

《程序设计与算法基础 II》课程教学大纲

课程英文名称： Programming and Algorithms II

课程代码： P0900630 学 时 数 ： 48 学 分 数： 3

课程类型：学科基础课

适用学科专业： 软件工程

先修课程：程序设计与算法基础 I

执 笔 者： 吴劲 编写日期： 2023.08 审 核 人： 廖勇

一、课程简介

《程序设计与算法基础II》课程是计算机程序设计的重要理论技术基础，是软件工程专业
的核心课程。掌握数据结构包括数组、队列、堆栈等线性结构，以及树、二叉树、平衡二叉树、
图等非线性结构在程序与算法设计中的应用技巧；理解数据的逻辑结构、存储结构对程序与算
法设计与实现的影响，评估算法的时间复杂度与空间复杂度，具备根据应用数据特点设计恰当
的逻辑结构、存储结构并实现相应算法的能力，具有深度优化程序的能力，形成开发核心基础
软件的潜力。

课程内容不仅是后继《编译原理》、《计算机网络》、《计算机操作系统》等课程的理论
基础，而且应用范围已扩展到图像处理与模式识别，海量数据挖掘，科学数据处理等许多领域。

二、课程目标

课程采用工程教育的基于学习成果教育模式（Outcomes Based Education，OBE）实施教学，
其课程目标、课程模块、培养要求之间的关系见表 1 所示。

表 1 课程目标、课程模块、专业培养要求之间关系

课程目标 (CO)	<p>C01: 理解数据结构、算法和程序设计的关系，掌握时间复杂度和空间复杂度计算方法；具备对复杂软件工程问题的数据结构抽取能力和系统建模能力，具备对模型进行推理和求解的能力；引导学生掌握核心软件技术的信心和决心。</p> <p>C02: 具备分析复杂软件系统的数据结构的能力，从复杂应用问题中抽象出揭示事物内在关系的逻辑数据结构，然后根据逻辑结构设计能够支撑高效处理的存储结构。掌握线性结构、树和图等非线性结构的分析方法，能根据逻辑结构、存储结构选用高效的算法，并编程实现的能力。</p> <p>C03: 分析不同应用数据的特点, 能根据工程需要, 选择并设计合理的逻辑结构、存储结构，在给定的时间复杂度和空间复杂度要求下，选择合适的算法并实现的能力；培养追求精益求精的大国工匠精神。</p> <p>C04: 通过课程的学习，掌握自主学习的方法，提高知识获取和解决问题的能力。</p>
--------------	---

课程模块 (CM)	CM1：程序设计与算法概论 掌握程序设计的数据结构和算法两个核心问题的相关基本概念，理解数据结构、算法和程序设计的关系；掌握时间复杂度和空间复杂度计算方法，理解数据逻辑结构与存储结构的关系，理解同一种逻辑数据结构可以有多种不同的存储结构和编程实现方法。 程序的不同实现方法有不同的效率，理解算法的时间复杂性和空间复杂性的内涵，理解算法时间复杂性和空间复杂性的分析方法。 引导学生掌握核心软件技术的信心和决心。			
	CM2：线性结构 掌握线性表的逻辑结构特点和存储结构，掌握特殊线性表：栈和队列的应用技巧；掌握字符串、数组和广义表等线性表的典型应用案例。在掌握线性表的基本存储结构基础上，根据具体应用问题，拓展和改进现有存储结构，以适应新的应用需求。 锻炼学生专注、谨慎求精、举一反三等优良素养，爱岗敬业，精益求精。培养学生的自主学习能力，提高学生的综合素质。			
	CM3：非线性结构 掌握二叉树、一般的树、图等复杂的非线性数据结构的应用技巧。 掌握树、二叉树、满二叉树和完全二叉树的不同特效，二叉树的遍历方法；掌握二叉树、树及森林的顺序存储及链式存储，以及各种遍历方法，相互间的转换关系；掌握哈夫曼树的构建和哈夫曼编码，线索二叉树的构建。 掌握图的基本操作和存储方法：邻接矩阵、邻接表、十字链表、多重邻接表；掌握图的遍历方法：深度优先遍历和广度优先遍历；掌握图的生成树和最小生成树的概念，最小生成树的两种构建方法：普里姆算法和克鲁斯卡尔算法；掌握有权图最短路径算法；有向无环图的拓扑排序、求关键路径的算法。 通过哈夫曼编码等科研故事启迪学生，勤于思考！勇于探索！坚持不懈！在创造力最强的青年时期也能早出成果，回馈社会！			
	CM4：典型算法 掌握典型的查找和排序算法。 掌握顺序查找、折半查找、索引查找、二叉排序树、平衡二叉树算法的设计与实现方法。掌握哈希函数构造方法及冲突处理策略、哈希表的应用技巧。 掌握内部排序算法，包括直接插入排序、折半插入排序、2 路插入排序、希尔排序、冒泡排序、快速排序、直接选择排序、树形选择排序、堆排序、归并排序；各种排序算法的性能及复杂度分析。 通过数据结构与算法的在线实验，培养学生精益求精的精神。			
	培养目标	毕业要求	指标点	课程目标
PO2	掌握专业知识，能选择恰当的数学模型，能用于描述复杂软件工程，对模型进行推理和求解。	GR1.4	CO1	CM1、CM2、CM3
	能应用数学、自然科学和工程科学的基本原理，识别、判断、表达和分析复杂工程问题的关键环节和参数；	GR2.1	CO2	CM2、CM3、CM4
	能认识到解决复杂工程问题有多种方案	GR2.2	CO3	CM2、CM3、CM4

