

# 数据结构课程设计报告

设计题目：教学计划排课系统

学生姓名：叶春亭

专 业：计算机科学与技术

班 级：20-01

学 号：2020215471

指导教师：李培培

完成日期：2021.7.24

# （一）需求和规格说明

**问题描述：**假设任何专业都有固定的学习年限，每学年含两学期，每学期的

时间长度和学分上限均相等。每个专业开设的课程都是确定的，且课程在开设时

间的安排必须满足先修关系。每门课程的先修课程是确定的（可自行设计），可

有任意多门，也可以没有。每门课恰好占一个学期。试根据此问题描述，设计一

个教学计划编制程序

**编程任务：**

1)输入参数包括：学期总数，一学期的学分上限，每门课的课程号（固

定占 8 位的字母数字串）、学分和直接先修课程的课程号；（数据课

程教师提供）

2) 自行设计编排策略，每个学期的学分尽量均衡；

3) 利用拓扑排序求出所有的拓扑序列，查看课程之间的拓扑序列；若无

解，可反馈教学计划编制出错信息；

4) 输出任意一个学期的课程信息，包括课程名称、课程与学分；

5) 按照课程查询任一门课程所属的学期；

6) 采用图形用户界面实现。

注：由于单词中空格不容易辨认，故将空格改为下划线。

# （二）设计

## 1．设计思想

本体是教学计划排课系统，再类的基础上，由于教学计划排课有两个课程中的先修关系，所以我们用到图这个数据结构，并使用拓扑排序来作为排课系统的算法依据，考虑到第六个要求需要用到图形界面实现，我们用到的编程工具为QT，以便实现界面需求，所以大部分之间需要用来研究qt编译器的使用，排课系统主要思路，首先读取文件，获取课程信息包括其先修课，再根据先修课进行拓扑排序，排出课程的拓扑序列，再设计编排序列，我这里设计的策略是：先根据输入的课程数量和学期数量，平均出每一学期的大致课程数，按照每学期课程压力平均原则，，当然还要保证每学期学分数不大于最大学分，输出任意学期课程信息和查询任意门课所属学期则相对简单，只需要在读取课程文件时，保存好并做好标注，以便后续的查找就好了。

## 2. 设计表示

1. 结点结构：

|  |  |  |  |
| --- | --- | --- | --- |
| **结构体名称** | **成员数据类型** | **成员名** | **描述** |
| **Lesson** | QString | LessonName | **课程名字** |
| QString | LessonNum | **课程代码** |
| double | Credit | **学分** |
| vector<int> | PreLessons | **先修课程数组** |
| int | tag | **标志** |
| bool | hasPre | **是否有先修课** |
|  | vector<list<int>> | Adj | **邻接表** |
| vector<int> | inDegree | **入度数组** |
| vector<Lesson> | lessons | **课程数组** |
| int | LessonNum | **课程数量** |
| int | TermsNum | **学期数量** |
| double | MaxCredit | **最高学分** |
| double | AllCredits | **所有学分之和** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | node\* | rear | 尾指针 |

1. 函数：

|  |  |  |
| --- | --- | --- |
| **数据类型** | **函数名称** | **描述** |
| 构造函数 | mainmenu() | Ui窗口类的构造函数用于构造窗口美化等作用 |
| void | **on\_lineEdit\_editingFinished()** | 添加文本框用于用户输入学期总数 |
| void | **on\_lineEdit\_2\_editingFinished()** | 添加文本框用于用户输入最高学分 |
| int | **searchLesson**(QString lessonNum,Graph g) | 用于根据课程代码查找返回课程编号（编号为读取课程文件时从上到下依次的数字顺序编号） |
| void | **GetCin**(Graph& G) | 用于得到两个文本框的值并传入Graph全局变量G |
| void | **on\_pushButton\_5\_clicked** | 这个按钮用来选择电脑中的文件（用鼠标点击的方式）选择文件后将全局变量filestr传入到下一个函数GetFile中，并进行读取文件 |
| void | **GetFile**(Graph& G) | 用于读取文件并将文件中的数据传入G中，这里使用 |
| void | **tpSort**(Graph& G ) | 对图中的课程用前述的编号进行拓扑排序，并将拓扑排序后的结果存入tpvec中的（tpvec需要通过stack进行倒序操作） |
| void | **SortLesson**(Graph& g) | 该函数为对拓扑排序后的课程进行前述排课策略的方法对每个学期进行排课，并且为了后面需要输出每个学期的课程以及查询每个学期课程和查询课程所属学期的方便，在这个函数中，对每个学期的课程用数组封装，以便后续的取用。注：该函数中直接输出排课后的所有学期的课程，后续直接在一个按钮中进行标记实现。 |
| void | **Show\_Term\_Lessons**(Graph& G , int num) | 对上方排课操作时所封装的每学期课程，当用户输入学期数时进行输出封装的课程信息。 |
| void | **Which\_Term**(Graph& G,QString number) | 对输入的课程代码在封装的课程中进行遍历查找，找到时输出对应的学期数 |

