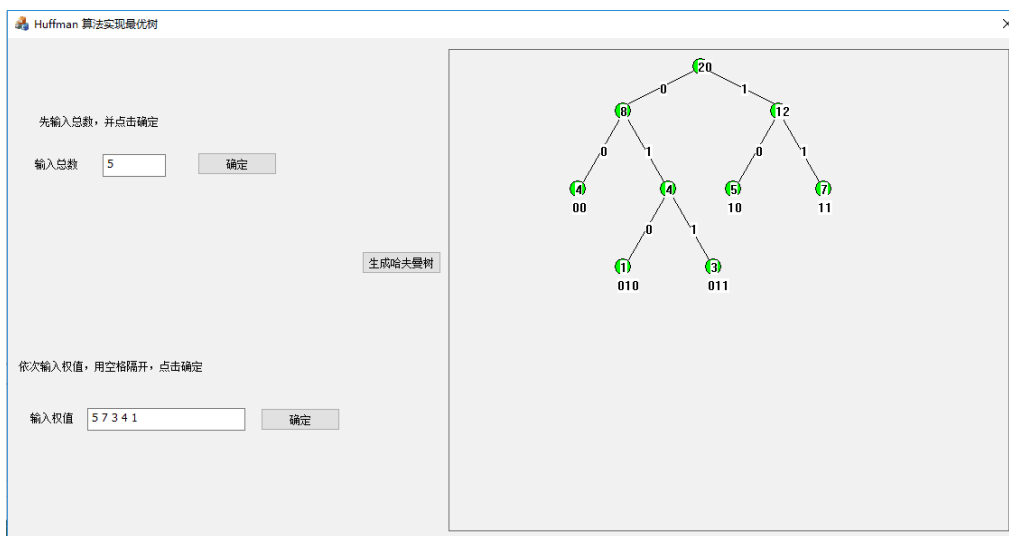


图 9-2 最优树的 Huffman 算法测试样例

第五篇：智能应用

实验 10：城市道路交通信号相位设置与配时系统设计与实现

10.1 实验要求和目的

● 实验目的：

为避免繁忙的城市道路中不同方向和车道交通流之间相互冲突，可以通过在时间上给各个方向和车道分配相应的通行权来解决。对于一组互不冲突的交通流同时获得通行权所对应的信号显示状态，将其称为信号相位，简称相位。交叉路口交通信号的相位设计与配时管理是保障道路交通安全与车辆通行效率的主要方式。

传统的定时信号灯控制方法不能保障高效通行，在高峰时间段容易出现拥堵，在非高峰时段又因等待红灯造成时间上的浪费。本实验的目的是在学习了平面图基础知识及相关着色算法的基础上，根据实际的车道输入及通行规则实现路口的相位设计，并在此基础上基于实时车流量利用模糊控制思想实现动态配时。

● 实验要求：

- (1). 借助已有平台进一步完善交互式交叉路口相位的设计与展示算法，并输出对应的图模型。对原平台在代码、展示度、交互方面需有一定的改进，在实验报告中详细体现。
- (2). 或自行编码实现以上任务。

10.2 实验课时

4 学时

10.3 实验环境

本实验提供实验平台

- 开发环境：Visual Studio、Matlab, Python 等

10.4 实验平台简介

● 平台界面：

数据输入：可以在该界面上对道路交叉路口的车道及拟采用的车道通行规则进行输入，分别对应图模型中的结点和边。

运行结点着色算法求解相位设计方案, 给出设计结果, 并展示不同输入对应的图模型。

- 测试样例展示 (如图 10-1)。

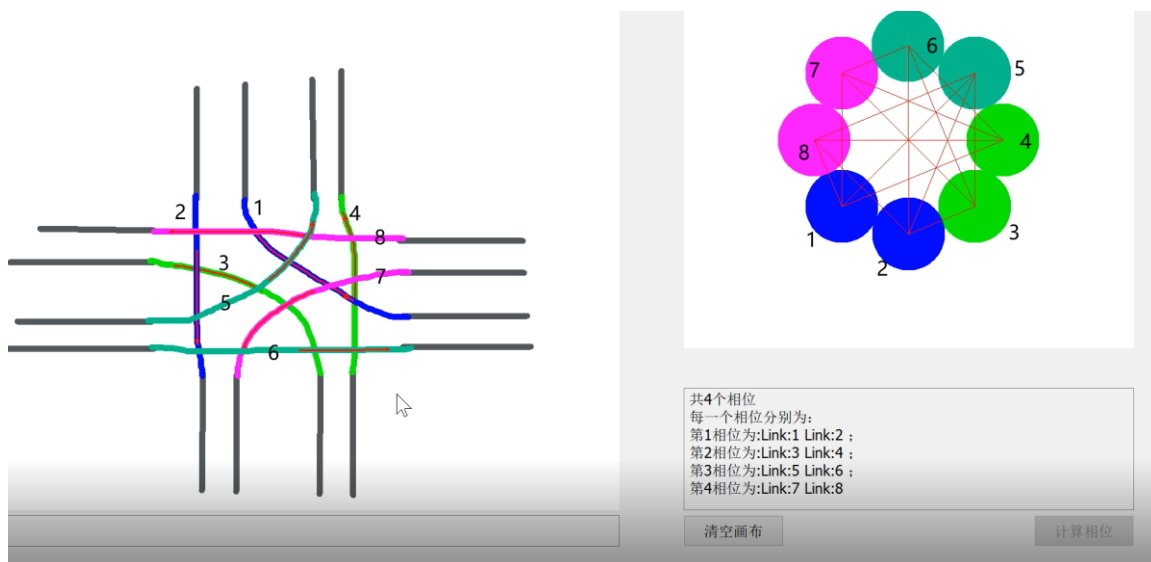


图 10-1 城市道路中某交叉路口的相位设计测试样例

10.5 思考题(选做)

基于实时的车流量数据, 在保障安全的基础之上, 对设定路口信号灯相位进行优化, 依据交叉路口周边环境和实时车流量大小, 设计合理相序, 并综合考虑车辆排队长度实现基于模糊控制思想的交叉路口交通信号灯自适应动态配时, 并给出具体实现方案。

附录 A:

实验 x : 实验名称

姓 名: _____

学 号: _____

班 级: _____

实验地点: _____

实验时间: _____

1 实验目的和要求

2 实验环境和工具

3 实验结果

3.1 算法核心流程图

3.2 关键代码

3.3 运行结果

3.4 结果分析

4 代码与实验结果优化(可选)

5 思考题(可选)

6 实验心得