

# 数据结构课程设计

## 第 3 次上机实验报告

学 院： 计算机科学与工程学院  
专 业： 计算机科学与技术  
学生姓名： 吴世杰  
班 级： 计算机 191  
学 号： 3190673013

2021 年 5 月 25 日

# 目录

<b>1</b>	<b>题目描述</b>	<b>1</b>
<b>2</b>	<b>主程序描述</b>	<b>2</b>
<b>3</b>	<b>数据结构设计</b>	<b>3</b>
3.1	课程信息节点的设计	3
3.2	课程链表节点的设计	4
3.3	图节点的设计	5
3.4	优先队列的设计	5
3.4.1	优先队列的数据类型	5
3.4.2	优先队列的定义	6
3.4.3	优先队列的初始化及判空操作	6
3.4.4	优先队列的插入操作	6
3.4.5	删除	7
<b>4</b>	<b>算法设计</b>	<b>8</b>
4.1	拓扑排序	8
4.1.1	拓扑排序流程图	8
4.1.2	拓扑排序代码及其注释	10
4.2	排课算法	11
4.2.1	排课算法分析及流程图	11
4.2.2	排课算法代码及注释	13
4.3	对程序时间复杂度的分析	14
<b>5</b>	<b>教学计划编制系统测试</b>	<b>14</b>
5.1	课程信息文件	14
5.2	测试过程	16
5.2.1	测试过程示意图	16
5.2.2	对错误信息的处理	19
<b>6</b>	<b>总结</b>	<b>22</b>
6.1	功能实现效果	22
6.2	存在问题	22
<b>7</b>	<b>附录</b>	<b>23</b>
7.1	附录一	23
7.1.1	八学期制的安排计划	23
7.1.2	十二学期制的安排计划	25
7.2	附录二	28

# 1 题目描述

**题目：**教学计划编制问题（图的应用）

**功能：**

大学的每个专业都要制定教学计划。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值均相等。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，可以有任意多门，也可以没有。每门课恰好占一个学期。试在这样的前提下设计一个教学计划编制程序。

实现提示：

- (1). 输入参数应包括：学期总数，一学期的学分上限
- (2). 每门课：课程号（可以是固定占 3 位的字母数字串）、学分、直接先修课的课程号表，是否基础课，是否专业核心课。至少提供 30 门课程信息，通过文件读入。
- (3). 按下列策略进行课程计划编排：
  - (a). 在各学期中的学习负担尽量均匀，即确定学分最大上限值。
  - (b). 在没有超学分时尽可能优先安排基础课和核心专业课程。
- (4). 若根据所给的所有课程，无法构造拓扑序列则报告适当的信息，建议修改开设的课程；否则给出构造的拓扑序列作为一种教学安排计划，并将该教学计划输出到用户指定的文件中。  
可设学期总数不超过 12，课程总数不超过 60。如果输入的先修课程号不在该专业开设的课程序列中，则作为错误处理。

要求分别输出 8 学期制和 12 学期制的一个计划。输出这两种学制下每一学期的课程选择（这些课程在本学期开课时其先修课程已经修完），且满足学分及基础、核心课优先开设的要求。

## 2 主程序描述

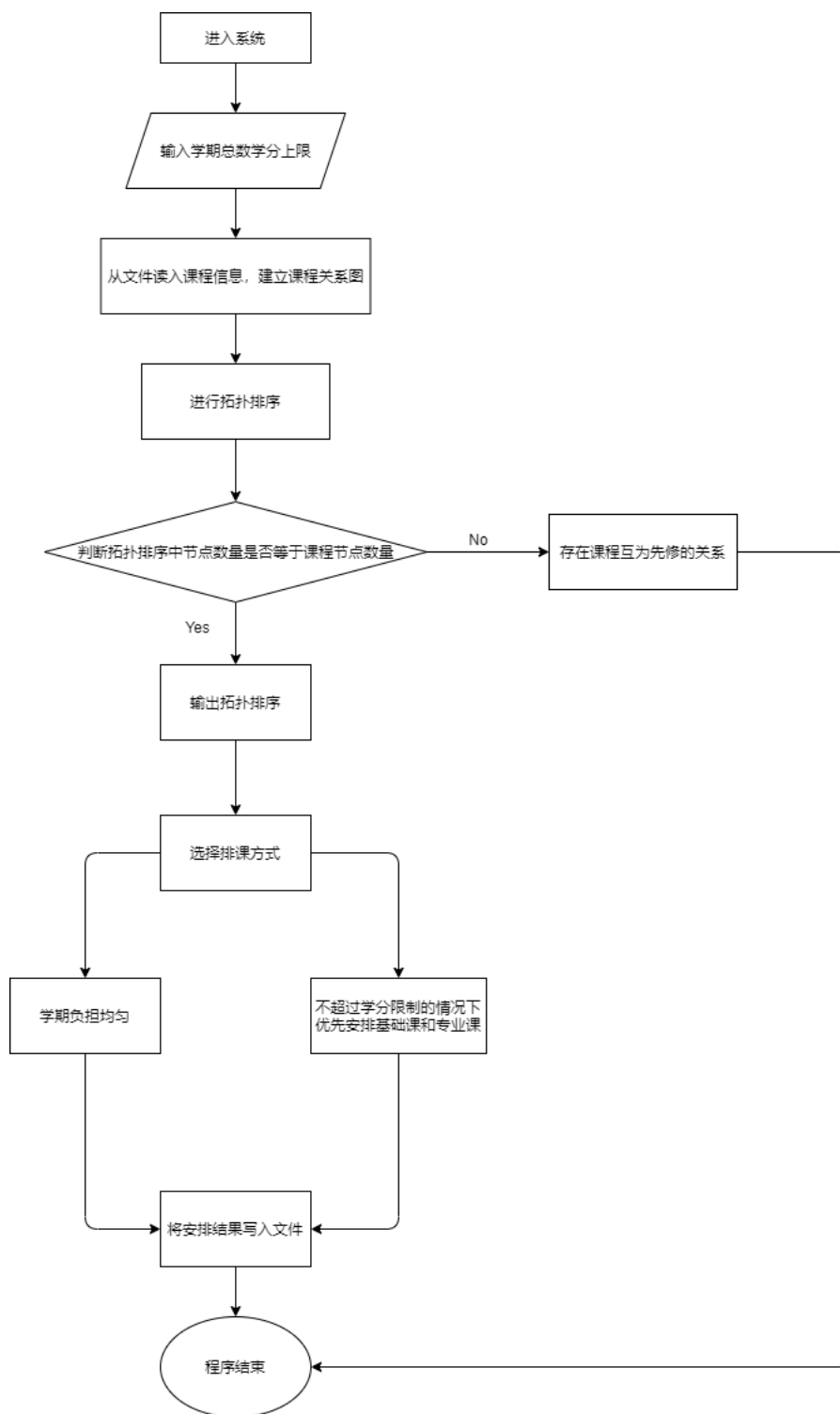


图 1: 主程序流程图示意图

## 3 数据结构设计

### 3.1 课程信息节点的设计

对于教学计划编制系统问题，选取邻接表作为 AOV 网的数据存储结构。

邻接表的表头包含的信息为课程的相关信息，主要为课程号、课程名称、课程学分、直接先修课程数量、直接后续课程数量、直接先修课程链表指针、直接后续课程链表指针、是否基础课、是否核心专业课，数据结构的定义如下：

```
1      typedef struct Node
2      {
3          char Course[10];
4          char CourseName[30];
5          double Score;
6          int isBase;
7          int isCenter;
8          int NumberOfPrimary;
9          int NumberOfFollow;
10         LinkNode *Primary;
11         LinkNode *Follow;
12     }Node;
```

- 课程号:Course
- 课程名称:CourseName
- 课程学分:Score
- 是否基础课:isBase
- 是否专业核心课:isCenter
- 直接先修课程的数量:NumberOfPrimary
- 直接后继课程的数量:NumberOfFollow
- 直接先修课程链表的头指针:Primary
- 直接后继课程链表的头指针:Follow

在从文件读入信息时，添加课程的先修课程时，同时为先修课程添加后继课程，这样在拓扑排序的时候可以方便的找到后继课程，记录后继课程数量作为优先队列比较的关键值

### 3.2 课程链表节点的设计

对于课程链表节点的设计是邻接表的形式,即用于课程先修课程的邻接表,也用于课程后继课程的邻接表,设计如下:

表 1: 课程链表节点的设计

课程号	next 指针
-----	---------

为了更好的理解课程链表节点对课程信息节点的意义,下面用一个样图来展示对于一个具体的图,先修课程链表和后继课程链表的存储结构。

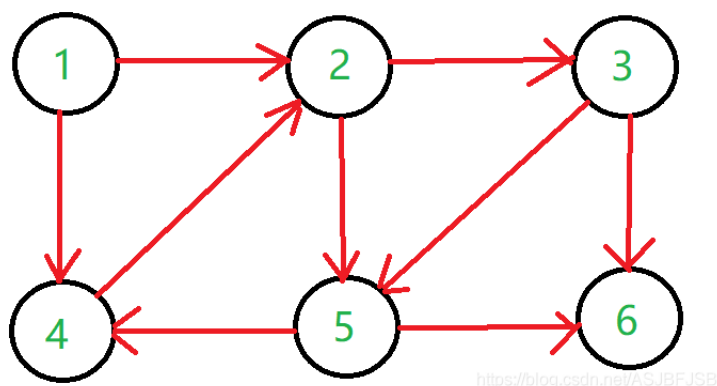


图 2: 示意图

对于上述样图,生成的有向图邻接表的存储结果如图所示,图 3 是每一个节点的直接后继课程链表示意图,图 4 是每一个节点直接先修课程示意图,在添加课程的直接先修的同时添加课程的直接后继。

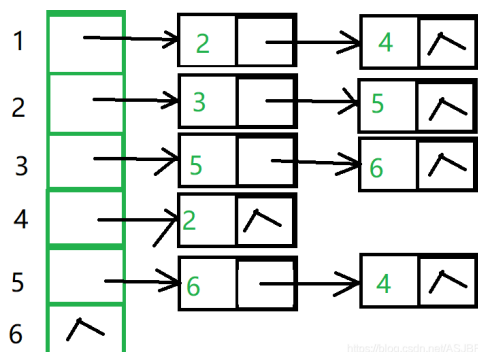


图 3: 后继课程邻接表示意图

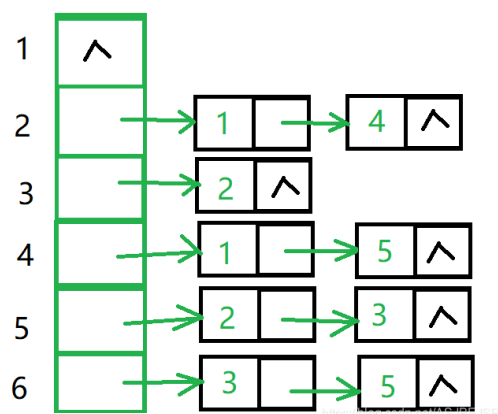


图 4: 先修课程邻接表示意图

### 3.3 图节点的设计

```

1  typedef struct Graph
2  {
3      Node *GraphPointer; //图中的课程数组
4      int NumberOfCourse; //图中课程节点的数量
5      int NumOfSorted; //已进入拓扑序列的课程节点数量
6      int *SortedCourse; //存放拓扑排序的数组
7  }Graph;

```

- 课程数组:GraphPointer
- 课程节点的总数:NumberOfCourse
- 已进入拓扑序列的课程节点数:NumOfSorted
- 拓扑排序数组:SortedCourse

### 3.4 优先队列的设计

**大根堆与小根堆:** 每个结点的值都大于其左孩子和右孩子结点的值, 称之为大根堆; 每个结点的值都小于其左孩子和右孩子结点的值, 称之为小根堆。

对于教学编制问题, 优先队列应该是允许进行下列两种操作的大根堆: Insert (插入) 以及 Delete(弹出当前队列中优先级最高的元素, 并且删除), , 它的工作是找出、返回和删除优先队列中优先级最高的元素。

在实现大根堆的优先队列时候, 要保证需始终保持**堆序**性质

- **堆序:** 即保证父节点大于左右孩子节点对于任何即满足下列公式

$$H \rightarrow node[i] > H \rightarrow node[2 * i] \quad (1)$$

$$H \rightarrow node[i] > H \rightarrow node[2 * i + 1] \quad (2)$$

**优先队列:** 对于教学编制系统实现的优先队列如下图, 可以进行插入和删除操作, 并且优先队列基于大根堆实现, 所以插入和删除的平均时间复杂度为: $O(\log n)$

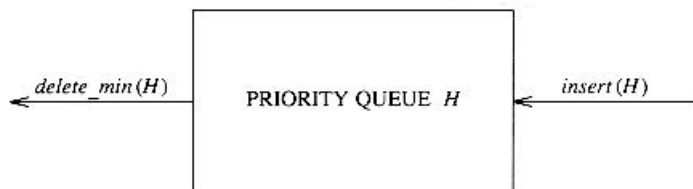


图 5: 优先队列示意图

#### 3.4.1 优先队列的数据类型

对于教学计划编制问题设计的优先队列, 只需要存储优先级来进行排序和存储对应的课程下标即可, 具体设计如下:

```

1  typedef struct Prior_Node
2  {
3      int point;
4      int level;
5  }PNode;

```

- 该节点对应的课程下标:point
- 该节点的优先级:level

### 3.4.2 优先队列的定义

对于优先队列需要队列中节点的信息的数组, 以便于在堆排序中进行调整, 同时需要记录最大容量和当前大小, 来判断队空或者队满的情况。

```

1      typedef struct priorityqueue
2      {
3          PNode *_node;
4          int Capacity;
5          int size;
6      }Priority_Queue;

```

- 优先队列的节点信息数组:node
- 优先队列的最大容量:Capacity
- 优先队列的当前大小:size

### 3.4.3 优先队列的初始化及判空操作

**判空:** 通过 size 判断是否为空, 为空返回 1, 非空返回 0

```

1      int Empty_Prior(Priority_Queue *H)
2      {
3          if(H -> size == 0) return 1;
4          return 0;
5      }

```

**初始化:** 为优先队列分配存储空间, 因此优先队列应该实现大根堆, 所以 0 号节点放置一个优先级无穷大的节点。

```

1      Priority_Queue* Init_Priority(int Size)
2      {
3          Priority_Queue *H = (Priority_Queue*)malloc(sizeof(Priority_Queue));
4          H -> _node = (PNode*)malloc((Size + 1) * sizeof(PNode));
5          if(H -> _node == NULL)
6          {
7              printf("优先队列创建失败!!!\n");
8          }
9          H -> Capacity = Size;
10         H -> size = 0;
11         H -> _node[0].level = 1e9;
12         H -> _node[0].point = -1;
13         return H;
14     }

```

### 3.4.4 优先队列的插入操作

插入操作一般使用的策略叫做上滤: 新元素在堆中上滤直到找出正确的位置 (设堆为 H, 待插入的元素为 x, 首先在 size+1 的位置建立一个空穴, 然后比较 x 和空穴的父亲的大小, 把“小的父亲”换下来, 以此推进, 最后把 x 放到合适的位置)。



```

1 void Insert(Priority_Queue *H, PNode *x)
2 {
3     int i;
4     if(H -> size == H -> Capacity)
5     {
6         printf("堆满!!!\n");
7     }
8
9     for( i = ++ (H -> size) ; H -> _node[i / 2].level < x -> level; i /= 2)
10    {
11        H -> _node[i] = H -> _node[i / 2];
12    }
13    H -> _node[i].level = x -> level;
14    H -> _node[i].point = x -> point;
15 }
16

```

### 3.4.5 删除

与插入“上滤”相对应，采用一种“下滤”的策略，就是逐层推进，把较大的儿子换上来，在具体算法实现上需要注意的是，设堆的最后一个元素是 L，在推进到倒数第二层时，将导致最后一层的某个孩子被换上去而产生一个洞，这时候为了保持堆的结构，必须把最后一个元素运过去补上，此时就存在一个问题，如果 L 比那个孩子小的话就不能保证堆序的性质了，所以在程序中要加一个 if 语句来进行这个边界条件的处理。

```

1 PNode* Delete(Priority_Queue *H)
2 {
3     int i, Child;
4     PNode *Min, *Last;
5     Min = (PNode*)malloc(sizeof(PNode));
6     Last = (PNode*)malloc(sizeof(PNode));
7     if(H -> size == 0)
8     {
9         printf("堆空!!!\n"); return &(H -> _node[0]);
10    }
11    Min -> level = H -> _node[1].level;
12    Min -> point = H -> _node[1].point;
13    Last -> level = H -> _node[H ->size].level;
14    Last -> point = H -> _node[H ->size].point;
15    H -> size --;
16    for(i = 1; i * 2 <= H -> size; i = Child)
17    {
18        Child = i * 2;
19        if(Child != H -> size && H -> _node[Child].level < H -> _node[Child + 1].
20           level) //确定左右孩子中较大的
21        Child ++;
22        if(Last -> level < H -> _node[Child].level)
23        {
24            H -> _node[i].level = H -> _node[Child].level;
25            H -> _node[i].point = H -> _node[Child].point;
26        }
27        else break;
28    }
29    H -> _node[i].level = Last -> level;
30    H -> _node[i].point = Last -> point;
31    return Min;
32 }
33

```

## 4 算法设计

### 4.1 拓扑排序

在教学计划编制系统中，课程拥有确定的先修课程，必须在先修课程完成修读后才能修读本课程。在这样的条件下，需要计算每学期课程的安排方案。

对所有课程构造 AOV 网络，其中课程的先修关系为图的有向边，以邻接表为图的存储结构。完成课程信息的输入并实现课程图的数据结构后，需要生成课程结点的拓扑序列。将课程结点进行拓扑排序得到拓扑序列的算法描述如下：

#### 4.1.1 拓扑排序流程图

- (1). 初始化清空优先队列
- (2). 计算节点的入度
- (3). 将直接先修课程数量为 0 的节点入队
- (4). 判断队列是否为空, 队列为空执行 (11), 否则执行 (5)
- (5). 队首节点出队
- (6). 队首节点进入拓扑序列
- (7). 从图中删除队首节点 (即使队首节点的后继节点入度减一)
- (8). 判断后继节点入度是否为 0, 入度为 0 执行 (9), 否则执行 (10)
- (9). 进入优先队列
- (10). 指针指向下一个后继节点
- (11). 队列为空, 拓扑排序结束
- (12). 判断拓扑排序中节点数量是否等于课程节点数量, 等于执行 (13), 否则执行 (14)
- (13). 输出拓扑排序结果
- (14). 存在课程互为先修关系, 问题无解
- (15). 算法结束