## 3. 核心算法

核心算法除给出描述外，必须添加相应的流程图。

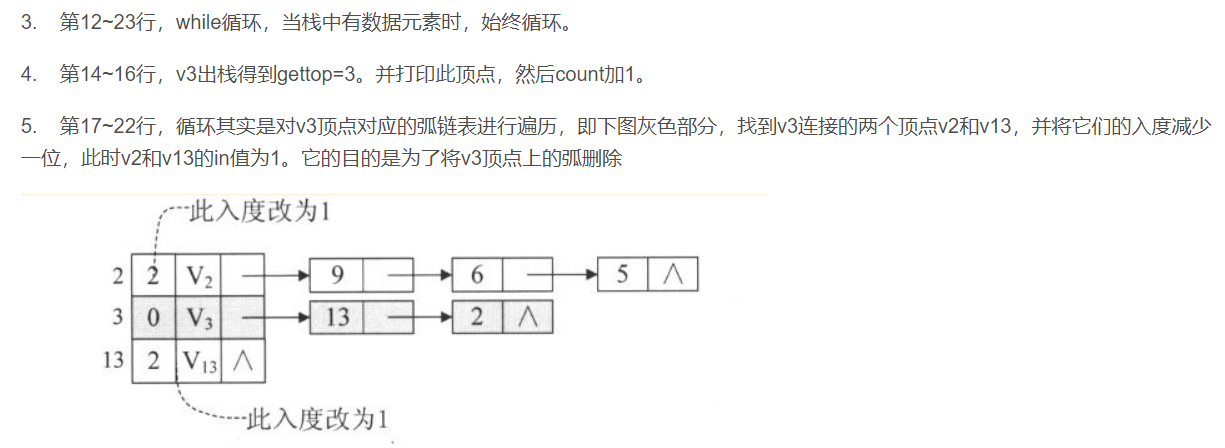
该程序的核心算法为图的拓扑排序，核心算法主要在tpSort函数中，

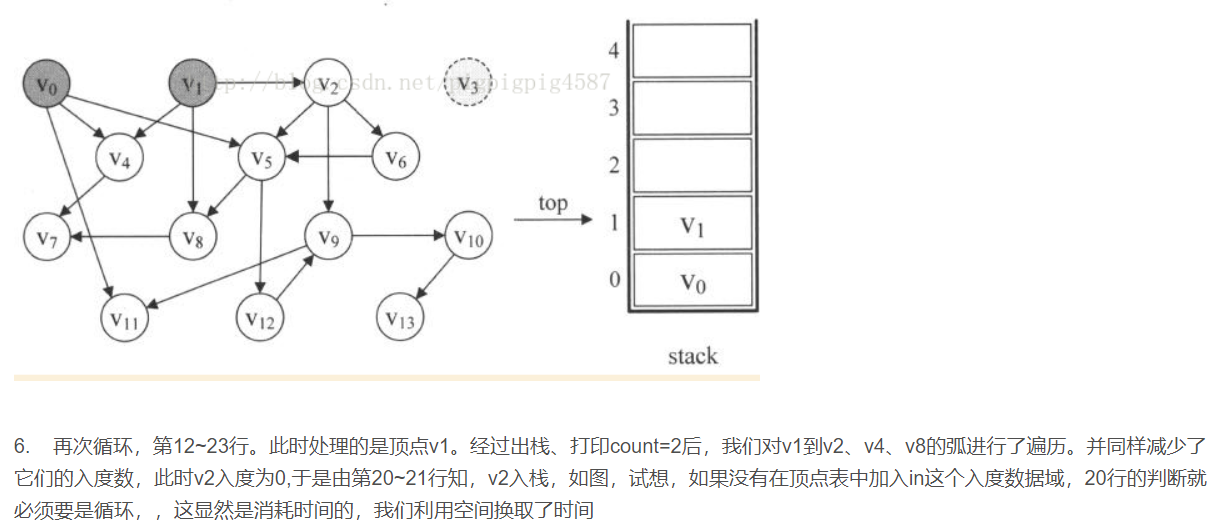
首先用stack存放文件中读取课程的排序编号，然后对邻接表和每个结点的入度进行初始化。