	可选择			
	掌握自主学习的方法，了解拓展知识和能力的途径；	GR12.2	CO4	CM2、CM3、CM4

本课程目标达成度评价方式见表 2 所示。

表 2 课程目标达成度评价

课程目标	考核方式			考核标准	权重系数	考核模块
	考试	实验	课堂表现			
CO1	√	√	√	百分制	0.3	CM1、CM2、CM3
CO2	√	√	√	百分制	0.2	CM2、CM3、CM4
CO3	√	√	√	百分制	0.3	CM2、CM3、CM4
CO4	√	√	√	百分制	0.2	CM2、CM3、CM4
达成度评价方式	CO1 达成度	(期末考试达成度)*0.5 + (实验达成度)*0.4 + (课堂表现达成度)*0.1				
	CO2 达成度	(期末考试达成度)*0.5 + (实验达成度)*0.4 + (课堂表现达成度)*0.1				
	CO3 达成度	(期末考试达成度)*0.5 + (实验达成度)*0.4 + (课堂表现达成度)*0.1				
	CO4 达成度	(期末考试达成度)*0.5 + (实验达成度)*0.4 + (课堂表现达成度)*0.1				
课程达成度		(CO1 达成度)*0.3 + (CO2 达成度)*0.2 + (CO3 达成度)*0.3 + (CO4 达成度)*0.2				

本课程满足毕业要求指标点的达成度评价方式见表 3 所示。

表 3 毕业要求指标点达成度评价

指标点	*权重系数	考核方式			考核模块
		考试	实验	课堂表现	
GR1.4		√	√	√	CM1、CM2、CM3
GR2.1		√	√	√	CM2、CM3、CM4
GR2.2		√	√	√	CM2、CM3、CM4
GR12.2		√	√	√	CM2、CM3、CM4
达成度评价方式	GR1.4 达成度	CO1 达成度			
	GR2.1 达成度	CO2 达成度			
	GR2.2 达成度	CO3 达成度			
	GR12.2 达成度	CO4 达成度			

*此权重系数指本课程对某项指标点达成度（一个指标点的达成度通常由多门课程支撑）的贡献

度，由毕业要求与高关联课程的支撑关系定义。

三、教学计划

（一）教学内容、要求及教学方法

1 绪论

课程模块：CM1

学时分配：3学时

教学方法：课堂面授 + 学生参与和互动

教学要求：通过本章的学习，理解数据结构和算法的重要性，引起学生的重视。掌握数据，数据结构及算法的基本概念，理解数据结构和算法的关系。掌握最常用的大O时空分析法分析算法复杂度。**引导学生掌握核心软件技术的信心和决心。**

教学内容：本章主要介绍计算机的发展历程，C语言的发展过程，C语言的特点，C语言程序的基本结构，C语言的语法单位，以及C语言程序的开发过程等，为以后各章的学习奠定全面基础。

1.1 数据结构

1.2 算法概述

1.3 算法的评价（算法的时间性能分析、空间性能分析）

重点：掌握数据结构、算法和程序设计的关系，掌握时间复杂度计算方法。

难点：算法的时间和空间复杂度的度量方法及数学内涵。

2 线性表

课程模块：CM2

学时分配：6学时

教学方法：课堂面授 + 学生参与和互动 + icoding在线测评平台作业

教学要求：通过本章学习，使学生更具体的理解抽象数据类型与程序设计的关系，理解线性表的数据元素关系的逻辑设计与物理储存的联系；掌握顺序存储结构和链式存储结构的实现方法及各自的优缺点，掌握逻辑结构上的基本操作及不同物理储存结构的实现（查找、插入和删除），学会分析实际问题的算法大O时空复杂度分析。

锻炼学生专注、谨慎求精、举一反三等优良素质，爱岗敬业，精益求精。培养学生的自主学习能力，提高学生的综合素质。

教学内容：

2.1 线性表的概念及其抽象数据类型定义（略讲+自学）

2.1.1 线性表的逻辑结构

2.1.2 线性表的抽象数据类型定义

2.2 线性表的顺序存储

2.2.1 线性表的顺序存储结构（略讲）

2.2.2 线性表顺序存储结构上的基本运算（细讲）

2.3 线性表的链式存储

2.3.1 单链表（略讲）

2.3.2 单链表上的基本运算（细讲）

2.3.3 循环链表（细讲）

2.3.4 双向链表（细讲）

*2.3.5 静态链表（自学不考）

2.4 线性表应用——一元多项式的表示及相加（细讲）

2.5 顺序表与链表的综合比较（自学）

2.5.1 顺序表和链表的比较

2.5.2 线性表链式存储方式的比较

2.6 总结与提高

2.6.1 主要知识点（自学）

2.6.2 典型题例（部分细讲）

重点：线性表顺序存储、线性表的链式存储结构（链表、单向链表、双向链表、循环链表）的设计与实现，基本操作算法的实现及时空复杂度。

难点：双向链表和循环链表的操作算法的编程实现；在给定的时空复杂度下的算法设计与实现。

3 限定性线性表——栈与队列

课程模块：CM2

学时分配：3 学时

教学方法：课堂面授 + 学生参与和互动 + icoding在线测评平台作业

教学要求：通过本章学习，使学生掌握栈和队列的存储结构（顺序存储、链式存储），基本操作（查找、插入和删除），时空复杂度性能分析；通过自学掌握的栈和队列的基本应用，培养学生根据需求选择合适的数据结构和对应算法并编程实现。

教学内容：

3.1 栈

3.1.1 栈的定义（略讲）

3.1.2 栈的表示和实现（细讲）

3.1.3 栈的应用举例（学生自学并编码实现）

3.1.4 栈与递归的实现（学生自学）

3.2 队列

3.2.1 队列的定义（略讲）

3.2.2 队列的表示和实现（细讲）

3.2.3 队列的应用举例（学生自学并编码实现）

3.3 总结与提高

3.3.1 主要知识点（自学）

3.3.2 典型题例（部分选讲）

重点：栈和队列基本操作算法及实现；顺序队列、循环队列、链式队列。

难点：循环队列的入队出队的条件判断，以及编程实现。

4 串

课程模块：CM2

学时分配：3学时

教学方法：课堂面授 + 学生参与和互动 + icoding在线测评平台作业

教学要求：通过本章学习，使学生理解串的概念；掌握顺序存储结构和链式存储结构的实现方法及各自的优缺点；掌握串的模式匹配方法。

教学内容：

4.1 串的基本概念

4.2 串的存储实现

4.2.1 定长顺序串（Brute-Force算法细讲）

4.2.2 堆串（细讲）

4.2.3 块链串（细讲）

4.3 串的应用举例:简单的行编辑器（学生自学并编码实现）

4.4 总结与提高

4.4.1 主要知识点（自学）

4.4.2 典型题例（自学）

重点：串的顺序存储结构、链式存储结构。

难点：串的模式匹配算法。

5 数组与广义表

课程模块：CM2

学时分配：3学时

教学方法：课堂面授 + 学生参与和互动 + icoding在线测评平台作业

教学要求：通过本章学习，使学生理解数组的概念；掌握数组的顺序存储方式和矩阵的压缩存储的实现方法；掌握对称矩阵、对角矩阵的压缩存储、稀疏矩阵压缩存储以及转置操作算法。