算法流程图如下所示

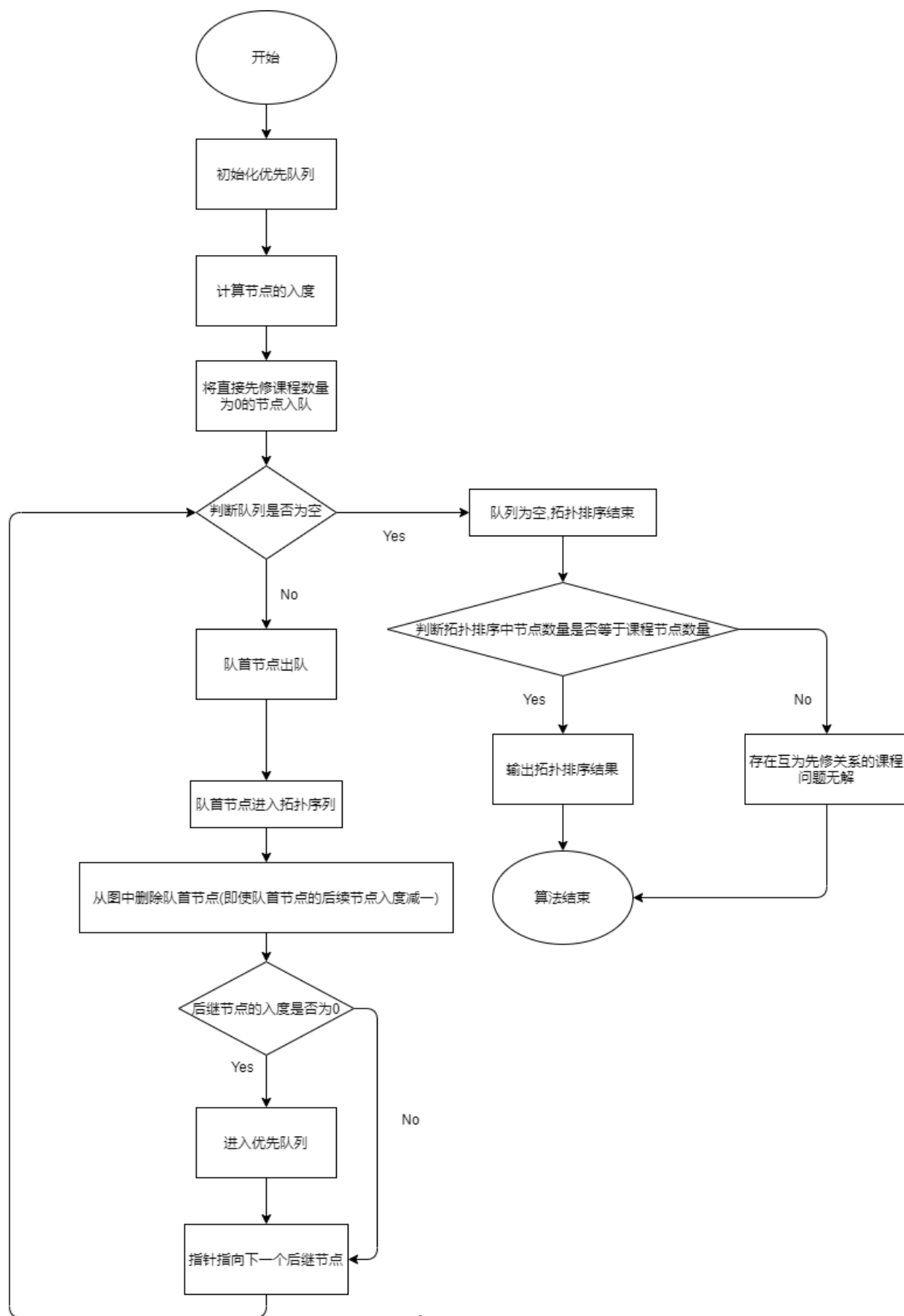


图 6: 拓扑排序流程图

## 4.1.2 拓扑排序代码及其注释

```

1 void TopSort()
2 {
3     Flush();
4     Priority_Queue *q = Init_Priority(2 * Size); //初始化清空优先队列
5     int InDegree[Size]; //节点的入度数组
6     int vis[Size]; //判断节点是否进入拓扑序列
7     memset(InDegree, 0, sizeof(InDegree));
8     memset(vis, 0, sizeof(vis));
9     for(int i = 0; i < Size; i ++ )
10    {
11        if(g.GraphPointer[i].NumberOfPrimary == 0) //将无先修课程的节点入队
12        {
13            PNode *x = (PNode*)malloc(sizeof(PNode));
14            x -> level = g.GraphPointer[i].NumberOfFollow;
15            x -> point = i;
16            if(g.GraphPointer[i].isBase || g.GraphPointer[i].isCenter)
17            {
18                x -> level += 100; //对于基础课和核心课程将优先级加100,保证能将其安排在前
                                   几个学期
19            }
20            Insert(q, x);
21        }
22        InDegree[i] = g.GraphPointer[i].NumberOfPrimary; //记录节点的入度
23    }
24    while(!Empty_Prior(q))
25    {
26        PNode *t;
27        t = Delete(q); //从优先队列中取出后继课程最多的节点
28        int now = t -> point; //指向t存储的课程节点
29        vis[now] = 1; //节点标记为已经进入拓扑序列
30        g.SortedCourse[g.NumOfSorted ++] = now; //节点进入拓扑序列
31        LinkNode *p = g.GraphPointer[now].Follow;
32        while(p != NULL) //遍历后继课程
33        {
34            int pos = Location(p -> Course);
35            if(vis[pos] == 0)
36            {
37                InDegree[pos] --; //将后续节点的入度-1
38                if(InDegree[pos] == 0)
39                {
40                    PNode *x = (PNode*)malloc(sizeof(PNode));
41                    x -> level = g.GraphPointer[pos].NumberOfFollow; //节点的后继课程数
                                   量作为优先队列的判断条件
42                    x -> point = pos;
43                    if(g.GraphPointer[pos].isBase || g.GraphPointer[pos].isCenter)
44                    {
45                        x -> level += 100; //对于基础课和核心课程将优先级加100,保证能将其
                                   安排在前几个学期
46                    }
47                    Insert(q, x);
48                }
49            }
50            p = p -> next;
51        }
52    }
53    if(g.NumOfSorted < Size)
54    {

```

```

55         printf("\033[31m课程存在相互为先修课的关系,请检查!!!\033[0m\n");
56         system("pause");
57         exit(0);
58     }
59     else
60     {
61         printf("拓扑排序结果为:");
62         for(int i = 0; i < g.NumOfSorted; i ++ )
63         {
64             if(i % 2 == 0) printf("\n");
65             int pos = g.SortedCourse[i];
66             printf("%-10s %-30s", g.GraphPointer[pos].Course, g.GraphPointer[pos].
67                 CourseName);
68             if(i != g.NumOfSorted - 1) printf(" -> ");
69         }
70         printf("\n");
71     }
72     system("pause");
73     Flush();
74 }

```

## 4.2 排课算法

### 4.2.1 排课算法分析及流程图

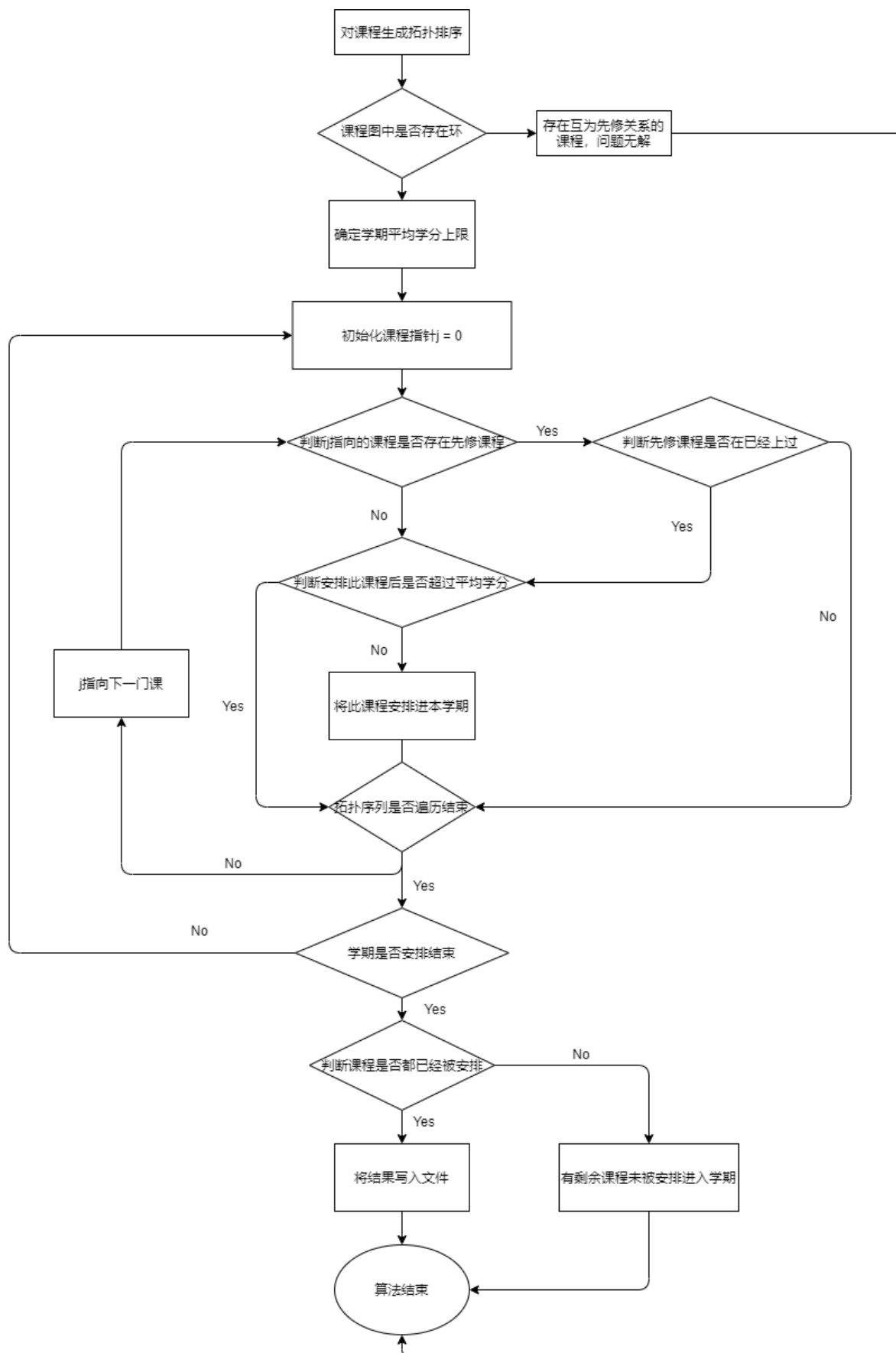
生成课程拓扑序列后, 需要进行排课, 对于以下两种情况

- 使学期负担尽可能均匀
- 在不超过学分限制的条件下尽可能优先安排基础课和核心课

区别只在于在排课时的学分上限不同, 对于第一种情况每学期学分上限应该为  $\frac{\text{所有课程学分和}}{\text{学期数}}$ , 这样可以保证使学期负担尽可能均匀的情况下, 安排完所有课程, 对于第二种情况每学期学分上限应该为最大学分上限.

**算法的描述如下:**

- (1). 生成拓扑排序
- (2). 判断图中是否存在环, 存在转 (15), 不存在转 (3)
- (3). 确定平均学分上限
- (4). 初始化课程指针 j
- (5). 判断 j 指向的课程是否存在先修课程, 不存在转 (7), 存在转 (6)
- (6). 判断先修课程是否在已经上过, 先修课程都已安排转 (7), 否则转 (9)
- (7). 判断安排此课程后是否超过平均学分, 超过转 (9), 未超过转 (8)
- (8). 将此课程安排进本学期
- (9). 判断拓扑序列是否遍历结束, 未结束  $j=j+1$ , 转 (10), 否则转 (10)
- (10). j 指向下一门课, 转 (5)
- (11). 判断学期是否安排结束, 未结束转 (4)
- (12). 判断课程是否都已经被安排, 有剩余课程未安排转 (14)
- (13). 将结果写入文件
- (14). 有剩余课程未被安排进入学期
- (15). 算法结束



12  
图 7: 排课算法流程图

## 4.2.2 排课算法代码及注释

```

1  int Arrange1() //学期负担均匀
2  {
3      int vis[Size]; //记录课程是否安排及安排在第几个学期
4      double lim = ceil(SumScore / CourseNum); //计算最大学分上限
5      memset(vis, 0, sizeof(vis));
6      for(int i = 1; i <= CourseNum; i ++ )
7      {
8          double limscore = 0; //当前学期的学分
9          for(int j = 0; j < g.NumOfSorted; j ++ )
10         {
11             int pos = g.SortedCourse[j];
12             if(g.GraphPointer[pos].NumberOfPrimary)
13             {
14                 if(vis[pos] == 0)
15                 {
16                     int flag = 1;
17                     LinkNode *tmp = g.GraphPointer[pos].Primary; //遍历先修课程
18                     while(tmp != NULL)
19                     {
20                         int idx = Location(tmp -> Course);
21                         if(vis[idx] == 0 || vis[idx] == i) //如果先修课程未安排或者安排在本学期则后继课程不能安排
22                         {
23                             flag = 0;
24                             break;
25                         }
26                         tmp = tmp -> next;
27                     }
28                     if(flag && limscore + g.GraphPointer[pos].Score <= lim) //如果先修课程安排且学分不超过上限完则可以安排进当前学期
29                     {
30
31                         vis[pos] = i;
32                         arr[i].course[arr[i].Count ++] = pos;
33                         limscore += g.GraphPointer[pos].Score;
34                     }
35                 }
36             }
37             else
38             {
39                 if(vis[pos] == 0 && limscore + g.GraphPointer[pos].Score <= lim) //如果没有先修课程且不超过学分上限则可以安排
40                 {
41                     vis[pos] = i;
42                     limscore += g.GraphPointer[pos].Score;
43                     arr[i].course[arr[i].Count ++] = pos;
44                 }
45             }
46         }
47         arr[i].SumScore = limscore;
48     }
49     int cnt = 12;
50     for(int i = 0; i < Size; i ++ )
51     {
52         if(vis[i] == 0)
53         {
54             vis[i] = cnt;

```

```
55         arr[cnt].course[arr[cnt].Count++] = i;
56         arr[cnt].SumScore += g.GraphPointer[i].Score;
57     }
58 }
59
60 for(int i = 0; i < Size; i++) //判读课程是否排完
61 {
62     if(vis[i] == 0)
63     {
64         printf("\033[31m课程未排完,请增加学分上限\033[0m\n");
65         system("pause");
66         exit(0);
67         return 0;
68     }
69 }
70 write(); //将排课结果写入文件
71 return 1;
72 }
```

4.3 对程序时间复杂度的分析

对上述课程图的拓扑排序和排课算法进行算法分析，对于 n 个课程和 e 对课程先修关系的课程 AOV 网而言，求每个课程的直接先修课程数量的时间复杂度为 O(e)；

在课程图的拓扑排序算法中，每个结点入一次队列，出一次队列，在循环中共执行了 n 次，而遍历所有结点的直接先修课程链表的总时间复杂度为 O(e)，所以，拓扑排序的时间复杂度为 O(n+e)，空间复杂度为 O(n+e)。

在排课算法过程中，每个课程结点和课程的直接先修课程链表均遍历了一遍，所以排课算法的总时间复杂度为 O(n+e)，空间复杂度为 O(n)。

5 教学计划编制系统测试

5.1 课程信息文件

课程信息的读入按照以下顺序:

表 2: 课程信息表格

课程号	课程名称	课程学分	课程直接先修课程数	先修课程课程号	是否基础课	是否专业课
A11	高等数学(上)	5.5	0	1	0	
A12	高等数学(下)	5.5	1	A11	1	0
A13	大学计算机基础	2.5	0	1	0	
A14	大学物理(上)	3.5	0	1	0	
A15	大学物理(下)	3.5	2	A12 A14	1	0
B11	概率论与数理统计	3.0	1	A11	1	0
B12	线性代数	2.5	1	A11	1	0
B13	C语言程序设计	4	1	A13	0	1
B14	电路与电子技术	3	1	A15	0	1
C11	数值计算方法	2	2	B11 B12	0	0
C12	离散数学	4	1	B12	0	1
C13	面向对象程序设计	3	1	B13	0	1
C14	算法分析与设计	3	1	B13	0	1
C15	数据结构	4	1	B13	0	1
C16	数字逻辑与数字系统设计	3	1	B14	0	1



16	D11 操作系统 3.5 4 C12 C13 C14 C15 0 1
17	D12 计算机组成原理 4 1 C16 0 1
18	E11 数据库原理 3.5 3 C11 D11 D12 0 1
19	E12 编译原理 3.5 3 C11 D11 D12 0 1
20	E13 微机原理及应用 4 3 C11 D11 D12 0 1
21	E14 计算机网络 3.5 4 C12 C13 C14 C15 0 1
22	E15 软件工程导论A 3 3 C11 D11 D12 0 1
23	F11 数据挖掘 2.5 5 E11 E12 E13 E14 E15 0 1
24	F12 人工智能导论 2 1 B11 0 1
25	F13 计算机图形学 2.5 1 C14 0 1
26	F14 数字图像处理 2 1 F13 0 1
27	F15 信号与系统 2 1 C16 0 1
28	F16 计算机创新能力训练 1 5 E11 E12 E13 E14 E15 0 0
29	F17 软件工具与环境(初级) 3 1 C13 0 1
30	F18 Unix/Linux操作系统分析 2 5 E11 E12 E13 E14 E15 0 1
31	F19 嵌入式系统 2.5 1 C16 0 1
32	F20 人机交互系统 2 5 E11 E12 E13 E14 E15 0 1
33	F21 信息安全技术 2 5 E11 E12 E13 E14 E15 0 1
34	F22 并行计算技术 2.5 5 E11 E12 E13 E14 E15 0 1
35	F23 计算方法 2 1 B12 0 1
36	F24 软件工具与环境(中级) 3 1 F17 0 1
37	F25 软件工具与环境(高级) 3 1 F24 0 1
38	G11 思想道德修养与法律基础 2 0 0 0
39	G12 中国近代史纲要 1 1 G11 0 0
40	G13 军事理论 1 1 G12 0 0
41	G14 毛泽东思想概论 3 1 G13 0 0
42	G15 英语1 4 0 0 0
43	G16 英语2 4 1 G15 0 0
44	G17 英语3 4 1 G16 0 0
45	G18 英语4 4 1 G17 0 0
46	G19 体育1 1 0 0 0
47	G20 体育2 1 1 G19 0 0
48	G21 体育3 1 1 G20 0 0
49	G22 体育4 1 1 G21 0 0
50	G23 工程训练B 2.5 0 0 0

