# 四川大学计算机学院 (软件学院、智能科学与技术学院) 实验报告

学号: <u>2023141460321</u> 姓名: <u>孙谦昊</u> 专业: <u>计算机科学与技术</u> 班级: <u>行政七班</u> 第 12 周

	7. <u>2023141400321</u> 灶石, <u>小旅关</u> ~亚, <u>万事小时</u>	<u> </u>								
课程 名称	数据结构与算法分析课程设计	实验课时	第 4 次							
实验	排课软件	实验时间	2024年11月19日							
题目		7	, , , ,							
	实验基本要求:									
	1. 根据输入的待排序课程序列文件,输出一种	合法的排课顺	顶序方案。							
实验	2. 输入数据的合理性判断;									
目的	3. 用文本存储输入数据与输出各学期的课表;									
和要	实验扩展:									
求	1. 排课策略:如 3 节连上,一周上多次的优先隔天排课,单双周上课,首次上课不为第 1 周,									
	等等									
	2.可视化课表									
\										
实验	12th Gen Intel(R) Core(TM) i7-1260P 2.10 GHz									
环境	Visual Studio 2022 Debug x64 本地 Windows 调试器	i I								
	本程序综合采用堆栈、队列、广义链表和有向无环图等多种方式实现课程安排和课表									
	可视化输出,具体实现方式如下:									
	进入程序,逐行读取 SCU-Course.cdv 文件中的课程信息至广义表 eachCourseInformati									
	onGeneralList 中进行存储。在 eachCourseInformationGeneralList 中,将课程类型 courseType									
	Id 作为最重要的信息来存储 courseTypeld 下的所有同名但是不同课程 courseId 的课程信息。									
	在此处 eachCourseInformationGeneralList 的作用类似于一个数据库,存储方式如下图所示:									
	上地元 cacheoursemonnationocheratist 用P用天医1 一   数加户,行阳万式知下国州小:									
	CourseTypeId=1 CourseTypeId=2 CourseTypeId=3	• • <u>c</u>	ourseTypeId=n+1							
	CourseTypeId=1 CourseTypeId=2 CourseTypeId=3	C	ourseTypeld=n							
算法	CoureId=1 CoureId=1 CoureId=1		Coureld=1							
描述	CourseTypeld=1 CourseTypeld=2 CourseTypeld=3	2	ourseTypeId=n							
	CoureId=2 CoureId=2 CoureId=2		Coureld=2							
	CourseTypeld=1 CourseTypeld=2 CourseTypeld=3 Coureld=3 Coureld=3	C	ourseTypeld=n Coureld=3							
			Source 0							
	因为某些课程是在某一个学期内必须上的必修课,于是使用了一个 eachTermTheCourse									
	MustStudy[8]的队列来分别存储各个学期内必须要上的必修课,以便在后续排课时直接排入									
	课表。	HJ2  50K)	>1 \C \D \D \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\							
	由于在采用广义表存储所有课程信息时使用的是比较落后原始的定义好课程类型 cour									
	seTypeld 最大种类数量的方式,即哪怕没有这节课	•, •, •, •, •, •, •, •, •, •, •, •, •, •	,,,,							
	SCIYPCIU 软八杆大数里的刀式,咔咖啡仅有处下体	<b>作巴</b> 石的建	1 1 亿,处付限公口用八里							

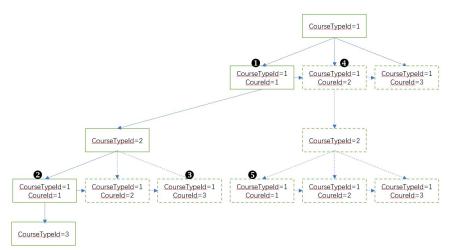
内存空间,增加后续查找某一课程时的时间花费,于是借助 eachCourseInformationGene

ralList 中的 ShortGeneralList();函数来简短广义表,删去删去没有课程内容的课程类型。

随后就是在 while 循环中获取根据所有课程的课程类型 courseTypeId 和课程先修依赖 dependenceTxt 来将课程逐一添加到有向无环图 courseGraph 中,随后马上调用 Topological Sort():函数对课程进行拓补排序。

接下来就是对每个学期的课程进行安排。因为在本程序的实现过程中考虑了课程的上课时间,因此统一 courseTypeld 下的课程 courseId 有很多,这也会导致在排课的时候不同 courseTypeld 之间的课程之间存在时间冲突问题。为了解决这个难题,本程序使用了一个比较类似于树的方式来安排课程。