教学内容：

5.1 数组的定义与运算（略讲）

5.2 数组的顺序存储与实现（略讲）

5.3 特殊矩阵的压缩存储（细讲）

5.3.1 规律分布的特殊矩阵（细讲）

5.3.2 稀疏矩阵（细讲）

5.4 广义表（自学）

5.4.1 广义表的概念

5.4.2 广义表的存储结构

5.4.3 广义表的操作实现

5.5 总结与提高（自学）

5.5.1 主要知识点

5.5.2 典型题例

重点：掌握对称矩阵、对角矩阵的压缩存储、稀疏矩阵压缩存储以及转置操作算法。

难点：稀疏矩阵的压缩存储、特殊矩阵压缩后的转置运算。

6 树和二叉树

课程模块：CM3

学时分配：10学时

教学方法：课堂面授 + 学生参与和互动 + icoding在线测评平台作业

教学要求：对二叉树和树的定义、基本术语、基本性质、基本操作，满二叉树和完全二叉树的概念及特征采用略讲；详细讲解二叉树的遍历方法；二叉树、树及森林的顺序存储及链式存储，各种遍历方法，相互间的转换等内容；对哈夫曼树的构建和哈夫曼编码在讲解理论方法的同时，加强在实验课的编程代码实现训练。学生自学线索二叉树，训练学生的自学能力。

通过哈夫曼编码等科研故事启迪学生，勤于思考！勇于探索！坚持不懈！在创造力最强的青年时期也能早出成果，回馈社会！

教学内容：

6.1 树的定义与基本术语

6.2 二叉树

6.2.1 二叉树的定义与基本操作（略讲）

6.2.2 二叉树的性质（细讲）

6.2.3 二叉树的存储结构（细讲）

6.3 二叉树的遍历与线索化

6.3.1 二叉树的遍历（细讲）

6.3.2 遍历算法应用（细讲）

6.3.3 基于栈的递归消除（细讲）

6.3.4 线索二叉树（自学不考）

6.3.5 由遍历序列确定二叉树（细讲）

6.4 树、森林和二叉树的关系

6.4.1 树的存储结构（细讲）

6.4.2 树、森林与二叉树的相互转换（细讲）

6.4.3 树与森林的遍历（细讲）

6.5 哈夫曼树及其应用

6.5.1 哈夫曼树（细讲）

6.5.2 哈夫曼编码（细讲）

*6.6 并查集与等价类划分（自学不考）

6.7 总结与提高

6.7.1 主要知识点（自学）

6.7.2 典型题例（自学）

重点：满二叉树和完全二叉树的基本概念；二叉树、树、森林存储结构；二叉树、树及森林相互间的转换；树的各种遍历方法；哈夫曼树构建及哈夫曼编码。

难点：二叉树的遍历算法和哈夫曼编码算法的实现。

7 图

课程模块：CM3

学时分配：10学时

教学方法：课堂面授 + 学生参与和互动 + icoding在线测评平台作业

教学要求：课堂简要介绍基本概念和术语；详细讲解邻接矩阵、邻接表、十字链表、多重邻接表，图的遍历方法——深度优先和广度优先、有向无环图的拓扑排序等算法思想；详细讲解图的生成树和最小生成树，最小生成树普里姆和克鲁斯卡尔、迪杰斯特拉算法、关键路径算法的数据结构和编程实现；学生自学图的连通性。

教学内容：

7.1 图的定义与基本术语

7.1.1 图的定义（略讲）

7.1.2 基本术语（略讲）

7.2 图的存储结构

7.2.1 邻接矩阵表示法（细讲）

7.2.2 邻接表表示法（细讲）

7.2.3 十字链表（简单讲）

*7.2.4 邻接多重表（自学不考）

7.3 图的遍历

7.3.1 深度优先搜索（细讲）

7.3.2 广度优先搜索（细讲）

7.4 图的应用

7.4.1 图的连通性问题（细讲）

7.4.2 有向无环图的应用（细讲）

7.4.3 最短路径问题（细讲）

7.5 总结与提高

7.5.1 主要知识点（自学）

7.5.2 典型题例（自学）

重点：图的邻接矩阵和邻接表的存储表示，图的深度优先和广度优先遍历，图的最小生成树普里姆和克鲁斯卡尔算法，结点间的最短路径算法，有向无环图的拓扑排序和关键路径的算法。

难点：最小生成树的普里姆和克鲁斯卡尔算法、关键路径的算法、图的拓扑排序算法和最短路径算法的编码实现。

8 查找

课程模块：CM4

学时分配：4 学时

教学方法：课堂面授 + 学生参与和互动 + icoding在线测评平台作业

教学要求：通过本章学习，使学生掌握顺序查找、折半查找、索引查找、二叉排序树、平衡二叉树以及哈希表的查找方法，掌握哈希函数的基本构造方法，解决地址冲突的基本策略（开放定址法、链接地址法）；理解各查找算法的时间复杂度。