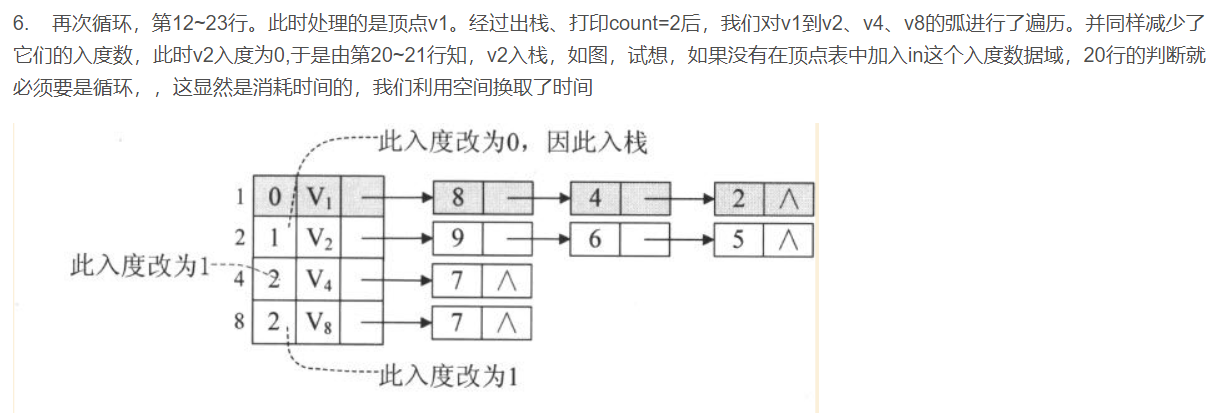
接下来用for循环对读取课程中的含有先修课程的课程（用haspre变量来判断是否含有先修课程）相对应的封装到Graph中的邻接表中，包括入度的操作增加操作，再用一个和for循环对入度为0的课程进行入栈操作然后对栈中的元素进行出栈操作并根据要求封装到vec容器中以便后续加入到tpvec容器，由于此时的顺序是相反的，所以需要后续再用一个栈进行倒序操作，得到拓扑序列后再将其封装至tpvec容器中，完成排序操作。

