同濟大學
TONGJI UNIVERSITY

项目说明文档

排课软件

指导老师：张颖

1851009 沈益立

目录

前记

如果说前面8道题目的难度是1-3颗星的话，这道题目的难度能打到四分半甚至五星。它所牵涉到的细小的操作太多，并且逻辑也比较复杂，我debug了很长时间才把样例搞定，虽然可能多多少少有点小问题，但是总体的思路应当是正确的。最后把它放到Ubuntu上编译的时候，发现g++编译和msvc编译有些细微的地方不一样。ss.eof()的意义不同，不能直接读进数据导致排课全空。找这种小bug也花了几十分钟时间，其他的时间加起来也不少了。说真的，上课听起来“排课“这件事情用拓扑排序做起来好像很简单，但是真的实现成代码还是有很多值得注意的地方的。

总的来说，这种有挑战性的题目完成之后说实话还蛮有成就感的，也对码力多多少少有些提升，哈哈！

1.分析

1.1 背景分析

大学的每个专业都要进行排课。假设任何专业都有固定的学习年限，每学年含两学期，每个专业开设的课程都是确定的，而且课程在开设时间的安排必须满足先修关系。每门课程有哪些先修课程是确定的。每门课恰好占一个学期，假定每天上午与下午各有5节课。是在这样的前提下设计一个教学计划编制程序。

1.2 功能分析

1　输入数据包括：个学期所开的课程数（必须使每学期所开的课程数之和与课程总数相等），课程编号，课程名称，周学时数，指定开课学期，先决条件。如指定开课学期为0，表示有电脑自行指定开课学期。

2　如输入数据不合理，比如每学期所开的课程数值和与课程总数不相等，应显示适当的提示信息。

3　用文本文件存储输入数据，并且读入计算机。

4　用文本文件存储产生的各学期的课表。

假设周一至周五上课，每天上10节课，第1大节为第1-2节课，第二大节为第3-5节课，第3大节为第6-7节课，第4大节为8-10节课，在排课时，如一门课程有3节课，则优先安排3节课连续上；如3节课连续无法按排，再优先安排两节课连续上，最后再安排单节课上的情况；如果一门课程需要安排上两天，为教学效果较好，最好不安排在相邻的两天，比如优先安排相隔2天上课，设weekday表示当前安排上课的工作日期，下一次排课的工作日是：weekday=（weekday+2-5）？（weekday+2-5）：（weekday+2）；

很显然，该程序的课程之间先修课程的关系即为偏序关系。对于偏序集，可以构造出一个有向无环图，因而使用拓扑排序可以依次对课程的先后顺序排序。在确定各个课程的先后顺序之后，再通过类似于搜索的方式来将课程塞入课表中：优先三节连排，次选两节连排，再选一节排课。

2.设计

2.1 数据结构设计

拓扑排序的序列不能直接输出，需要容器来储存。因此容易联想到用先进先出（First in First out, FIFO）的队列结构来储存各个节点的处理顺序。在读到该节点时将其塞入拓扑排序的序列，并且处理时依次将队首序列弹出。

对于每节课程而言，其包含了课程名、课号、课时、课程指定学期、先导课程等信息，因此使用一个结构体`Course`来储存这些信息。

对于课表，采用了8 * 5 * 10的矩阵来存储课程信息。

本题未采用STL，而使用了自己设计的链式队列类`LinkedQueue<T>`。

链式队列一定会含有链表节点的元素，因此另外设计了一个数据结构`LNode<T>`作为储存节点的数据结构。

此外，为了加强代码的复用性和可读性，设计了一个`Solution`类作为答案求解的类，实现基本的IO和求解答案、算法实现。

2.2 类结构设计

由于作业的规定，n个小区之间通过电网来连通，对小区的定义为带权值的无向图，设计了一个类Graph来存储这个无向图，并且使用邻接矩阵来作为图的存储形式。由于可能输入的顶点是char类型，因此设计了两个数组：`m_hash[256]`将字符(char)转化为顶点在邻接矩阵中的真实位置(int)、`m_charHash[101]`将顶点在邻接矩阵中的真实位置(int)转化为它的字符形式(char)。

定义了一个结构体：课程结构`Course`，用于存储每节课程的课程名、课号、课时、课程指定学期、先导课程等信息

定义了一个类：节点类`LNode`，作为链式队列的节点。

定义了一个类：链式队列`LinkedQueue`，可以将拓扑排序的序列存储下来。

定义了一个类：`Solution`，用于处理IO、调用算法和存放各种指针，避免使用全局变量。

2.3 成员和操作设计

链式节点类`LNode`

公有操作：

```
LNode() {};                     //默认无参数构造函数
LNode(int val, LNode<T>* next)  //带值和下一个指针域的构造函数
        :m_val(val), m_next(next) {};
```

公有成员：

```
T m_val;                //节点值
LNode<T>* m_next;       //下一个指针域
```

链式队列类`LinkedQueue<T>`

公有操作：

```
LinkedQueue();          //默认的构造函数
~LinkedQueue();         //析构函数，执行所有链表节点内存的释放
void pushBack(T& val);  //推入元素
T popFront();           //弹出元素
T getFront();           //得到队首元素的值
bool isEmpty();         //返回该队列是否为空的布尔值
void makeEmpty();       //清空队列，释放内存
void printQueue();      //将队列的值输出到屏幕上
```

私有成员:

```
LNode<T>* m_rear = nullptr; //队尾指针，默认为空
LNode<T>* m_front = nullptr;//队头指针，默认为空
```

解决方案类`Solution`

公有操作:

```
Solution();

void loadFile();                            //从文件中读取课程信息
void topoSort();                            //进行拓扑排序

void saveFile(ostream& fs);                 //将排好的课表输出到文件里
void arrangeCourse(int x);                  //将第x门课放入课表中
int put3(int curTerm, int& curDay, int i);  //连排3门课
int put2(int curTerm, int& curDay, int i);  //连排两门课
int put1(int curTerm, int& curDay, int i);  //连排一门课

void outputCourse();                        //将读取到的课程输出至屏幕
```

公有成员

```
int m_courseNum;                            //课程数量
int m_courseTable[8][5][10];                //container of result
Course* m_courseArr = nullptr;              //课程数组
```

课程结构体`Course`

公有操作:

```
struct Course {
    string id = "";                         //课程id
    string name = "";                       //课程名称
    int period = 0;                         //课程学时
    int term = 0;                           //指定开课学期
    string prevCourses[10] = {};            //前置课程数组
    int prevNum = 0;                        //前序课程数量
    bool termLocked = 0;                    //是否指定学期的标志位
};
```

公有成员

```
int m_courseNum;                            //课程数量
int m_courseTable[8][5][10];                //container of result
Course* m_courseArr = nullptr;              //课程数组
```