## 5.2 测试过程

### 5.2.1 测试过程示意图

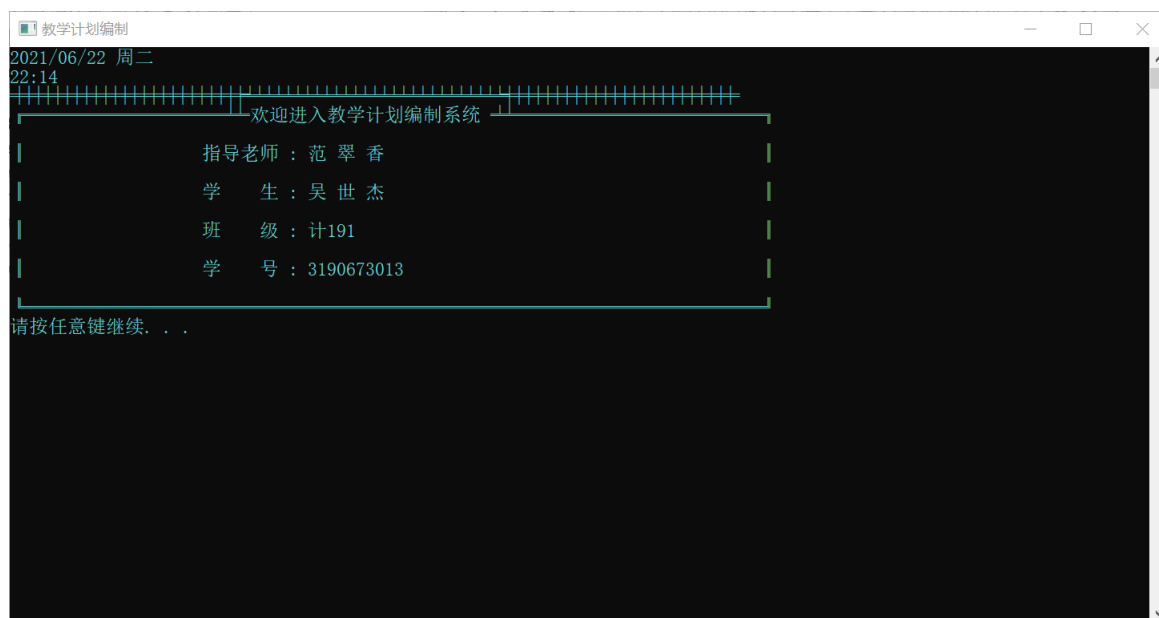


图 8: 系统界面

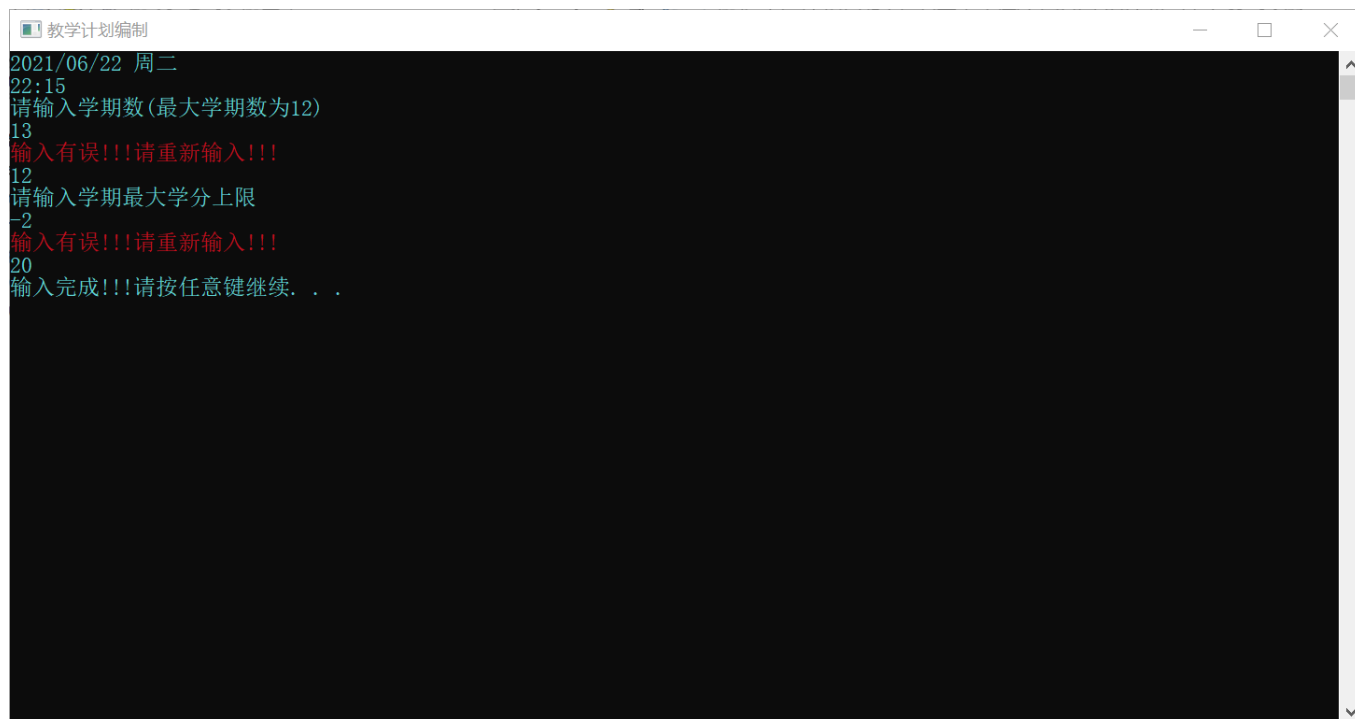


图 9: 输入学期信息和最大学分信息

2021/06/22 周二				
22:15				
课程信息如下				
课程号	课程名称	课程学分	是否基础课	是否专业核心课
A11	高等数学(上)	5.5	是	不是
A12	高等数学(下)	5.5	是	不是
A13	大学计算机基础	2.5	是	不是
A14	大学物理(上)	3.5	是	不是
A15	大学物理(下)	3.5	是	不是
B11	概率论与数理统计	3.0	是	不是
B12	线性代数	2.5	是	不是
B13	C语言程序设计	4.0	不是	是
B14	电路与电子技术	3.0	不是	是
C11	数值计算方法	2.0	不是	不是
C12	离散数学	4.0	不是	不是
C13	面向对象程序设计	3.0	不是	是
C14	算法分析与设计	3.0	不是	是
C15	数据结构	4.0	不是	是
C16	数字逻辑与数字系统设计	3.0	不是	是
D11	操作系统	3.5	不是	是
D12	计算机组成原理	4.0	不是	是
E11	数据库原理	3.5	不是	是
E12	编译原理	3.5	不是	是
E13	微机原理及应用	4.0	不是	是
E14	计算机网络	3.5	不是	是
E15	软件工程导论A	3.0	不是	是
F11	数据挖掘	2.5	不是	是
F12	人工智能导论	2.0	不是	是
F13	计算机图形学	2.5	不是	是
F14	数字图像处理	2.0	不是	是

图 10: 从文件读入的课程信息

2021/06/22 周二				
22:15				
拓扑排序结果为:				
A11	高等数学(上)	-> B12	线性代数	->
B11	概率论与数理统计	-> A12	高等数学(下)	->
C12	离散数学	-> A13	大学计算机基础	->
B13	C语言程序设计	-> C13	面向对象程序设计	->
A14	大学物理(上)	-> C14	算法分析与设计	->
F17	软件工具与环境(初级)	-> A15	大学物理(下)	->
F24	软件工具与环境(中级)	-> B14	电路与电子技术	->
C15	数据结构	-> D11	操作系统	->
C16	数字逻辑与数字系统设计	-> D12	计算机组成原理	->
F25	软件工具与环境(高级)	-> C11	数值计算方法	->
E15	软件工程导论A	-> E12	编译原理	->
E11	数据库原理	-> E13	微机原理及应用	->
E14	计算机网络	-> F23	计算方法	->
F20	人机交互系统	-> F12	人工智能导论	->
F13	计算机图形学	-> F14	数字图像处理	->
F19	嵌入式系统	-> F11	数据挖掘	->
F18	Unix/Linux操作系统分析	-> F22	并行计算技术	->
F15	信号与系统	-> F21	信息安全技术	->
G11	思想道德修养与法律基础	-> G15	英语1	->
G12	中国近代史纲要	-> G16	英语2	->
G13	军事理论	-> G17	英语3	->
G19	体育1	-> G20	体育2	->
G21	体育3	-> G18	英语4	->
G22	体育4	-> F16	计算机创新能力训练	->
G23	工程训练B	-> G14	毛泽东思想概论	->

请按任意键继续...

图 11: 拓扑排序结果

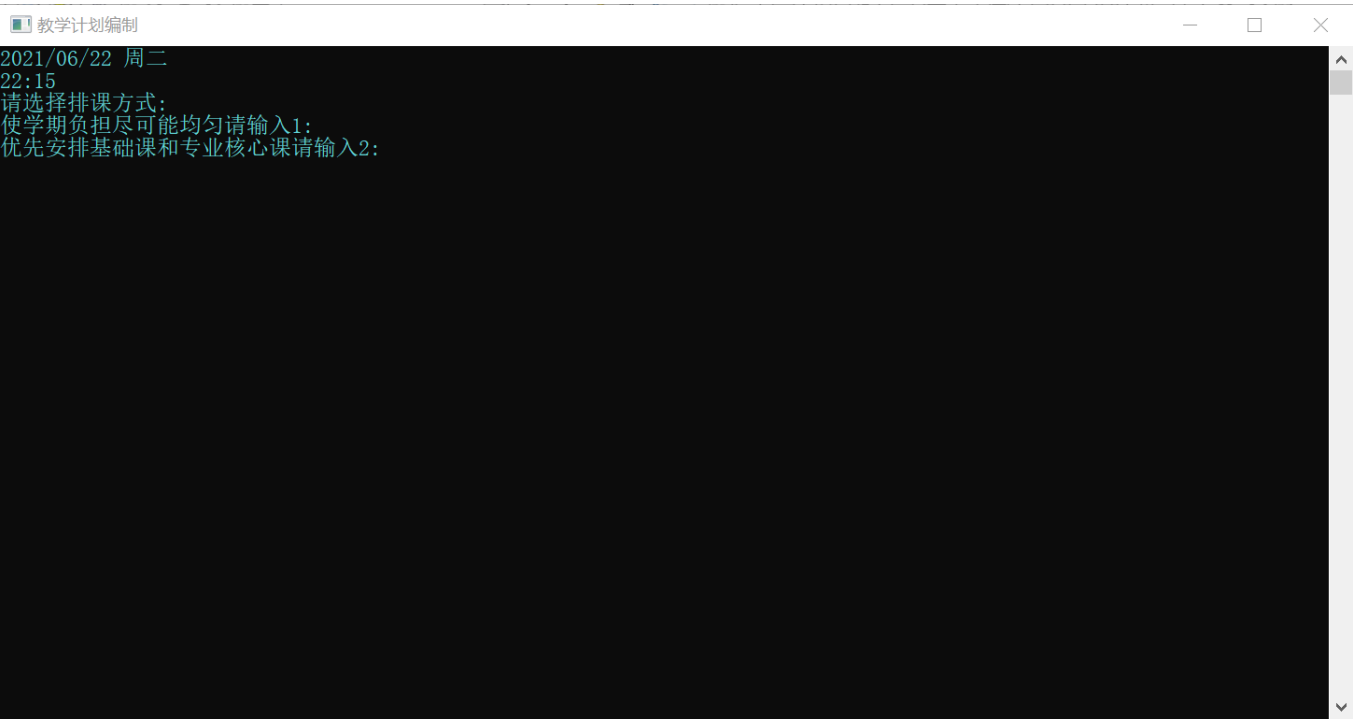


图 12: 选择排课方式

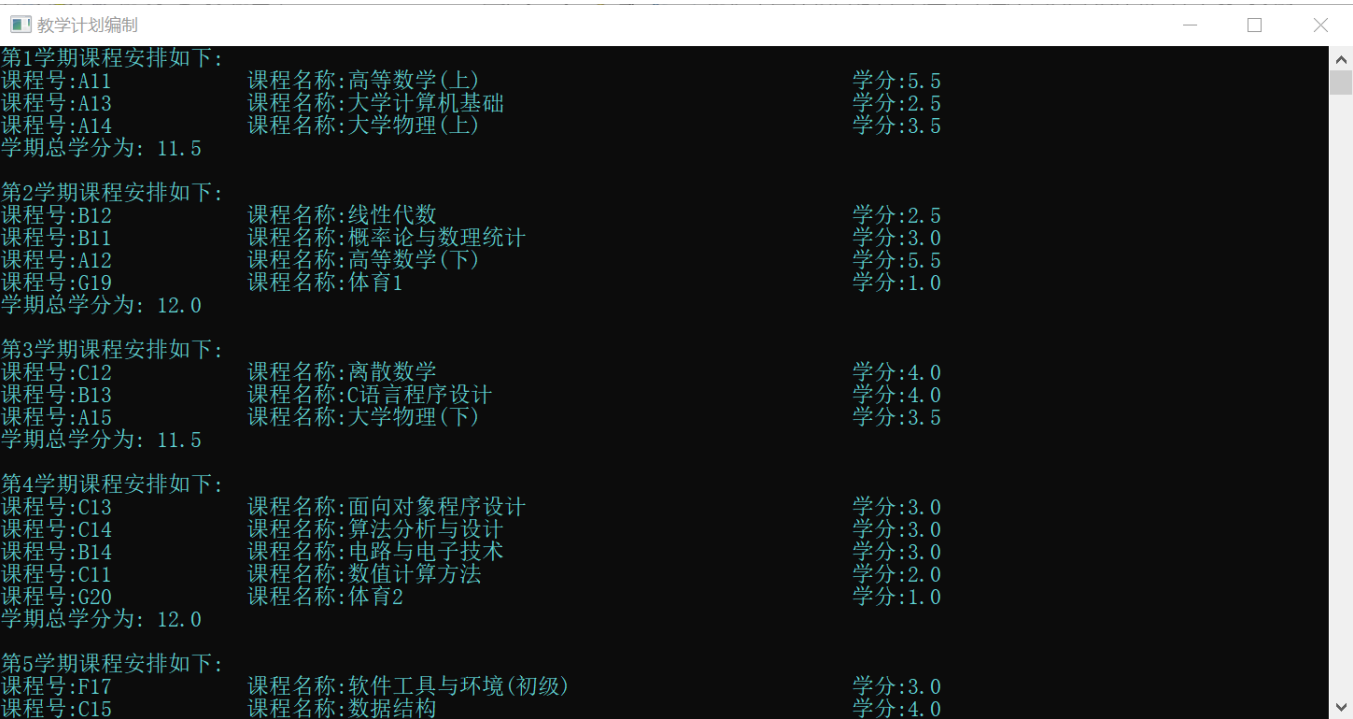


图 13: 使学期负担尽可能均匀的排课方法

教学计划编制		
第1学期课程安排如下:		
课程号:A11	课程名称:高等数学(上)	学分:5.5
课程号:A13	课程名称:大学计算机基础	学分:2.5
课程号:A14	课程名称:大学物理(上)	学分:3.5
课程号:G11	课程名称:思想道德修养与法律基础	学分:2.0
课程号:G15	课程名称:英语1	学分:4.0
课程号:G19	课程名称:体育1	学分:1.0
学期总学分为: 18.5		
第2学期课程安排如下:		
课程号:B12	课程名称:线性代数	学分:2.5
课程号:B11	课程名称:概率论与数理统计	学分:3.0
课程号:A12	课程名称:高等数学(下)	学分:5.5
课程号:B13	课程名称:C语言程序设计	学分:4.0
课程号:G12	课程名称:中国近代史纲要	学分:1.0
课程号:G16	课程名称:英语2	学分:4.0
学期总学分为: 20.0		
第3学期课程安排如下:		
课程号:C12	课程名称:离散数学	学分:4.0
课程号:C13	课程名称:面向对象程序设计	学分:3.0
课程号:C14	课程名称:算法分析与设计	学分:3.0
课程号:A15	课程名称:大学物理(下)	学分:3.5
课程号:C15	课程名称:数据结构	学分:4.0
课程号:C11	课程名称:数值计算方法	学分:2.0
学期总学分为: 19.5		
第4学期课程安排如下:		
课程号:F17	课程名称:软件工具与环境(初级)	学分:3.0
课程号:B14	课程名称:电路与电子技术	学分:3.0

图 14: 尽可能安排基础课和专业课的排课方法

对于具体的课程安排见附录 1

5.2.2 对错误信息的处理

- 输入错误

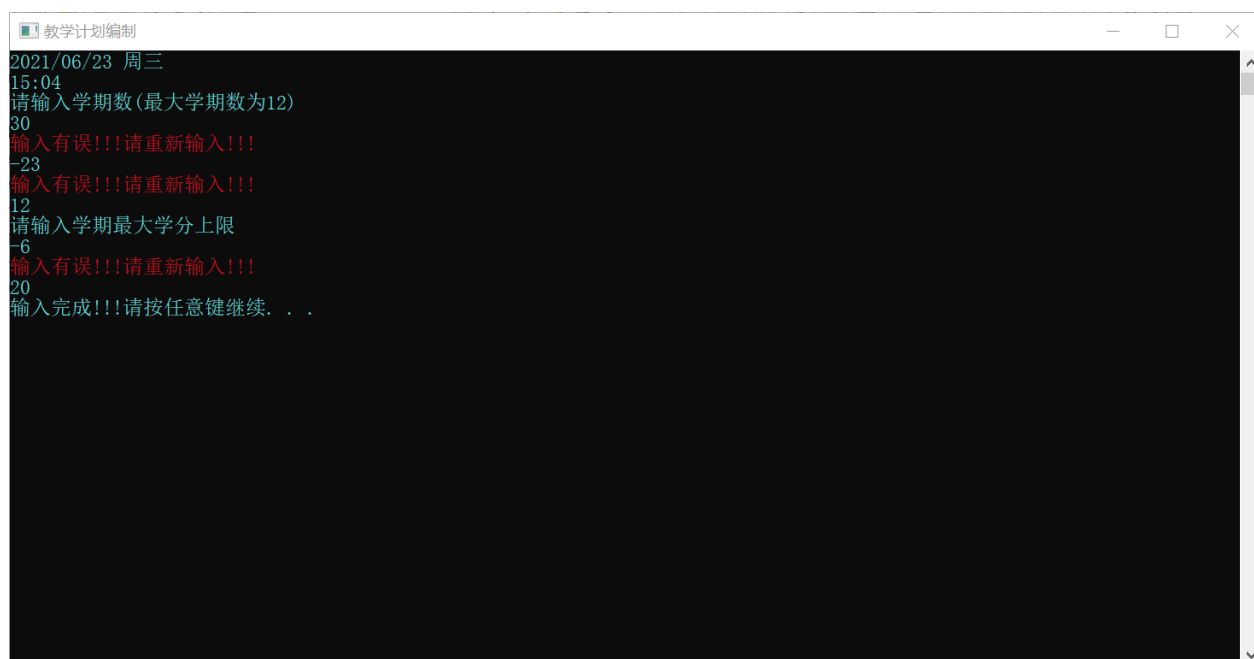


图 15: 输入错误

- 课程不能在给定的学分上限和学期制中排完

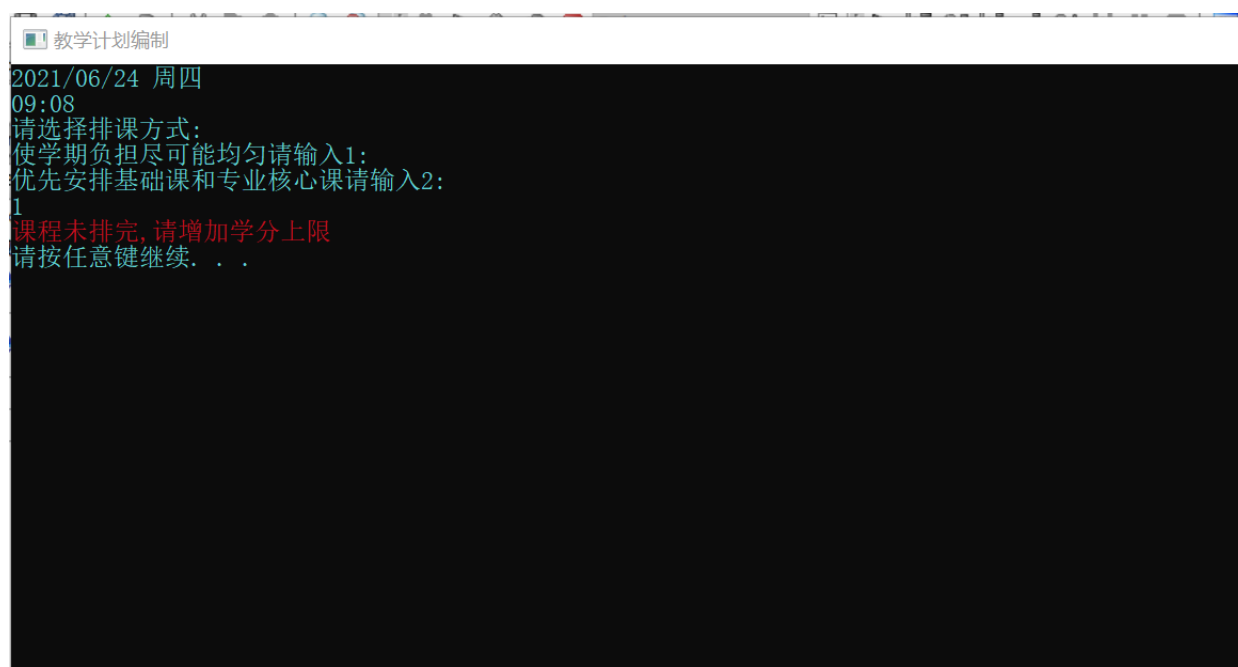


图 16: 在 8 学期制, 学分上限 10 的情况下

- 课程中存在互为先修关系

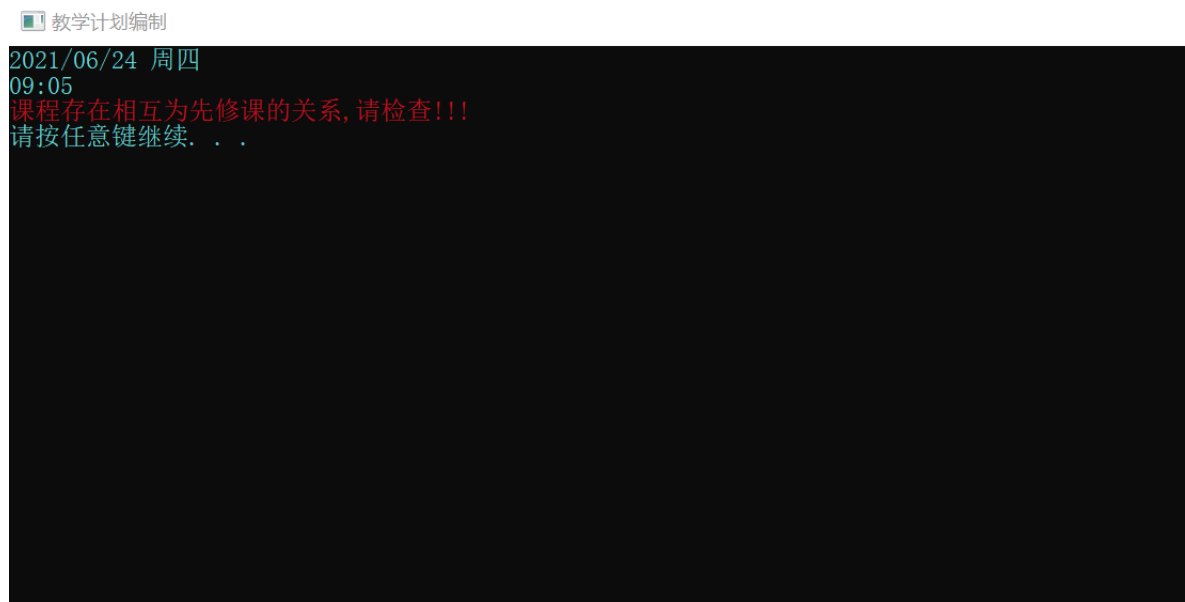


图 17: 互为先修 1

## 6 总结

### 6.1 功能实现效果

本文基于图论,对教学计划编制问题进行了建模,并按课程的先修后续关系采用拓扑排序作为关键算法对课程的 AOV 网络进行求解。在数据结构和算法的设计上,本文采用图的邻接链表实现课程的 AOV 网络,并采用拓扑排序计算课程图的拓扑序列。求出拓扑序列后,在此基础上设计了编制课程图教学计划的排课算法,并使用算法描述和算法流程图详细描述了拓扑排序与排课算法。

在进行拓扑排序时使用了大根堆的优先队列,这样可以满足后继课程多的节点优先安排,同时基础课和专业课的优先级别为无穷,这样也可以满足优先安排基础课和专业课的要求。对于拓扑排序无解(即图种存在环的情况有特判),对于拓扑排序存在的情况,会输出拓扑排序。

对于两种排课方式,8 学期制和 12 学期制的排课情况放在附录一,对于完整的程序(可以通过超链接获取)放在附录二

### 6.2 存在问题

本文仅仅对教学计划编制系统进行了算法研究和实现,基于上述的成果,将来还可以进行后续的研究:

(1) 此程序只能实现无其他复杂关系的课程安排,但大学的课程和教学计划远远比现有模型复杂,还有选修课、实践课等不同特征的课程安排需要进一步研究与考虑。

(2) 由于本人能力有限,所以只能对教学计划编制系统进行了数学建模和算法研究和实现,对于图形化界面和多用户操作等方面,尚需后续进行深入的学习和开发。

(3) 本文的测试数据都较为简单,实际的高校教学计划编制问题的数据量往往较大,需要建立有效的数据库,并实现系统对数据库的访问和存储过程,以及应用程序的数据添加、修改、删除、查询等基本操作。



## 7 附录

### 7.1 附录一

#### 7.1.1 八学期制的安排计划

对于八学期制, 最大学分上限为 20

- 学期负担均匀课程安排

1	第1学期课程安排如下:
2	课程号:A11 课程名称:高等数学(上) 学分:5.5
3	课程号:A13 课程名称:大学计算机基础 学分:2.5
4	课程号:A14 课程名称:大学物理(上) 学分:3.5
5	课程号:G11 课程名称:思想道德修养与法律基础 学分:2.0
6	课程号:G15 课程名称:英语1 学分:4.0
7	学期总学分为: 17.5
8	
9	第2学期课程安排如下:
10	课程号:B12 课程名称:线性代数 学分:2.5
11	课程号:B11 课程名称:概率论与数理统计 学分:3.0
12	课程号:A12 课程名称:高等数学(下) 学分:5.5
13	课程号:B13 课程名称:C语言程序设计 学分:4.0
14	课程号:G12 课程名称:中国近代史纲要 学分:1.0
15	课程号:G19 课程名称:体育1 学分:1.0
16	学期总学分为: 17.0
17	
18	第3学期课程安排如下:
19	课程号:C12 课程名称:离散数学 学分:4.0
20	课程号:C13 课程名称:面向对象程序设计 学分:3.0
21	课程号:C14 课程名称:算法分析与设计 学分:3.0
22	课程号:A15 课程名称:大学物理(下) 学分:3.5
23	课程号:C15 课程名称:数据结构 学分:4.0
24	学期总学分为: 17.5
25	
26	第4学期课程安排如下:
27	课程号:F17 课程名称:软件工具与环境(初级) 学分:3.0
28	课程号:B14 课程名称:电路与电子技术 学分:3.0
29	课程号:D11 课程名称:操作系统 学分:3.5
30	课程号:C11 课程名称:数值计算方法 学分:2.0
31	课程号:G16 课程名称:英语2 学分:4.0
32	课程号:G13 课程名称:军事理论 学分:1.0
33	课程号:G20 课程名称:体育2 学分:1.0
34	学期总学分为: 17.5
35	
36	第5学期课程安排如下:
37	课程号:F24 课程名称:软件工具与环境(中级) 学分:3.0
38	课程号:C16 课程名称:数字逻辑与数字系统设计 学分:3.0
39	课程号:G17 课程名称:英语3 学分:4.0
40	课程号:G21 课程名称:体育3 学分:1.0
41	课程号:G23 课程名称:工程训练B 学分:2.5
42	课程号:G14 课程名称:毛泽东思想概论 学分:3.0
43	学期总学分为: 16.5
44	
45	第6学期课程安排如下:
46	课程号:D12 课程名称:计算机组成原理 学分:4.0
47	课程号:F25 课程名称:软件工具与环境(高级) 学分:3.0
48	课程号:G18 课程名称:英语4 学分:4.0
49	课程号:G22 课程名称:体育4 学分:1.0

50 学期总学分为：12.0  
51  
52 第7学期课程安排如下：  
53 课程号:E15 课程名称: 软件工程导论A 学分:3.0  
54 课程号:E12 课程名称: 编译原理 学分:3.5  
55 课程号:E11 课程名称: 数据库原理 学分:3.5  
56 课程号:E13 课程名称: 微机原理及应用 学分:4.0  
57 课程号:E14 课程名称: 计算机网络 学分:3.5  
58 学期总学分为：17.5  
59  
60 第8学期课程安排如下：  
61 课程号:F23 课程名称: 计算方法 学分:2.0  
62 课程号:F20 课程名称: 人机交互系统 学分:2.0  
63 课程号:F12 课程名称: 人工智能导论 学分:2.0  
64 课程号:F13 课程名称: 计算机图形学 学分:2.5  
65 课程号:F14 课程名称: 数字图像处理 学分:2.0  
66 课程号:F19 课程名称: 嵌入式系统 学分:2.5  
67 课程号:F11 课程名称: 数据挖掘 学分:2.5  
68 课程号:F18 课程名称: Unix/Linux操作系统分析 学分:2.0  
69 学期总学分为：17.5

• 不超过学分上限的情况下优先安排基础课和专业课

1 第1学期课程安排如下：  
2 课程号:A11 课程名称: 高等数学(上) 学分:5.5  
3 课程号:A13 课程名称: 大学计算机基础 学分:2.5  
4 课程号:A14 课程名称: 大学物理(上) 学分:3.5  
5 课程号:G19 课程名称: 体育1 学分:1.0  
6 课程号:G11 课程名称: 思想道德修养与法律基础 学分:2.0  
7 课程号:G15 课程名称: 英语1 学分:4.0  
8 学期总学分为：18.5  
9  
10 第2学期课程安排如下：  
11 课程号:B12 课程名称: 线性代数 学分:2.5  
12 课程号:B11 课程名称: 概率论与数理统计 学分:3.0  
13 课程号:A12 课程名称: 高等数学(下) 学分:5.5  
14 课程号:B13 课程名称: C语言程序设计 学分:4.0  
15 课程号:G20 课程名称: 体育2 学分:1.0  
16 课程号:G12 课程名称: 中国近代史纲要 学分:1.0  
17 课程号:G23 课程名称: 工程训练B 学分:2.5  
18 学期总学分为：19.5  
19  
20 第3学期课程安排如下：  
21 课程号:C12 课程名称: 离散数学 学分:4.0  
22 课程号:C14 课程名称: 算法分析与设计 学分:3.0  
23 课程号:C13 课程名称: 面向对象程序设计 学分:3.0  
24 课程号:C15 课程名称: 数据结构 学分:4.0  
25 课程号:A15 课程名称: 大学物理(下) 学分:3.5  
26 课程号:F23 课程名称: 计算方法 学分:2.0  
27 学期总学分为：19.5  
28  
29 第4学期课程安排如下：  
30 课程号:E14 课程名称: 计算机网络 学分:3.5  
31 课程号:D11 课程名称: 操作系统 学分:3.5  
32 课程号:F17 课程名称: 软件工具与环境(初级) 学分:3.0  
33 课程号:F13 课程名称: 计算机图形学 学分:2.5  
34 课程号:B14 课程名称: 电路与电子技术 学分:3.0  
35 课程号:F12 课程名称: 人工智能导论 学分:2.0

36	课程号:C11 课程名称:数值计算方法 学分:2.0
37	学期总学分为: 19.5
38	
39	第5学期课程安排如下:
40	课程号:F24 课程名称:软件工具与环境(中级) 学分:3.0
41	课程号:C16 课程名称:数字逻辑与数字系统设计 学分:3.0
42	课程号:F14 课程名称:数字图像处理 学分:2.0
43	课程号:G21 课程名称:体育3 学分:1.0
44	课程号:G13 课程名称:军事理论 学分:1.0
45	课程号:G16 课程名称:英语2 学分:4.0
46	学期总学分为: 14.0
47	
48	第6学期课程安排如下:
49	课程号:D12 课程名称:计算机组成原理 学分:4.0
50	课程号:F25 课程名称:软件工具与环境(高级) 学分:3.0
51	课程号:F15 课程名称:信号与系统 学分:2.0
52	课程号:F19 课程名称:嵌入式系统 学分:2.5
53	课程号:G17 课程名称:英语3 学分:4.0
54	课程号:G14 课程名称:毛泽东思想概论 学分:3.0
55	课程号:G22 课程名称:体育4 学分:1.0
56	学期总学分为: 19.5
57	
58	第7学期课程安排如下:
59	课程号:E15 课程名称:软件工程导论A 学分:3.0
60	课程号:E11 课程名称:数据库原理 学分:3.5
61	课程号:E12 课程名称:编译原理 学分:3.5
62	课程号:E13 课程名称:微机原理及应用 学分:4.0
63	课程号:G18 课程名称:英语4 学分:4.0
64	学期总学分为: 18.0
65	
66	第8学期课程安排如下:
67	课程号:F22 课程名称:并行计算技术 学分:2.5
68	课程号:F18 课程名称:Unix/Linux操作系统分析 学分:2.0
69	课程号:F11 课程名称:数据挖掘 学分:2.5
70	课程号:F21 课程名称:信息安全技术 学分:2.0
71	课程号:F20 课程名称:人机交互系统 学分:2.0
72	课程号:F16 课程名称:计算机创新能力训练 学分:1.0
73	学期总学分为: 12.0

7.1.2 十二学期制的安排计划

- 学期负担尽可能均匀

1	第1学期课程安排如下:
2	课程号:A11 课程名称:高等数学(上) 学分:5.5
3	课程号:A13 课程名称:大学计算机基础 学分:2.5
4	课程号:A14 课程名称:大学物理(上) 学分:3.5
5	学期总学分为: 11.5
6	
7	第2学期课程安排如下:
8	课程号:B12 课程名称:线性代数 学分:2.5
9	课程号:B11 课程名称:概率论与数理统计 学分:3.0
10	课程号:A12 课程名称:高等数学(下) 学分:5.5
11	课程号:G19 课程名称:体育1 学分:1.0
12	学期总学分为: 12.0
13	
14	第3学期课程安排如下:
15	课程号:C12 课程名称:离散数学 学分:4.0

16 课程号:B13 课程名称:C语言程序设计 学分:4.0  
 17 课程号:A15 课程名称:大学物理(下) 学分:3.5  
 18 学期总学分为: 11.5  
 19  
 20 第4学期课程安排如下:  
 21 课程号:C13 课程名称:面向对象程序设计 学分:3.0  
 22 课程号:C14 课程名称:算法分析与设计 学分:3.0  
 23 课程号:B14 课程名称:电路与电子技术 学分:3.0  
 24 课程号:C11 课程名称:数值计算方法 学分:2.0  
 25 课程号:G20 课程名称:体育2 学分:1.0  
 26 学期总学分为: 12.0  
 27  
 28 第5学期课程安排如下:  
 29 课程号:F17 课程名称:软件工具与环境(初级) 学分:3.0  
 30 课程号:C15 课程名称:数据结构 学分:4.0  
 31 课程号:C16 课程名称:数字逻辑与数字系统设计 学分:3.0  
 32 课程号:G11 课程名称:思想道德修养与法律基础 学分:2.0  
 33 学期总学分为: 12.0  
 34  
 35 第6学期课程安排如下:  
 36 课程号:F24 课程名称:软件工具与环境(中级) 学分:3.0  
 37 课程号:D11 课程名称:操作系统 学分:3.5  
 38 课程号:D12 课程名称:计算机组成原理 学分:4.0  
 39 课程号:G12 课程名称:中国近代史纲要 学分:1.0  
 40 学期总学分为: 11.5  
 41  
 42 第7学期课程安排如下:  
 43 课程号:F25 课程名称:软件工具与环境(高级) 学分:3.0  
 44 课程号:E15 课程名称:软件工程导论A 学分:3.0  
 45 课程号:E12 课程名称:编译原理 学分:3.5  
 46 课程号:G13 课程名称:军事理论 学分:1.0  
 47 课程号:G21 课程名称:体育3 学分:1.0  
 48 学期总学分为: 11.5  
 49  
 50 第8学期课程安排如下:  
 51 课程号:E11 课程名称:数据库原理 学分:3.5  
 52 课程号:E13 课程名称:微机原理及应用 学分:4.0  
 53 课程号:E14 课程名称:计算机网络 学分:3.5  
 54 课程号:G22 课程名称:体育4 学分:1.0  
 55 学期总学分为: 12.0  
 56  
 57 第9学期课程安排如下:  
 58 课程号:F23 课程名称:计算方法 学分:2.0  
 59 课程号:F20 课程名称:人机交互系统 学分:2.0  
 60 课程号:F12 课程名称:人工智能导论 学分:2.0  
 61 课程号:F13 课程名称:计算机图形学 学分:2.5  
 62 课程号:F14 课程名称:数字图像处理 学分:2.0  
 63 课程号:F16 课程名称:计算机创新能力训练 学分:1.0  
 64 学期总学分为: 11.5  
 65  
 66 第10学期课程安排如下:  
 67 课程号:F19 课程名称:嵌入式系统 学分:2.5  
 68 课程号:F11 课程名称:数据挖掘 学分:2.5  
 69 课程号:F18 课程名称:Unix/Linux操作系统分析 学分:2.0  
 70 课程号:F22 课程名称:并行计算技术 学分:2.5  
 71 课程号:F15 课程名称:信号与系统 学分:2.0  
 72 学期总学分为: 11.5  
 73  
 74 第11学期课程安排如下:  
 75 课程号:F21 课程名称:信息安全技术 学分:2.0

76 课程号:G15 课程名称:英语1 学分:4.0  
 77 课程号:G23 课程名称:工程训练B 学分:2.5  
 78 课程号:G14 课程名称:毛泽东思想概论 学分:3.0  
 79 学期总学分为: 11.5

80  
 81 第12学期课程安排如下:  
 82 课程号:G16 课程名称:英语2 学分:4.0  
 83 课程号:G17 课程名称:英语3 学分:4.0  
 84 课程号:G18 课程名称:英语4 学分:4.0  
 85 学期总学分为: 12.0

- 在不超过学分的情况下, 优先安排基础课和专业课

1 第1学期课程安排如下:  
 2 课程号:A11 课程名称:高等数学(上) 学分:5.5  
 3 课程号:A13 课程名称:大学计算机基础 学分:2.5  
 4 课程号:A14 课程名称:大学物理(上) 学分:3.5  
 5 课程号:G11 课程名称:思想道德修养与法律基础 学分:2.0  
 6 课程号:G15 课程名称:英语1 学分:4.0  
 7 课程号:G19 课程名称:体育1 学分:1.0  
 8 学期总学分为: 18.5

9  
 10 第2学期课程安排如下:  
 11 课程号:B12 课程名称:线性代数 学分:2.5  
 12 课程号:B11 课程名称:概率论与数理统计 学分:3.0  
 13 课程号:A12 课程名称:高等数学(下) 学分:5.5  
 14 课程号:B13 课程名称:C语言程序设计 学分:4.0  
 15 课程号:G12 课程名称:中国近代史纲要 学分:1.0  
 16 课程号:G16 课程名称:英语2 学分:4.0  
 17 学期总学分为: 20.0

18  
 19 第3学期课程安排如下:  
 20 课程号:C12 课程名称:离散数学 学分:4.0  
 21 课程号:C13 课程名称:面向对象程序设计 学分:3.0  
 22 课程号:C14 课程名称:算法分析与设计 学分:3.0  
 23 课程号:A15 课程名称:大学物理(下) 学分:3.5  
 24 课程号:C15 课程名称:数据结构 学分:4.0  
 25 课程号:C11 课程名称:数值计算方法 学分:2.0  
 26 学期总学分为: 19.5

27  
 28 第4学期课程安排如下:  
 29 课程号:F17 课程名称:软件工具与环境(初级) 学分:3.0  
 30 课程号:B14 课程名称:电路与电子技术 学分:3.0  
 31 课程号:D11 课程名称:操作系统 学分:3.5  
 32 课程号:G13 课程名称:军事理论 学分:1.0  
 33 课程号:G17 课程名称:英语3 学分:4.0  
 34 课程号:G20 课程名称:体育2 学分:1.0  
 35 课程号:G23 课程名称:工程训练B 学分:2.5  
 36 学期总学分为: 18.0

37  
 38 第5学期课程安排如下:  
 39 课程号:F24 课程名称:软件工具与环境(中级) 学分:3.0  
 40 课程号:C16 课程名称:数字逻辑与数字系统设计 学分:3.0  
 41 课程号:G21 课程名称:体育3 学分:1.0  
 42 课程号:G18 课程名称:英语4 学分:4.0  
 43 课程号:G14 课程名称:毛泽东思想概论 学分:3.0  
 44 学期总学分为: 14.0  
 45

第6学期课程安排如下:

课程号:D12 课程名称:计算机组成原理 学分:4.0

课程号:F25 课程名称:软件工具与环境(高级) 学分:3.0

课程号:G22 课程名称:体育4 学分:1.0

学期总学分为: 8.0

第7学期课程安排如下:

课程号:E15 课程名称:软件工程导论A 学分:3.0

课程号:E12 课程名称:编译原理 学分:3.5

课程号:E11 课程名称:数据库原理 学分:3.5

课程号:E13 课程名称:微机原理及应用 学分:4.0

课程号:E14 课程名称:计算机网络 学分:3.5

学期总学分为: 17.5

第8学期课程安排如下:

课程号:F23 课程名称:计算方法 学分:2.0

课程号:F20 课程名称:人机交互系统 学分:2.0

课程号:F12 课程名称:人工智能导论 学分:2.0

课程号:F13 课程名称:计算机图形学 学分:2.5

课程号:F14 课程名称:数字图像处理 学分:2.0

课程号:F19 课程名称:嵌入式系统 学分:2.5

课程号:F11 课程名称:数据挖掘 学分:2.5

课程号:F18 课程名称:Unix/Linux操作系统分析 学分:2.0

课程号:F22 课程名称:并行计算技术 学分:2.5

学期总学分为: 20.0

第9学期课程安排如下:

课程号:F15 课程名称:信号与系统 学分:2.0

课程号:F21 课程名称:信息安全技术 学分:2.0

课程号:F16 课程名称:计算机创新能力训练 学分:1.0

学期总学分为: 5.0

第10学期课程安排如下:

学期总学分为: 0.0

第11学期课程安排如下:

学期总学分为: 0.0

第12学期课程安排如下:

学期总学分为: 0.0

## 7.2 附录二

主程序代码链接:<https://paste.ubuntu.com/p/xXPHygcZsG/>

Graph.h 头文件链接:<https://paste.ubuntu.com/p/283VtcTBB6/>

prior.h 头文件链接:<https://paste.ubuntu.com/p/ZM4xQjTkWc/>