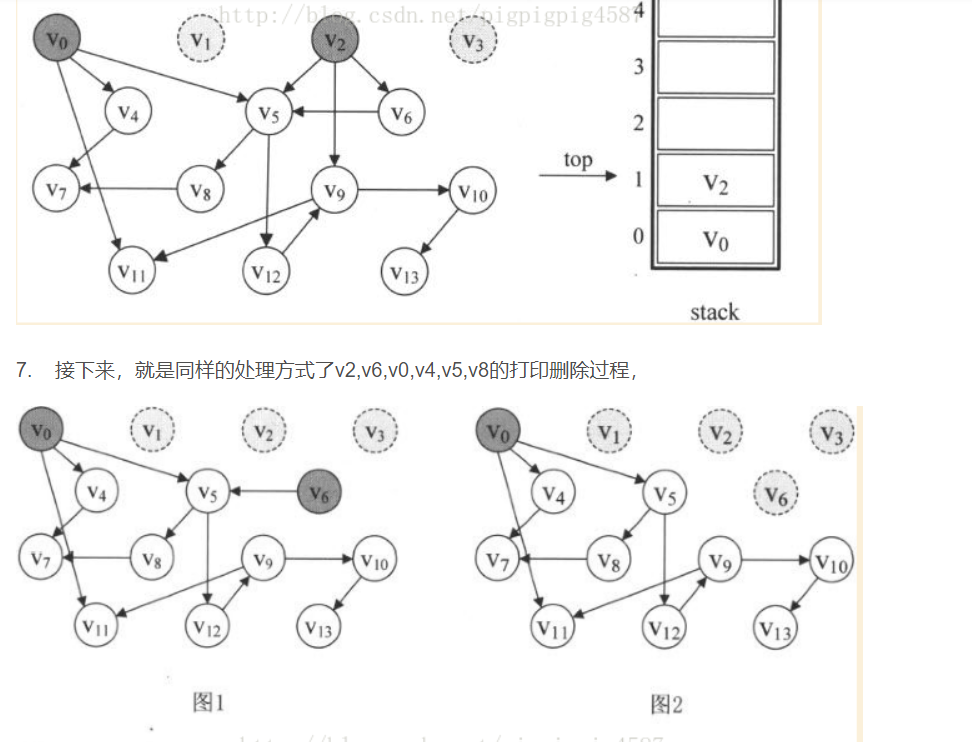
主要的算法图解如下图所示

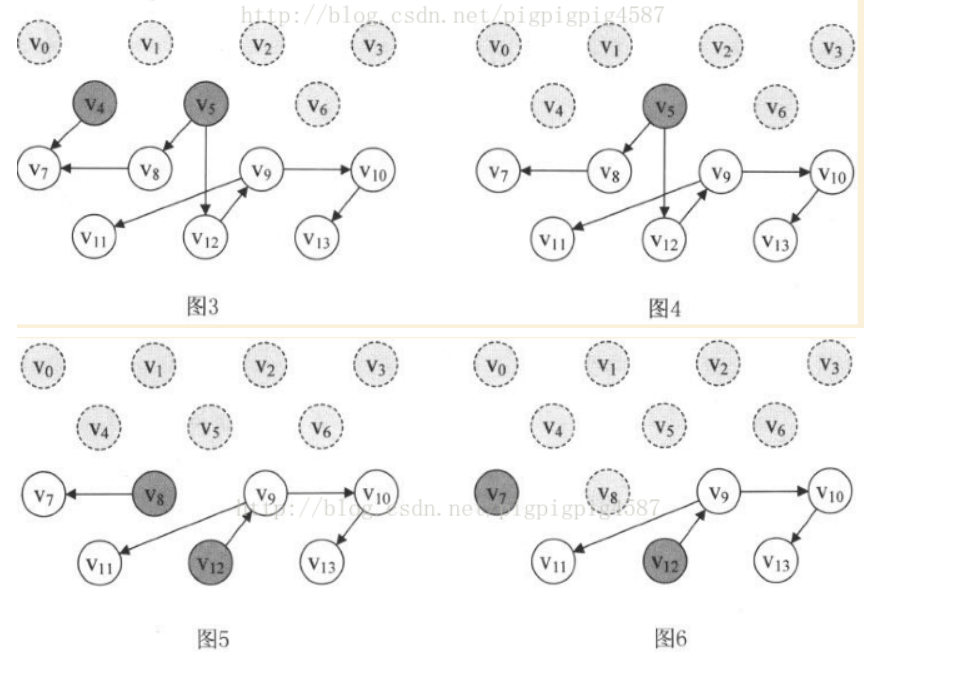
# （三）用户手册





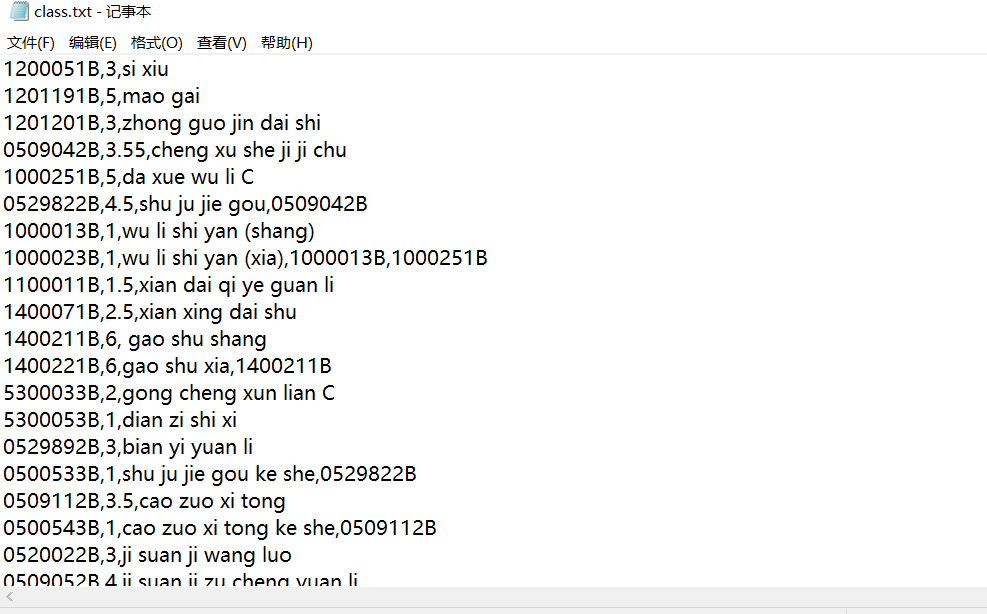


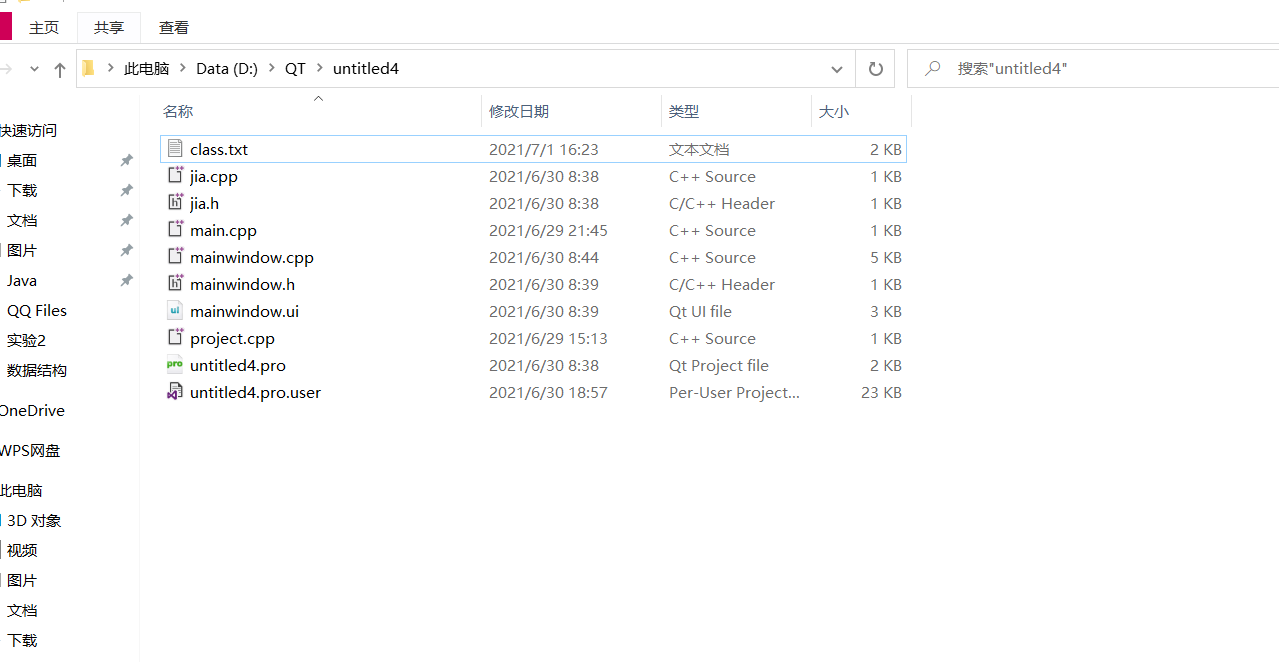




# （三）调试及测试

1.在计算机内存中的某一处编写课程文件以便读取。



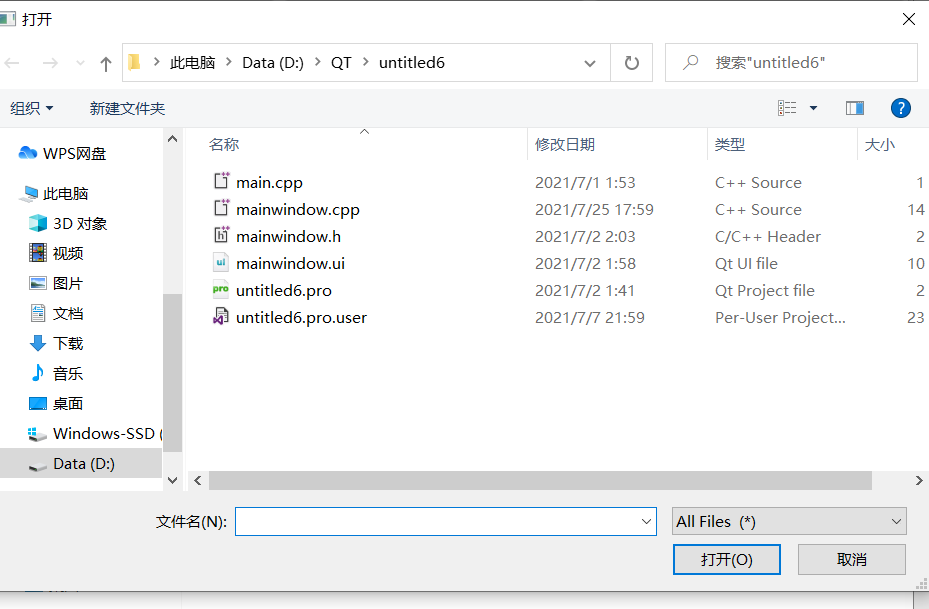


2、运行程序后出现界面窗口

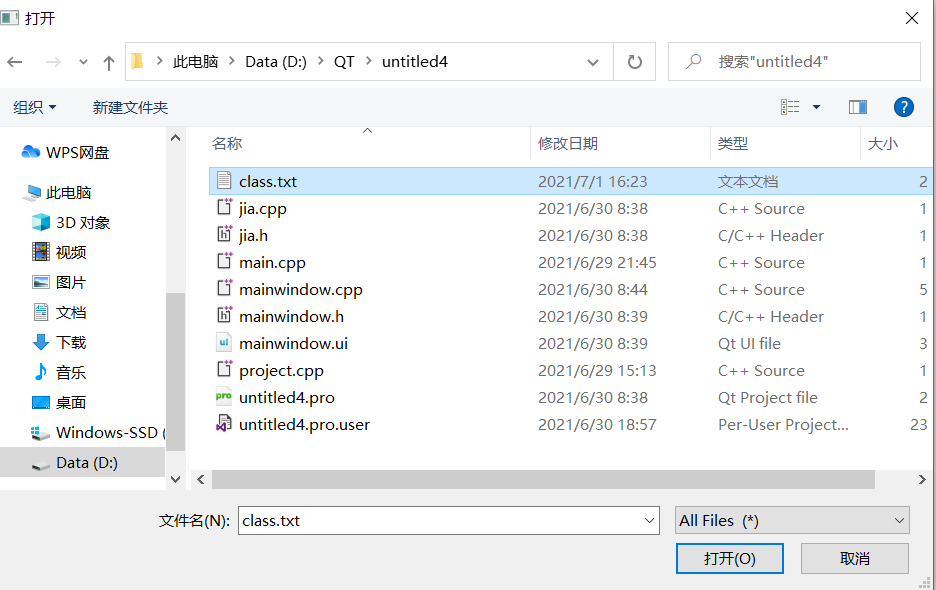


3、首先选取文件，点击左上角的“选择课程文件”。

系统弹出资源管理器



在管理器中找到刚才保存的课程文件class.txt并选择

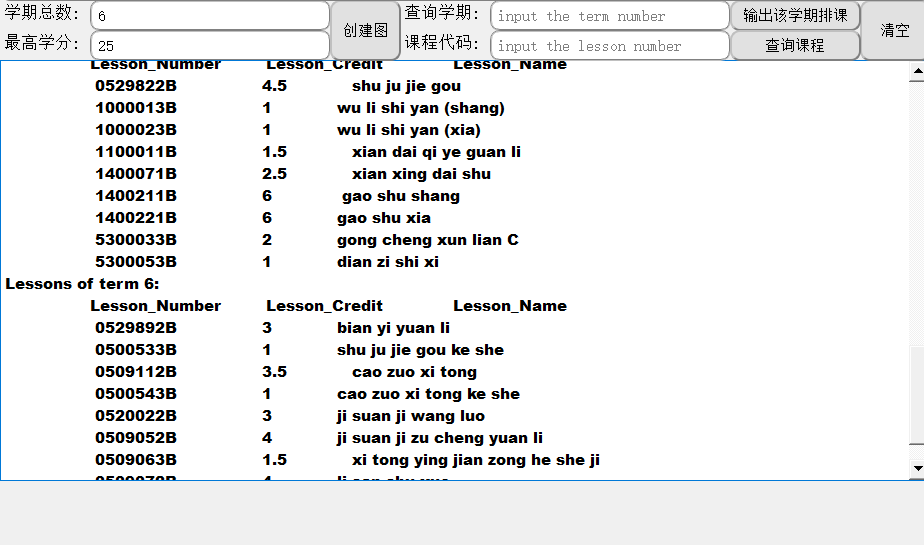
  
选择后在窗口界面输入学期总数和最高学分（这里以6学期，最高25学分为例）输入后点击“创建图”按钮，进行拓扑排序



4、点击生成拓扑序列按钮打印拓扑排序后的课程代码和前面的文件读取序号



为了界面整洁，我加入了清空界面按钮，在最左边，清空后点击输出专业排课（调用lessonsort函数）



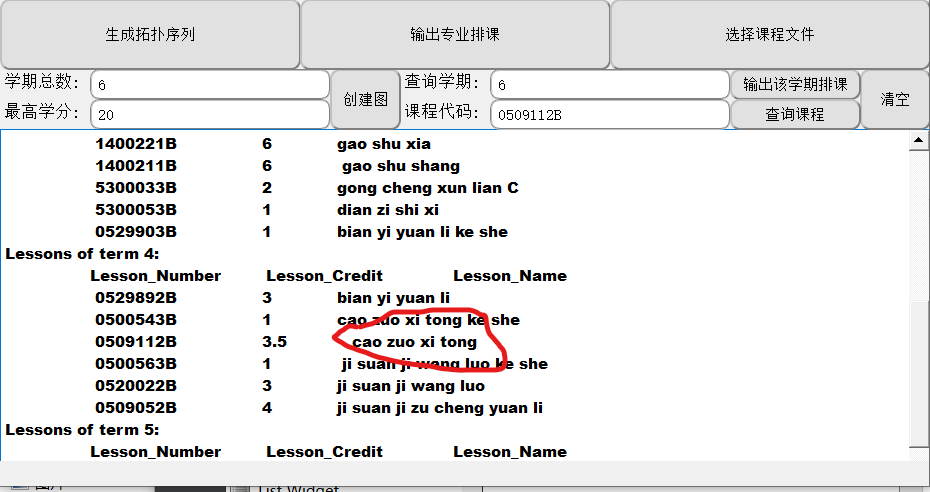
如图所示界面中出现了排课后每个学期的课程信息。

5、清空后实现下一个功能：输出某学期的所有课程（这里以第六学期为例）



可以看到第六学期的课程和上个图片中的一致。

6、清空后实现下一个功能：查询课程所属学期（这里输入了0509112B课程代码，从图中可以看出为操作系统课程，所属学期为4）





程序完成！

## 3. 进一步改进：

（1）原本读取文件时我是按照直接写出文件的名字来读取文件，为了让操作更加简便并且不需要修改代码中文件的位置，通过查询大量资料，我用了QFileDialog类以及file\_dir函数进行通过资源管理器读取文件，并将文件名通过selectedFiles()、replace()、QRegExp()函数转化为读取文件时能识别的形式。

(2)原来的界面枯燥而且很丑，我通过查询资料对界面进行了美化，学会了用QSS美化界面，让案件文本框变得圆润，改变界面颜色，按钮背景等等。。

# (五) 感想

做这道题我觉得最大的难度并不是在数据结构本身，而是如何学会用QT做界面，由于QT编程虽然语言基础还是C++，但是首先想要设计UI界面就得搞懂其中的原理，如何在界面中放置一个按钮，文本框等等插件，放置之后要怎样让他实现你想要的功能，查了许多资料，学会了指令和槽的用法，学会了基础的一些界面插件的使用，之后为了让界面更好看，也是查了许多资料，学会了基本的QSS原理和用法，还有在设计程序的时候由于要输出中文，查了许多的解码方法，最终以UTF-8为解码环境，终于能输出中文的时候不显示乱码了。还有就是在编写程序时，由于有些函数，类和在vs上的有所不同，比如string要换成qstring等，在数据结构部分的话，只有一个拓扑排序需要思考一下，其他的一些功能就是基本的面向对象的应用，总之，完成这份作业，主题算法数据结构部分可能只占了我百分之20的时间精力，占大多数的是如何使用界面实现这个算法。