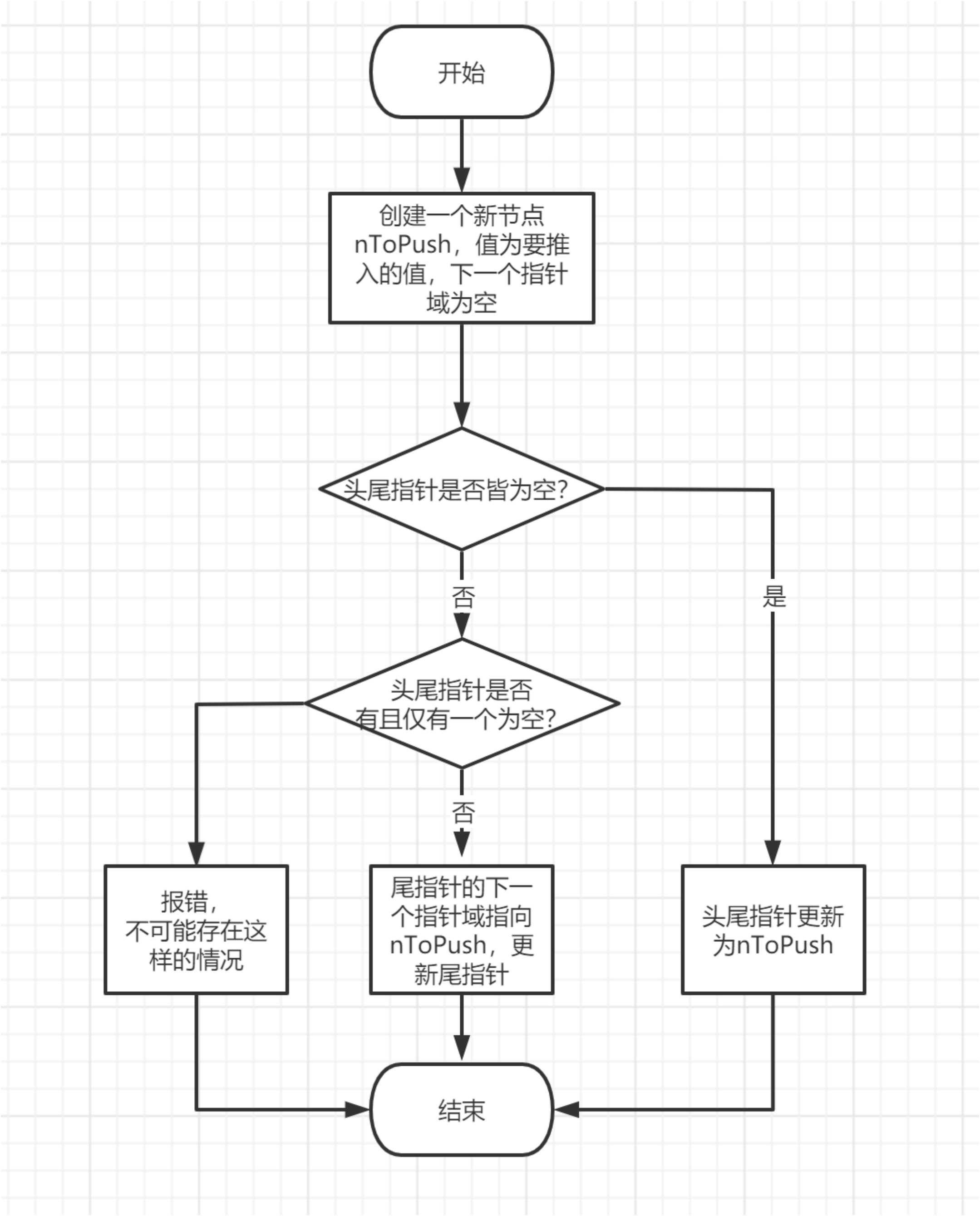
2.4 系统设计

程序运行后，系统会自动创建一个解决方案的实例化对象`sol`，随后调用`loadFile()`，对于根目录下的testpoint文件进行读取操作，并向`m_courseArr[]`中写入这些数据。随后，`sol`调用`topoSort()`，对课程进行拓扑排序并且将其放入课表中。最后，`sol`调用`saveFile(ofs)`，将排好序的课表按照指定的格式输出到`res`文件中。

3.实现

3.1 队列压入的实现

3.1.1 队列压入功能流程图

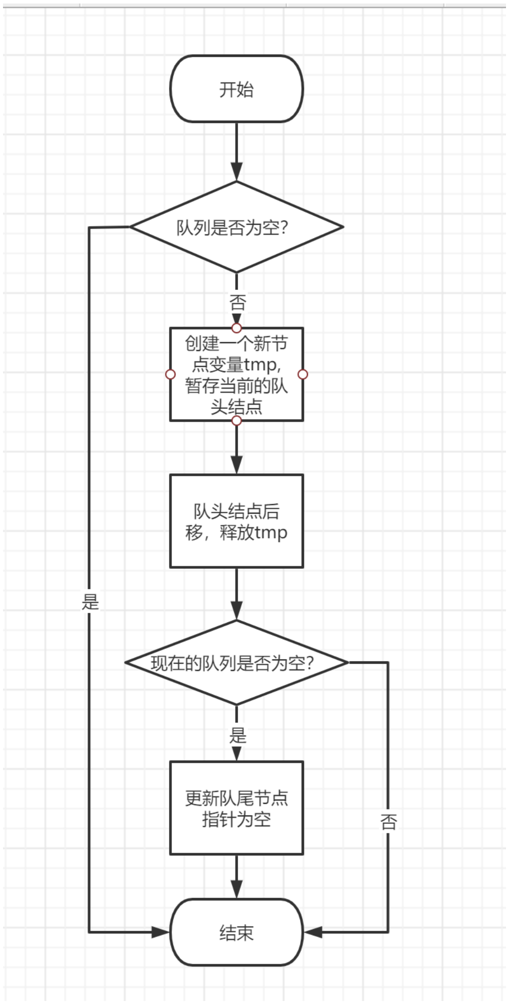

开始
创建一个新节点nToPush，值为要推入的值，下一个指针域为空
头尾指针是否皆为空?
否
是
头尾指针是否有且仅有一个为空?
否
报错，不可能存在这样的情况
尾指针的下一个指针域指向nToPush，更新尾指针
头尾指针更新为nToPush
结束

3.1.2 队列压入功能核心代码实现

```
template<class T>
void LinkedQueue<T>::pushBack(T& val) {
    LNode<T>* nToPush = new LNode<T>(val, nullptr);
    if (!m_rear && !m_front) {//空集判定，若空则将头尾指针都设为其值
        m_rear = nToPush;
        m_front = nToPush;
        return;
    }
    if (!m_rear || !m_front) {
        cerr << "指针错误" << endl;
        return;
    }
    m_rear->m_next = nToPush;
    m_rear = nToPush;
    return;
}
```

3.2 队列弹出功能的实现

3.2.1 队列弹出功能流程图

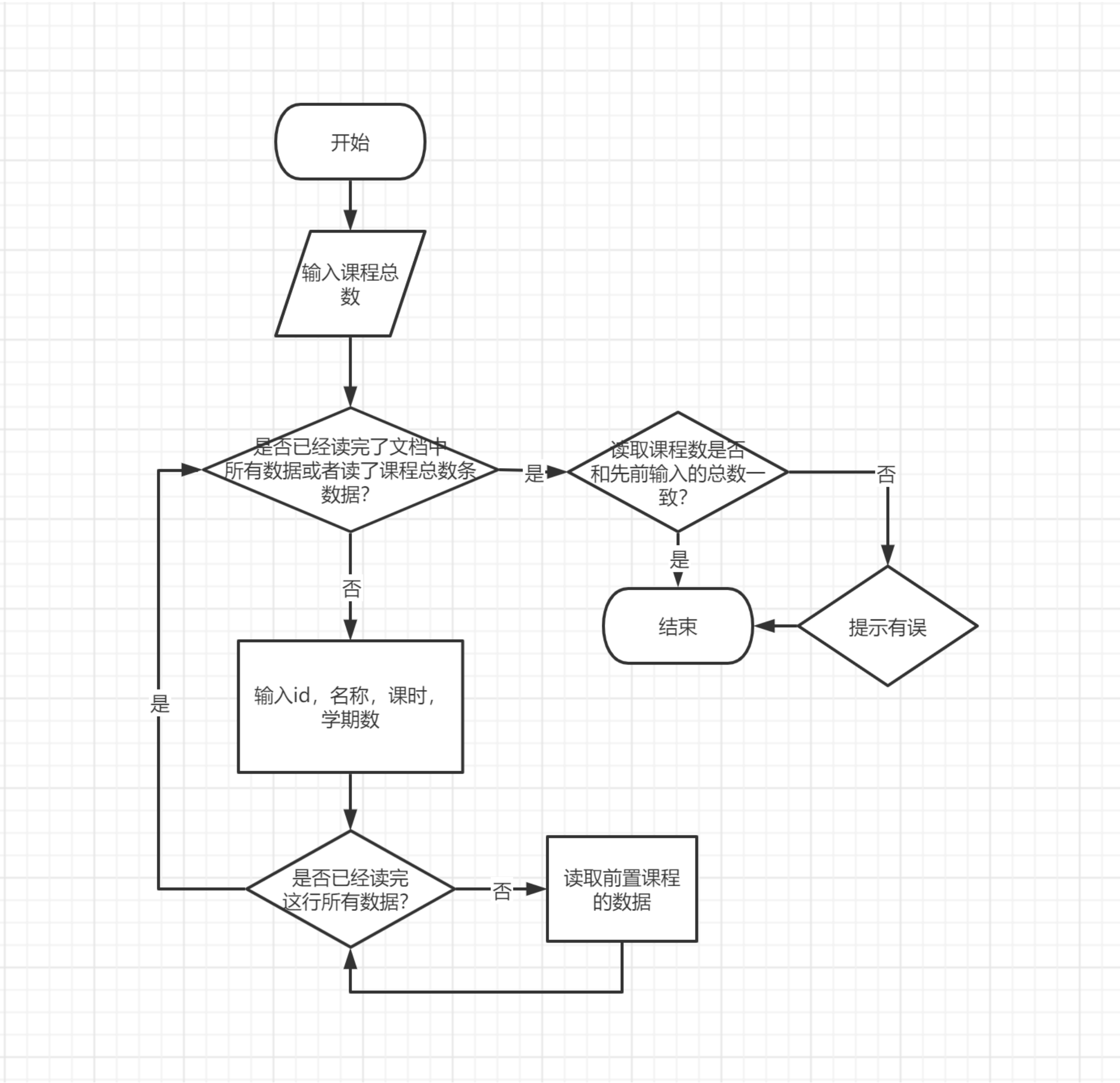


3.2.2 队列弹出功能核心代码实现

```
template<class T>
T LinkedQueue<T>::popFront(){
    if (isEmpty()) {
        cerr << "空队列，不可弹出" << endl;
        return T();
    }
    LNode<T>* tmp = m_front;
    T retVal = tmp->m_val;
    m_front = m_front->m_next;
    if (m_front == nullptr) {//空集情况判定
        m_rear = nullptr;
    }
    delete tmp;//Tips： 内存泄漏
    return retVal;
}
```

3.3 读取信息功能的实现

3.3.1 读取信息功能流程图

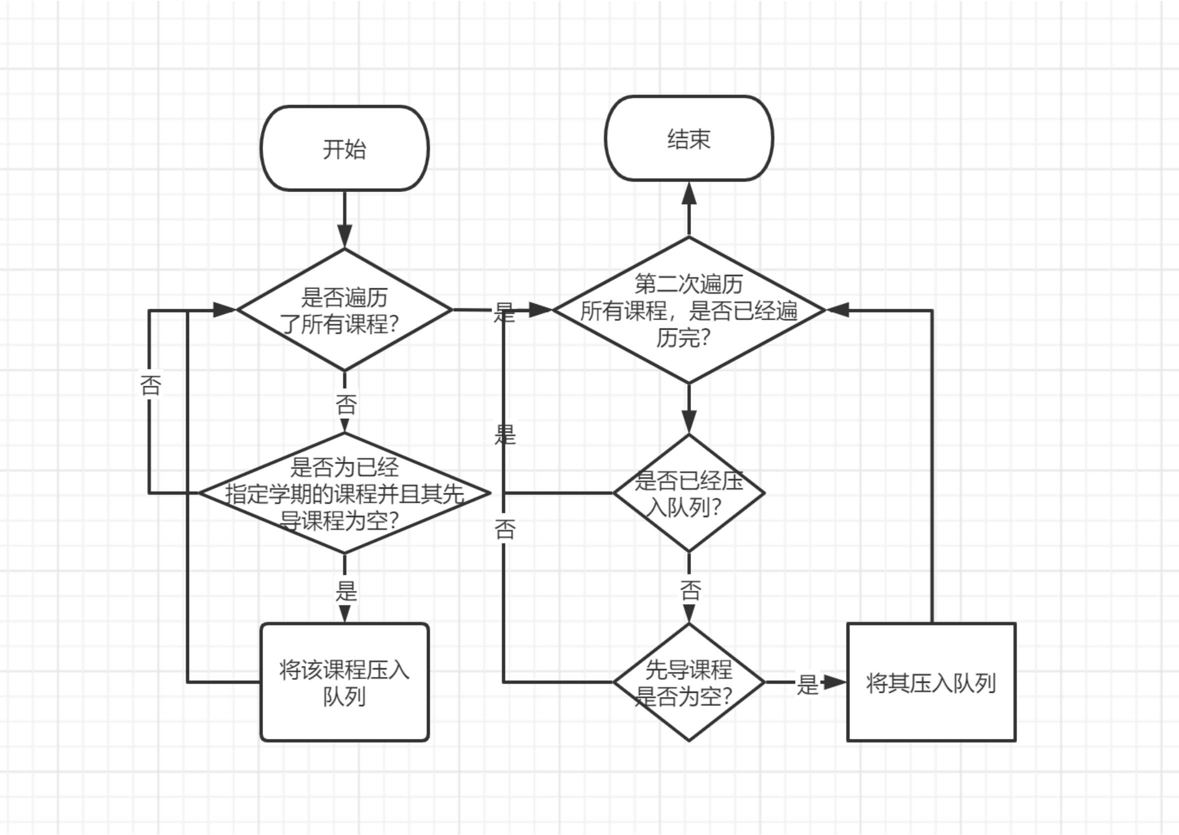

开始
输入课程总数
是否已经读完了文档中所有数据或者读了课程总数条数据?
是
读取课程数是否和先前输入的总数一致?
否
是
结束
提示有误
否
输入id，名称，课时，学期数
是
是否已经读完这行所有数据?
否
读取前置课程的数据

3.3.2 读取信息功能核心代码实现

```
void Solution::loadFile() {
    string fileName = "testpoint";

    ifstream ifs;
    ifs.open(fileName);
    ifs >> m_courseNum;
    string tmp;
    getline(ifs, tmp);
    m_courseArr = new Course[m_courseNum + 1];
    int i = 0;
    for (i = 0; !ifs.eof(); i++) {
        string strLine;
        getline(ifs, strLine);
        stringstream ss(strLine);

        ss >> m_courseArr[i].id >>
            m_courseArr[i].name >>
            m_courseArr[i].period >>
            m_courseArr[i].term;
        if (m_courseArr[i].term > 0) {
            m_courseArr[i].termLocked = 1;
            m_courseArr[i].term--;
        }
        else {
            m_courseArr[i].termLocked = 0;
        }
        int k = 0;
        while (!ss) {
            string prev;
            ss >> prev;
            m_courseArr[i].prevCourses[k++] = prev;
        }
        m_courseArr[i].prevNum = k;
    }
    ifs.close();

    if (i != m_courseNum) {
        cerr << "Input data doesn't match the total number of courses.";
        m_courseNum = i;
    }
    //outputCourse();
}
```