首先,对于每一个学期的必修课,我们先定义一个二维数组 int courseMap[14][7] = {0}; 其中,0代表当前课时没有课程,反之则为对应课程类型的 courseTypeld。接下来根据对应的学期从存储每个学期必修课课程的队列 eachTermTheCourseMustStudy[8]中获取首元素。对于首元素而言是能够直接填入课表的,于是将 courseMap[14][7]对应位置处标记为首元素的 courseTypeld,并将其作为成功放入课表的课程存储到栈 courseStack 中。接下来从 eachTermTheCourseMustStudy[8]中获取下一个元素。根据第二个元素的 courseTypeld,我们试着把 eachCourseInformationGeneralList 中对应 courseTypeld 的课程子表下的第一节课放入课表中。如果能够放入则直接放入,继续安排下一节必修课。反之我们从子表中读取下一节课。如果成功则放入,继续安排下一节必修课。如果到最后我们都没有成功的把节必修课放入课表,代表在当前排课下是无法满足排课任务的,于是我们需要退回上一节成功放置的必修课,将其修改时间。修改上一节必修课时间时,栈 courseStack 中的作用就体现出来了。因为上一节必修课就在栈 courseStack 的栈顶。将其取出的同时恢复抹除其在课表占的位置,重新将上一节必修课换一个时间安排入课表中。随后试着重新安排下一节课。如此往复,直至能够排好课表。如果排不好课表遍历了所有情况后依旧无法解决课程冲突问题,就证明这课根本没法排,报一个错误让教务处老师解决吧!



这里采用的排课思路很类似于一个树状结构。在程序中并不真实存在这样一棵树(太占空间),而是用栈和链表(每个courseTypeId下属课程可以看为链表)来构建出了这样一棵"虚拟树"。如果一切顺利的话我们希望这棵树能够沿着❶❷直接走下去。但是大概率是不可能的。假如在❷处无法完成排课,我们就需要往❸的方向去尝试。如果到头了仍不行,则需要在重新安排❸后重新开始安排CourseTypeId=2,也就是❺处。

再接下来的程序就是根据学期至少要学的课程数在排完必修课的课表空隙中安排选修课和输出课表了。没啥太大技术难度也没跟算法沾太多边,就不再此叙述了。

源程 序清 单

- 1. ClassInEachTerm.h 输出每学期的课表。
- 2.CourseInformation.h 用于表示课程信息的类

- 3.CourseInformationOverview.h 输出每学期要上的课程的概览
- 4.CourseStack.h 栈,用于以栈的形式存储课程
- 5.GeneralList.h 广义表,用以模拟数据库存储所有课程信息
- 6.Graph.h 有向无环图,用于排序课程
- 7.QueueList.h 队列,用于以队列的形式存储课程
- 8.SCU-Course.csv 两个学期的测试数据集
- 9.实验 4.cpp 主程序,调用各类函数和实现课表排序

# 一、程序可运行功能

程序功能: 1. 根据输入的待排序课程序列文件,输出一种合法的排课顺序方案。

- 2. 输入数据的合理性判断;
- 3. 用文本存储输入数据与输出各学期的课表;
- 4.可视化输出每学期的课表概览和每学期的可视化课表

# 二、程序实际运行结果

第1学期课表总览										
CourseType	Course Name	Teacher Name	Credit	Term	Sessions	Prerequisite				
1	yingyu-1_1	Sun-Qianhao	1	1	1: 1-2					
2	weijifen-1_1	Li-Guo	5	1	3: 1-2	[				
3	C_1	Hu-Dasha	3   3	1	1: 5-9					
4	tiyu-1_1	Shen-Jinxian	1   1	1	5: 8-9					
5	huaxishiyanke_1	Yang-Shiyi	1 1	0	3: 3-4					
6	congtanjidaoguiji_1	Zhao-Hui	1   1	0   0	+   3: 10-11	ļ				
7	jisuanjidaolun_1	Guo-Xinyi	2	0	5: 5-7	 				

运行 结果

第2学期课表总览									
 CourseType	Course Name	Teacher Name	Credit	Term	Sessions	Prerequisite			
8	yingyu-2_1	Sun-Qianhao	1	2	1: 1-2	1			
9	weijifen-2_1	Li-Guo	5	2	3: 1-2	2			
10	Cpp_1	Hu-Dasha	3	2	1: 5-9	3			
11	xinjingji_1	Tang-Xiaoyi	1	0	2: 5-7				
12	shengwuzhi_1	Chen-Sichong	1	0	4: 10-12 				
13	tiangongkaiwu_1	Zhang-Weihua	1	0	6: 1-9				