教学内容：

8.1 查找的基本概念

8.2 基于线性表的查找法

8.2.1 顺序查找法（自学）

8.2.2 折半查找法（细讲）

8.2.3 分块查找法（略讲）

8.3 基于树的查找法

8.3.1 二叉排序树（细讲）

8.3.2 平衡二叉排序树（细讲）

*8.3.3 B树（自学不考）

8.4 计算式查找法——哈希法

8.4.1 哈希函数的构造方法（细讲）

8.4.2 处理冲突的方法（细讲）

8.4.3 哈希表的查找过程（细讲）

8.4.4 哈希法性能分析（略讲）

8.5 总结与提高

8.5.1 主要知识点（自学）

8.5.2 典型题例（自学）

重点：顺序表的顺序查找、折半查找、二叉排序树、平衡二叉树；哈希表的查找及哈希函数的构造和地址冲突解决办法。

难点：哈希法。

9 内部排序

课程模块：CM4

学时分配：6学时

教学方法：课堂面授 + 学生参与和互动 + icoding在线测评平台作业

教学要求：简要讲解基本概念，详细讲解直接插入排序、折半插入排序、希尔排序、冒泡排序、快速排序、直接选择排序、树形选择排序、堆排序算法、归并排序算法的基本思想，同时，详细讲解这些算法实现的编程技巧。培养追求精益求精的大国工匠精神。

教学内容：

9.1 排序的基本概念（自学）

9.2 插入类排序

9.2.1 直接插入排序（略讲）

9.2.2 折半插入排序（细讲）

9.2.3 希尔排序（细讲）

9.2.4 小结（自学）

9.3 交换类排序

9.3.1 冒泡排序（细讲）

9.3.2 快速排序（细讲）

9.3.3 小结（自学）

9.4 选择类排序

9.4.1 简单选择排序（略讲）

9.4.2 树形选择排序（略讲）

9.4.3 堆排序（细讲）

9.4.4 小结（自学）

9.5 归并排序（细讲）

9.6 分配类排序

9.6.1 多关键字排序（略讲）

9.6.2 链式基数排序（细讲）

9.6.3 基数排序的顺序表实现（自学）

9.7 各种排序方法的综合比较（略讲）

9.8 总结与提高

9.8.1 主要知识点（自学）

9.8.2 典型题例（自学）

重点：插入排序、选择排序、希尔排序、冒泡排序、快速排序、堆排序、归并排序等各种内部排序算法。

难点：堆排序和归并排序。

（二）自学内容和要求

教学内容里面标注为自学的部分，学生可以在教材配套的国家精品在线开放课程上自己学习，自学内容不直接纳入期末最终成绩，学习链接如下：

<https://www.icourse163.org/course/NWU-298002?tid=1472102547>

（三）实践性教学环节和要求

上机作业全部提交到为课程定制的在线测评系统上，强化课后上机实验，强调做中学，上机实验学习链接如下：

<https://icoding.run/login>

通过数据结构与算法的上机实践，培养学生的责任感、注重细节的工匠精神，培养学生精益求精的大国工匠精神。

四、考核方式

icoding自动测评平台实验成绩40% + 期末机考50% + 学生参与和互动（课堂表现）10%

五、建议教材及参考资料

（一）教材：

数据结构：用C语言描述（第3版），耿国华，高等教育出版社，2021年7月。

（二）参考资料

1. C程序设计的抽象思维，闪四清(译)，机械工业出版社，2012年5月。
2. 深入理解计算机系统（原书第3版），龚奕利(译)，机械工业出版社，2017年7月。
3. 数据结构教程(第5版)，李春葆主编，清华大学出版社，2017.07
4. Data Structures and Algorithm Analysis in C (second edition), Mark Allen Weiss, 人民邮电出版社，2005。
5. 数据结构与经典算法, 李千目等，清华大学出版社，2015.07。
6. 数据结构教程(第5版)学习指导，李春葆主编，清华大学出版社，2017.07。
7. 数据结构教程(第5版)上机实验指导，李春葆主编 清华大学出版社，2017.07。
8. 数据结构基础（C语言版）（第2版），朱仲涛译，清华大学出版社 2009。