3.4 初始队列构造的实现

3.4.1 初始队列构造功能流程图

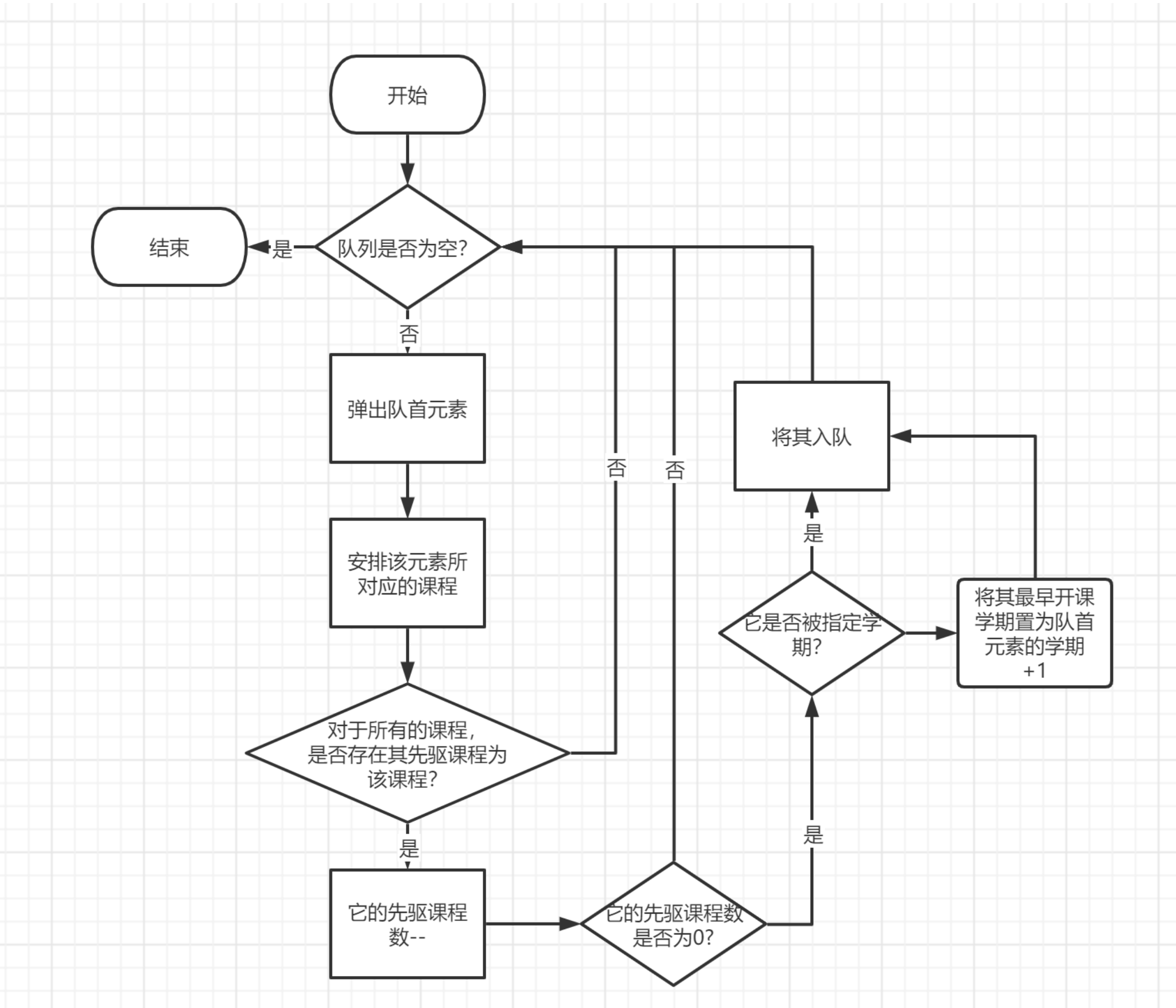


3.4.2 初始队列构造核心代码实现

```
LinkedQueue<int> q;

int* preCount = new int[m_courseNum];
//优先安排锁定学期的课程

for (int i = 0; i < m_courseNum; i++) {
    if (m_courseArr[i].termLocked &&
m_courseArr[i].prevNum == 0) {
        q.pushBack(i);
    }
}
//先入队没有任何前驱的元素
for (int i = 0; i < m_courseNum; i++) {
    preCount[i] = m_courseArr[i].prevNum;
    if (m_courseArr[i].termLocked &&
m_courseArr[i].prevNum == 0) {
        continue;
    }
    if (!preCount[i]) {
        q.pushBack(i);
    }
}
```

3.5 排课总体逻辑的实现

3.5.1 排课总体逻辑流程图

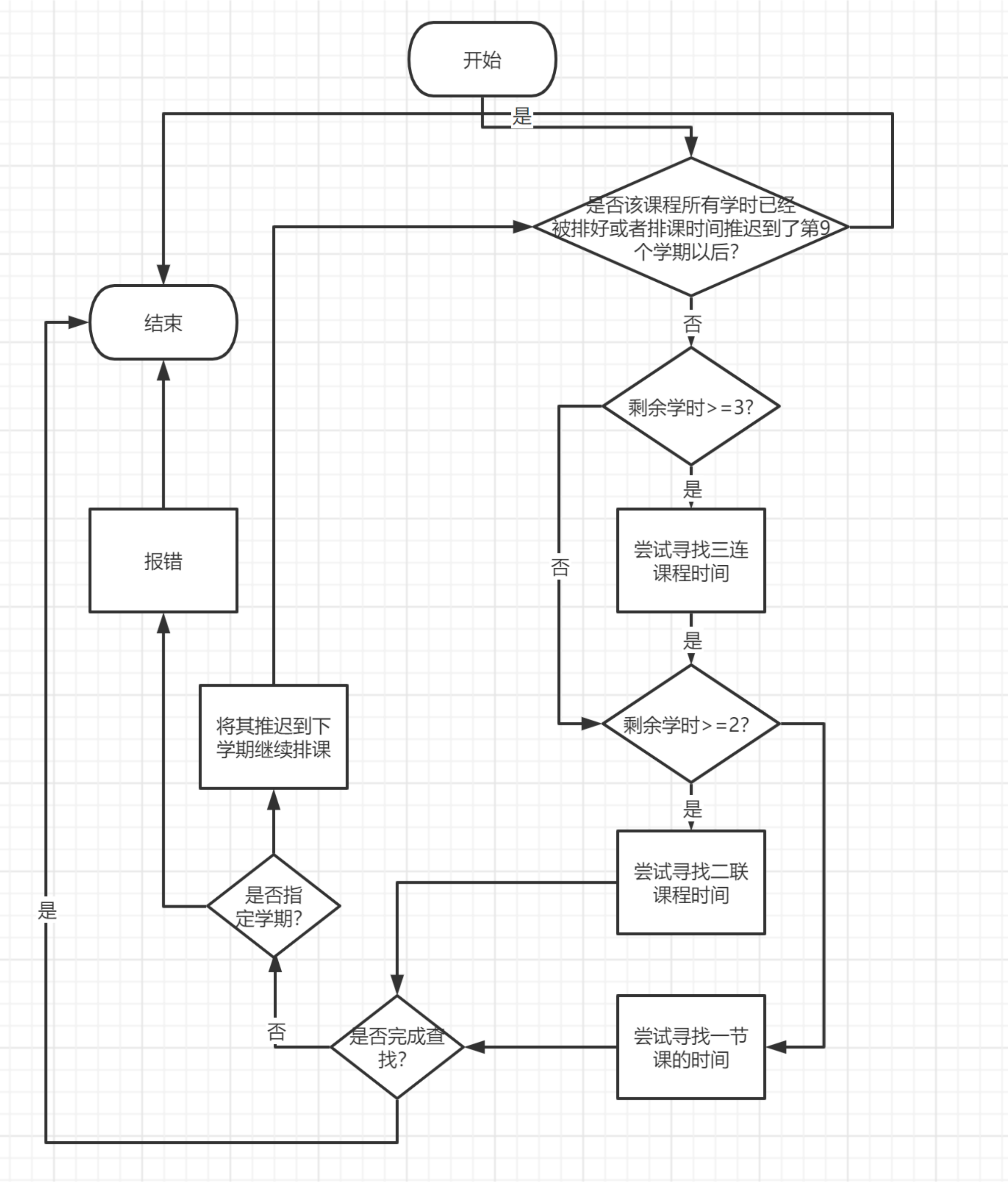


3.5.2 排课总体逻辑核心代码实现

```
while (!q.isEmpty()) {
    auto x = q.popFront();
    arrangeCourse(x);// sure to succeed
    for (int i = 0; i < m_courseNum; i++) {
        for (int j = 0; j < m_courseArr[i].prevNum; j++) {
            if (m_courseArr[i].prevCourses[j] ==
m_courseArr[x].id) {
                preCount[i]--;
                if (preCount[i] == 0) {


                    if (m_courseArr[i].term <
m_courseArr[x].term + 1) {
                        if (!m_courseArr[i].termLocked) {
                            m_courseArr[i].term =
m_courseArr[x].term + 1; //置后一学期
                        }
                    }
                    q.pushBack(i);
                }
            }
        }
    }

}
```

3.6 每节课排课逻辑的实现

3.6.1 每节课排课逻辑流程图



3.6.2 每节课排课核心代码实现

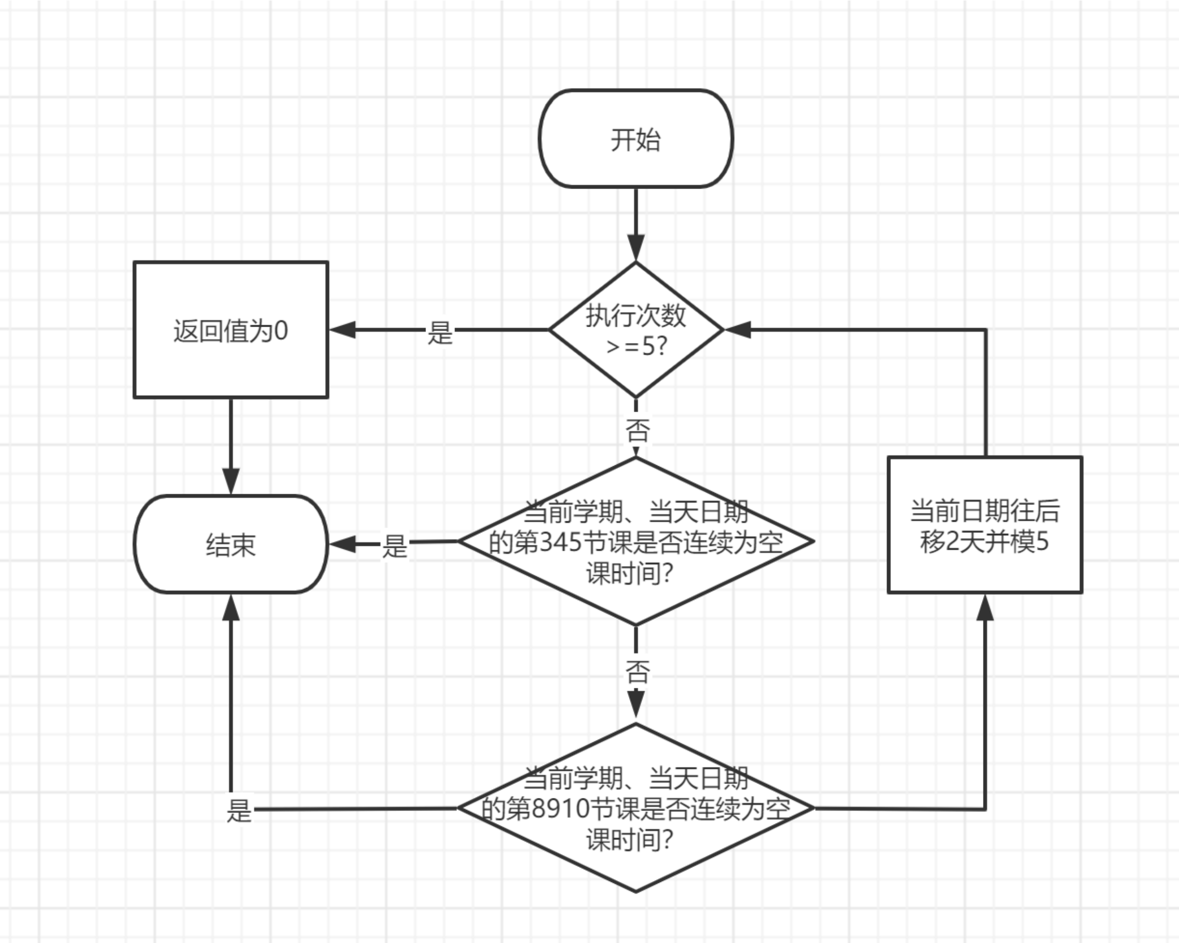
```
void Solution::arrangeCourse(int x) {
    int curDay = 0;
    int alreadyPut = 0;
    auto remainToPut = m_courseArr[x].period - alreadyPut;
    int loopTime = 0;
    if (m_courseArr[x].termLocked) {
        while (alreadyPut < m_courseArr[x].period &&
m_courseArr[x].term < 8) {
```

```
            remainToPut = m_courseArr[x].period -
alreadyPut;
            int flag = 0;
            if (remainToPut >= 3) {
                flag = put3(m_courseArr[x].term, curDay,
x);

            }
            else if (remainToPut == 2) {
                flag = put2(m_courseArr[x].term, curDay,
x);
            }
            else {
                flag = put1(m_courseArr[x].term, curDay,
x);
            }

            alreadyPut += flag;
            loopTime++;
            if (loopTime > 15) {
                if (remainToPut) {
                    cerr << "第" << x + 1 << "门课不可排进课
表中。" << endl;
                }
            }
        }
    }
    else {
        while (alreadyPut < m_courseArr[x].period &&
m_courseArr[x].term < 8) {
            remainToPut = m_courseArr[x].period -
alreadyPut;
            int flag = 0;
            if (remainToPut >= 3) {
                flag = put3(m_courseArr[x].term, curDay,
x);

            }
            else if (remainToPut == 2) {
                flag = put2(m_courseArr[x].term, curDay,
x);
```

```
            }
            else {
                flag = put1(m_courseArr[x].term, curDay, x);
            }

            alreadyPut += flag;

            if (!flag) {//寻找下个学期的机会
                if (!m_courseArr[x].termLocked) {
                    m_courseArr[x].term++;
                    curDay = 0;
                }
            }
        }
    }
    remainToPut = m_courseArr[x].period - alreadyPut;

    if (remainToPut) {
        cerr << "第" << x + 1 << "门课不可排进课表中。" << endl;
    }
}
```

3.7 寻找三节课排课机会功能的实现

3.7.1 寻找三节课排课机会功能流程图

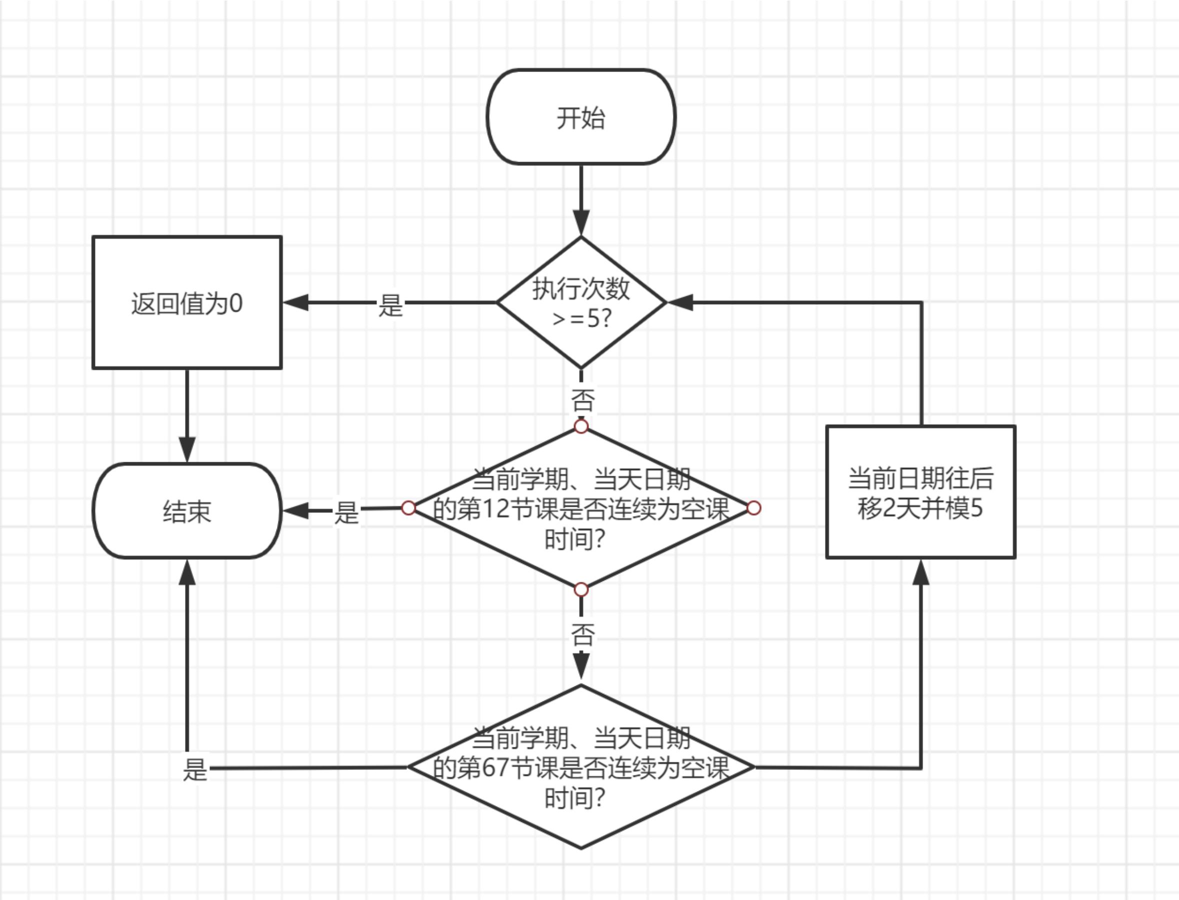

开始
执行次数>=5?
是
返回值为0
否
当前学期、当天日期的第345节课是否连续为空课时间?
是
结束
否
当前学期、当天日期的第8910节课是否连续为空课时间?
是
当前日期往后移2天并模5

3.7.2 寻找三节课排课机会核心代码实现

```
int Solution::put3(int curTerm, int& curDay, int i) {
    for (auto t = 0; t < 5; t++) {

        //234连排
        if (m_courseTable[curTerm][curDay][2] == -1
            && m_courseTable[curTerm][curDay][3] == -1
            && m_courseTable[curTerm][curDay][4] == -1) {

            m_courseTable[curTerm][curDay][2] = i;
            m_courseTable[curTerm][curDay][3] = i;
            m_courseTable[curTerm][curDay][4] = i;

            curDay += 2;
            curDay %= 5;
            return 3;
        }
        //789连排
        if (m_courseTable[curTerm][curDay][7] == -1
            && m_courseTable[curTerm][curDay][8] == -1
            && m_courseTable[curTerm][curDay][9] == -1) {
            m_courseTable[curTerm][curDay][7] = i;
            m_courseTable[curTerm][curDay][8] = i;
            m_courseTable[curTerm][curDay][9] = i;
            curDay += 2;
            curDay %= 5;
            return 3;
        }

        curDay += 2;
        curDay %= 5;
    }
    return 0;
}
```

3.8 寻找两节课排课机会功能的实现

3.8.1 寻找两节课排课机会功能流程图

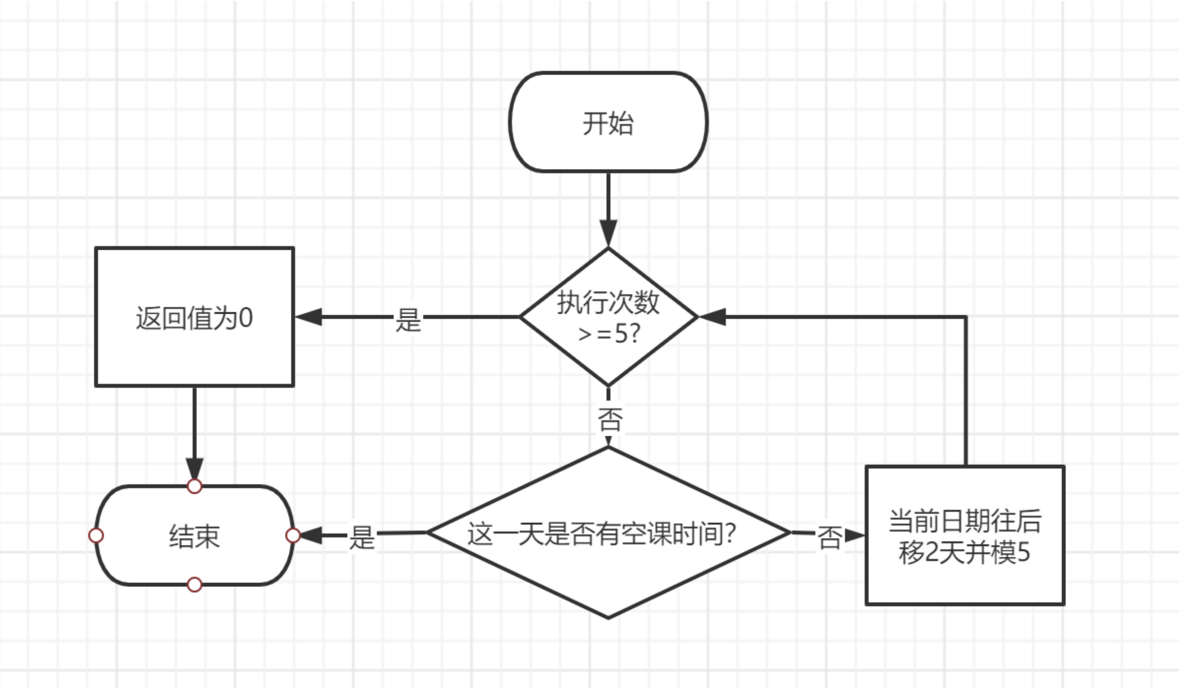


3.8.2 寻找两节课排课机会核心代码实现

```
int Solution::put2(int curTerm, int& curDay, int i) {
    for (auto t = 0; t < 5; t++) {

        //01连排
        if (m_courseTable[curTerm][curDay][0] == -1
            && m_courseTable[curTerm][curDay][1] == -1) {
            m_courseTable[curTerm][curDay][0] = i;
            m_courseTable[curTerm][curDay][1] = i;


            curDay += 2;
            curDay %= 5;
            return 2;
        }
        //56连排
        if (m_courseTable[curTerm][curDay][5] == -1
            && m_courseTable[curTerm][curDay][6] == -1) {
            m_courseTable[curTerm][curDay][5] = i;
            m_courseTable[curTerm][curDay][6] = i;

            curDay += 2;
            curDay %= 5;
            return 2;
        }
        curDay += 2;
        curDay %= 5;

    }
    return 0;
}
```

3.9 寻找一节课排课机会功能的实现

3.9.1 寻找一节课排课机会功能流程图



3.9.2 寻找一节课排课机会核心代码实现

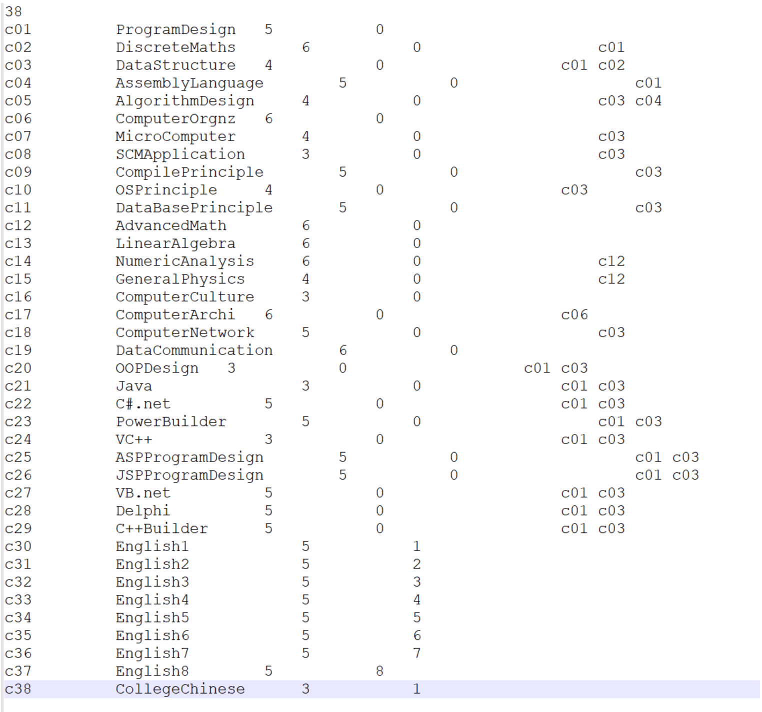
```
int Solution::put1(int curTerm, int& curDay, int i) {
    for (auto t = 0; t < 5; t++) {

        for (int j = 0; j < 10; j++) {
            if (m_courseTable[curTerm][curDay][j] == -1) {
                m_courseTable[curTerm][curDay][j] = i;
                curDay += 2;
                curDay %= 5;
                return 1;
            }
        }
        curDay += 2;
        curDay %= 5;
    }
    return 0;
}
```

4.测试

4.1 常规测试

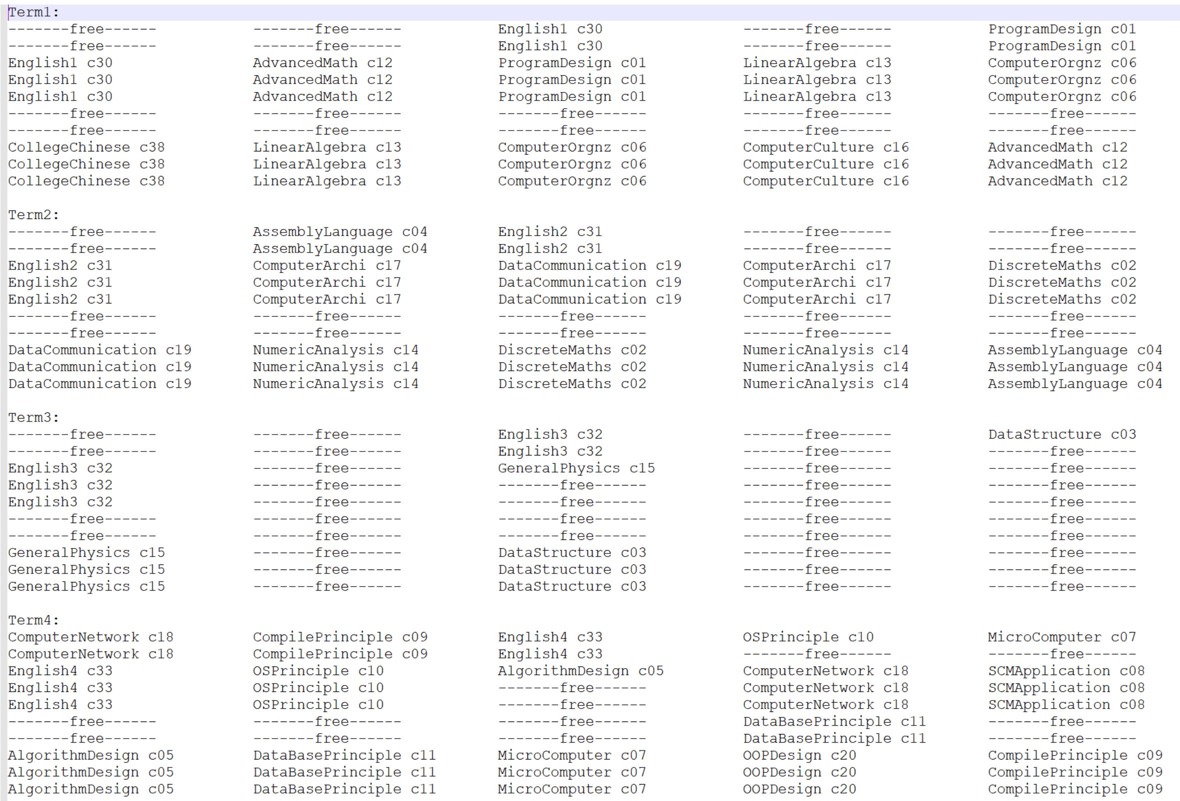
4.1.1 题目要求测试

测试用例：

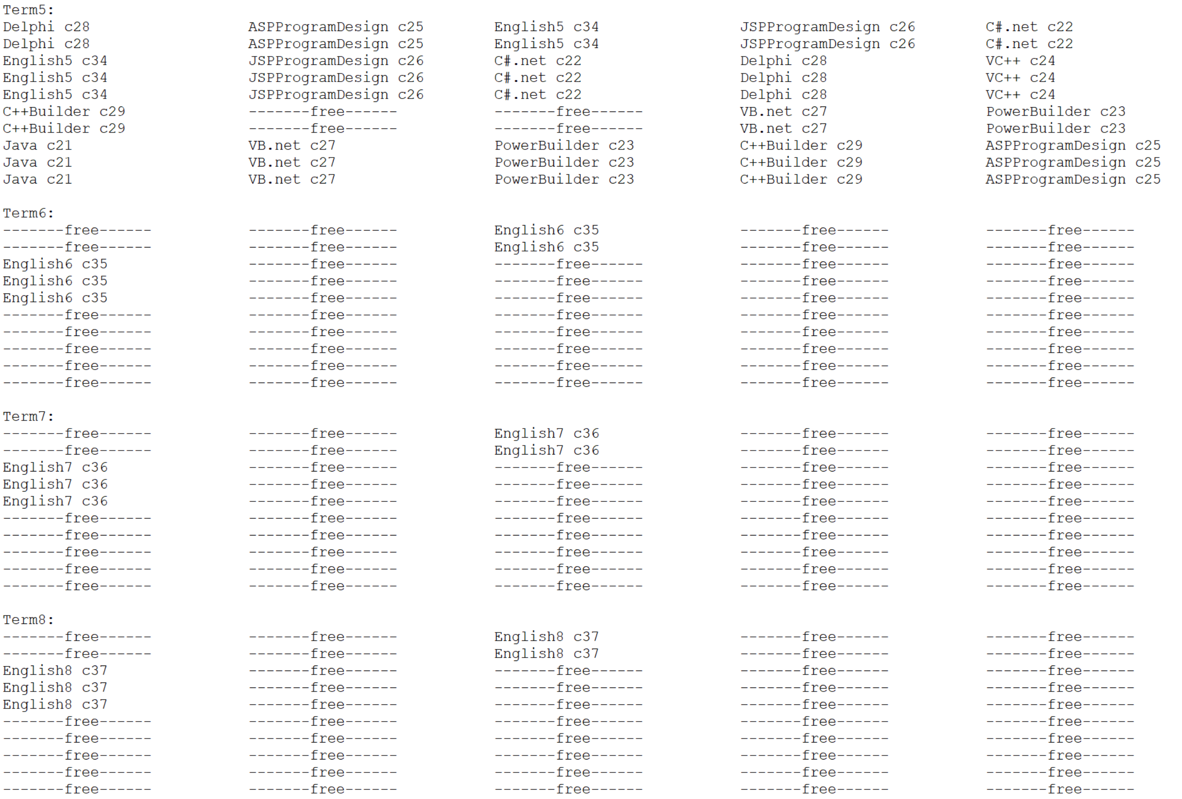


```
38
c01         ProgramDesign   5           0
c02         DiscreteMaths       6           0                   c01
c03         DataStructure   4           0                   c01 c02
c04         AssemblyLanguage        5           0                   c01
c05         AlgorithmDesign     4           0                   c03 c04
c06         ComputerOrgnz   6           0
c07         MicroComputer       4           0                   c03
c08         SCMApplication      3           0                   c03
c09         CompilePrinciple        5           0                   c03
c10         OSPrinciple     4           0                   c03
c11         DataBasePrinciple       5           0                   c03
c12         AdvancedMath        6           0
c13         LinearAlgebra       6           0
c14         NumericAnalysis     6           0                   c12
c15         GeneralPhysics      4           0                   c12
c16         ComputerCulture     3           0
c17         ComputerArchi   6           0                   c06
c18         ComputerNetwork     5           0                   c03
c19         DataCommunication       6           0
c20         OOPDesign   3           0                   c01 c03
c21         Java                3           0               c01 c03
c22         C#.net          5           0                   c01 c03
c23         PowerBuilder        5           0                   c01 c03
c24         VC++            3           0                   c01 c03
c25         ASPProgramDesign        5           0                   c01 c03
c26         JSPProgramDesign        5           0                   c01 c03
c27         VB.net          5           0                   c01 c03
c28         Delphi          5           0                   c01 c03
c29         C++Builder      5           0                   c01 c03
c30         English1            5           1
c31         English2            5           2
c32         English3            5           3
c33         English4            5           4
c34         English5            5           5
c35         English6            5           6
c36         English7            5           7
c37         English8        5           8
c38         CollegeChinese      3           1
```

实验结果：



```
Term1:
-------free------           -------free------           English1 c30                -------free------           ProgramDesign c01
-------free------           -------free------           English1 c30                -------free------           ProgramDesign c01
English1 c30                AdvancedMath c12            ProgramDesign c01           LinearAlgebra c13           ComputerOrgnz c06
English1 c30                AdvancedMath c12            ProgramDesign c01           LinearAlgebra c13           ComputerOrgnz c06
English1 c30                AdvancedMath c12            ProgramDesign c01           LinearAlgebra c13           ComputerOrgnz c06
-------free------           -------free------           -------free------           -------free------           -------free------
-------free------           -------free------           -------free------           -------free------           -------free------
CollegeChinese c38          LinearAlgebra c13           ComputerOrgnz c06           ComputerCulture c16         AdvancedMath c12
CollegeChinese c38          LinearAlgebra c13           ComputerOrgnz c06           ComputerCulture c16         AdvancedMath c12
CollegeChinese c38          LinearAlgebra c13           ComputerOrgnz c06           ComputerCulture c16         AdvancedMath c12

Term2:
-------free------           AssemblyLanguage c04        English2 c31                -------free------           -------free------
-------free------           AssemblyLanguage c04        English2 c31                -------free------           -------free------
English2 c31                ComputerArchi c17           DataCommunication c19       ComputerArchi c17           DiscreteMaths c02
English2 c31                ComputerArchi c17           DataCommunication c19       ComputerArchi c17           DiscreteMaths c02
English2 c31                ComputerArchi c17           DataCommunication c19       ComputerArchi c17           DiscreteMaths c02
-------free------           -------free------           -------free------           -------free------           -------free------
-------free------           -------free------           -------free------           -------free------           -------free------
DataCommunication c19       NumericAnalysis c14         DiscreteMaths c02           NumericAnalysis c14         AssemblyLanguage c04
DataCommunication c19       NumericAnalysis c14         DiscreteMaths c02           NumericAnalysis c14         AssemblyLanguage c04
DataCommunication c19       NumericAnalysis c14         DiscreteMaths c02           NumericAnalysis c14         AssemblyLanguage c04

Term3:
-------free------           -------free------           English3 c32                -------free------           DataStructure c03
-------free------           -------free------           English3 c32                -------free------           -------free------
English3 c32                -------free------           GeneralPhysics c15          -------free------           -------free------
English3 c32                -------free------           -------free------           -------free------           -------free------
English3 c32                -------free------           -------free------           -------free------           -------free------
-------free------           -------free------           -------free------           -------free------           -------free------
-------free------           -------free------           -------free------           -------free------           -------free------
GeneralPhysics c15          -------free------           DataStructure c03           -------free------           -------free------
GeneralPhysics c15          -------free------           DataStructure c03           -------free------           -------free------
GeneralPhysics c15          -------free------           DataStructure c03           -------free------           -------free------

Term4:
ComputerNetwork c18         CompilePrinciple c09        English4 c33                OSPrinciple c10             MicroComputer c07
ComputerNetwork c18         CompilePrinciple c09        English4 c33                -------free------           -------free------
English4 c33                OSPrinciple c10             AlgorithmDesign c05         ComputerNetwork c18         SCMApplication c08
English4 c33                OSPrinciple c10             -------free------           ComputerNetwork c18         SCMApplication c08
English4 c33                OSPrinciple c10             -------free------           ComputerNetwork c18         SCMApplication c08
-------free------           -------free------           -------free------           DataBasePrinciple c11       -------free------
-------free------           -------free------           -------free------           DataBasePrinciple c11       -------free------
AlgorithmDesign c05         DataBasePrinciple c11       MicroComputer c07           OOPDesign c20               CompilePrinciple c09
AlgorithmDesign c05         DataBasePrinciple c11       MicroComputer c07           OOPDesign c20               CompilePrinciple c09
AlgorithmDesign c05         DataBasePrinciple c11       MicroComputer c07           OOPDesign c20               CompilePrinciple c09
```



```
Term5:
Delphi c28                  ASPProgramDesign c25        English5 c34                JSPProgramDesign c26        C#.net c22
Delphi c28                  ASPProgramDesign c25        English5 c34                JSPProgramDesign c26        C#.net c22
English5 c34                JSPProgramDesign c26        C#.net c22                  Delphi c28                  VC++ c24
English5 c34                JSPProgramDesign c26        C#.net c22                  Delphi c28                  VC++ c24
English5 c34                JSPProgramDesign c26        C#.net c22                  Delphi c28                  VC++ c24
C++Builder c29              -------free------           -------free------           VB.net c27                  PowerBuilder c23
C++Builder c29              -------free------           -------free------           VB.net c27                  PowerBuilder c23
Java c21                    VB.net c27                  PowerBuilder c23            C++Builder c29              ASPProgramDesign c25
Java c21                    VB.net c27                  PowerBuilder c23            C++Builder c29              ASPProgramDesign c25
Java c21                    VB.net c27                  PowerBuilder c23            C++Builder c29              ASPProgramDesign c25

Term6:
-------free------           -------free------           English6 c35                -------free------           -------free------
-------free------           -------free------           English6 c35                -------free------           -------free------
English6 c35                -------free------           -------free------           -------free------           -------free------
English6 c35                -------free------           -------free------           -------free------           -------free------
English6 c35                -------free------           -------free------           -------free------           -------free------
-------free------           -------free------           -------free------           -------free------           -------free------
-------free------           -------free------           -------free------           -------free------           -------free------
-------free------           -------free------           -------free------           -------free------           -------free------
-------free------           -------free------           -------free------           -------free------           -------free------
-------free------           -------free------           -------free------           -------free------           -------free------

Term7:
-------free------           -------free------           English7 c36                -------free------           -------free------
-------free------           -------free------           English7 c36                -------free------           -------free------
English7 c36                -------free------           -------free------           -------free------           -------free------
English7 c36                -------free------           -------free------           -------free------           -------free------
English7 c36                -------free------           -------free------           -------free------           -------free------
-------free------           -------free------           -------free------           -------free------           -------free------
-------free------           -------free------           -------free------           -------free------           -------free------
-------free------           -------free------           -------free------           -------free------           -------free------
-------free------           -------free------           -------free------           -------free------           -------free------
-------free------           -------free------           -------free------           -------free------           -------free------

Term8:
-------free------           -------free------           English8 c37                -------free------           -------free------
-------free------           -------free------           English8 c37                -------free------           -------free------
English8 c37                -------free------           -------free------           -------free------           -------free------
English8 c37                -------free------           -------free------           -------free------           -------free------
English8 c37                -------free------           -------free------           -------free------           -------free------
-------free------           -------free------           -------free------           -------free------           -------free------
-------free------           -------free------           -------free------           -------free------           -------free------
-------free------           -------free------           -------free------           -------free------           -------free------
-------free------           -------free------           -------free------           -------free------           -------free------
-------free------           -------free------           -------free------           -------free------           -------free------
```

4.2 鲁棒性测试

4.2.1 较小的课程总数测试

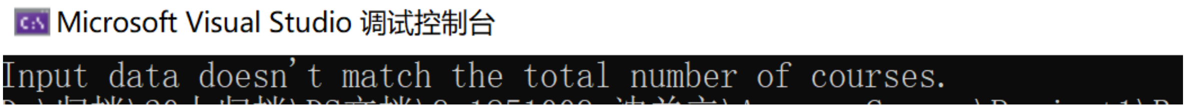
测试用例：

```
同上，但是课程总数为37.
```

预期结果：

```
同上，但是控制台弹出课程数量错误提示。
```

实验结果：



Microsoft Visual Studio 调试控制台

```
Input data doesn't match the total number of courses.
```

4.2.2 较大的课程总数测试

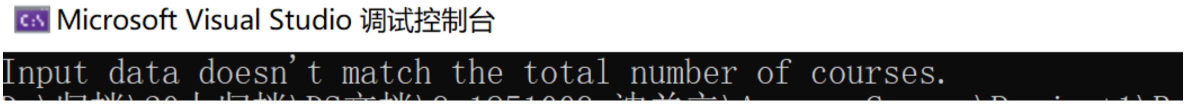
测试用例：

```
同上，但是课程总数为39.
```

预期结果：

```
同上，但是控制台弹出课程数量错误提示。
```

实验结果：



Microsoft Visual Studio 调试控制台

```
Input data doesn't match the total number of courses.
```