				课表			
	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
	yingyu-1_1 Sun  -Qianhao		weijifen-1_1 L  i-Guo				
	yingyu-1_1 Sun  -Qianhao		weijifen-1_1 L  i-Guo				
			huaxishiyanke_  1 Yang-Shiyi				
			huaxishiyanke_  1 Yang-Shiyi				
						<del>-</del>	
5	C_1 Hu-Dasha				jisuanjidaolun _1 Guo-Xinyi		
	C_1 Hu-Dasha				jisuanjidaolun _1 Guo-Xinyi		
	C_1 Hu-Dasha				jisuanjidaolun _1 Guo-Xinyi		
	C_1 Hu-Dasha				tiyu-1_1 Shen- Jinxian		
	C_1 Hu-Dasha				tiyu-1_1 Shen-   Jinxian		
10			congtanjidaogu  iji_1 Zhao-Hui				
11			congtanjidaogu  iji_1 Zhao-Hui				
12							

				课表			
	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
1	yingyu-2_1 Sun  -Qianhao		weijifen-2_1 L  i-Guo			tiangongkaiwu_ 1 Zhang-Weihua	
	yingyu-2_1 Sun  -Qianhao		weijifen-2_1 L  i-Guo			tiangongkaiwu_   1 Zhang-Weihua	
						tiangongkaiwu_ 1 Zhang-Weihua	
						tiangongkaiwu_   1 Zhang-Weihua	
	Cpp_1 Hu-Dasha 	xinjingji_1 Ta ng-Xiaoyi				tiangongkaiwu_  1 Zhang-Weihua	
	Cpp_1 Hu-Dasha 	xinjingji_l Ta ng-Xiaoyi				tiangongkaiwu_ 1 Zhang-Weihua	
	Cpp_1 Hu-Dasha	xinjingji_1 Ta ng-Xiaoyi				tiangongkaiwu_  1 Zhang-Weihua	
	Cpp_1 Hu-Dasha					tiangongkaiwu_  1 Zhang-Weihua	
	Cpp_1 Hu-Dasha					tiangongkaiwu_   1 Zhang-Weihua	
10				shengwuzhi_1 C   hen-Sichong			
				shengwuzhi_1 C   hen-Sichong			
				shengwuzhi_1 C   hen-Sichong			

实验 运行 情况 分析 (包 括算 法、 运行 结 果、 运行 环 境 等问 题 的 讨 论 )

#### 一、程序运行情况分析

本程序采用的算法运行出的结果与预期期望相同。正确的实现了课程的安排和课表的可视化输出。

# 二、运行环境讨论

本程序的编写、编译、运行和调试皆是借助 Visual Studio 2022 Debug x64 本地 Windows 调试器。若将本程序使用 DEV C++等软件上编译、运行和调试时可能报错,而这是因为各软件对于支持的 C++版本不同所导致的。例如在 Visual Studio 2022 中可以使用 nullptr,而在部分 DEV C++中仅能使用 NULL。

本程序因为采取的写作思路与事例程序有一定差别,因此代码长度显著长于事例代码。 借助 Visual Studio 2022 Debug x64 本地 Windows 调试器强大的功能能够快速编译运行,而 在部分 DEV C++上则速度较慢。表明程序仍有一定优化空间。

### 三、个人心得

此次程序的写作过程中自己付出了较多的精力,从代码的最底层设计思路和代码的写作基本上是靠自己一个人完成的(中途借鉴了下拓补排序的写法)这一次实验中综合地使用了广义表、栈、队列等形式的数据结构,使我对之前的知识点进行了复习。

在写此次程序过程中最令我自豪的就是使用栈、队列和链表的的形式实现了一棵虚拟树,能够使程序解决课程冲突问题实现正常的排课。当然,这种实现方式也只能安排一个班级的课表,如果是同时安排多个班级的课表话那么程序就费劲儿了。

这一个程序代码行数零零散散又有 900 行左右,在程序写作时就需要开始进行架构。 虽然在开始写作时就已经大致设计好了应该如何去做,不过在实际编写过程中还是出现了 很多 bug,有时为了修复某一个 bug 会编写更多的代码,但是同样地产生了更多的 bug。为 了彻底解决这个问题,不得不在写作过程中进行一些"返工"来解决 bug 优化程序。

总体而言,完成这样一个程序是非常让我自豪的!

在编写程序过程中的相关版本代码发布已经在 Github 中,链接为: https://github.com/StrayerSQH/Learning/tree/main/%E6%95%B0%E6%8D%AE%E7%BB%93%E6%9E%84%E4%B8%8E%E7%AE%97%E6%B3%95%E5%88%86%E6%9E%90%E5%AE%9E%E9%AA%8C4

指导师议

成绩评定:

指导教师签名: