高等学校计算机专业教材精选·算法与程序设计

数据结构学·练·考

董洁 赵明 主编

清华大学出版社

高等学校计算机专业教材精选·算法与程序设计

数据结构学·练·考

董 洁 赵 明 主编
任 义 韩子扬 编著

清华大学出版社
北京

内 容 简 介

本书以帮助读者掌握数据结构的基本理论和基本方法，加强学生对数据结构算法实际应用技能的训练，提高分析问题和解决问题的能力为目标。全书主要分为4大部分：学习辅导、上机实验、课程设计和模拟试卷。学习辅导部分共有绪论、线性表、栈和队列、串、数组、树与二叉树、图、查找、内部排序9章，每章包括知识结构分析、重难点解析、学习方法指导、典型例题、习题与答案等内容，从知识结构入手，旨在指导读者快速地理解数据结构中的各个知识点，掌握其重点，突破其学习的难点，提供内容丰富、题型多样的例题和习题，有助于读者从中举一反三，深入学习巩固数据结构的相关知识。通过上机实验和课程设计等实践环节加强对数据结构算法实际应用技能的训练，帮助读者领会每种数据结构实现方式的异同，通过感性认识加深对基本概念的理解。模拟试卷，便于学生对自己的学习成果进行综合评估。

本书既可以供高等院校本、专科学生使用，也可以作为计算机专业硕士研究生入学考试的参考书，也可供各类学习数据结构的人员参考使用。

图书在版编目（CIP）数据

数据结构学·练·考/董洁，赵明主编. --北京：清华大学出版社，2013
高等学校计算机专业教材精选·算法与程序设计
ISBN 978-7-302-32054-8

Ⅰ. ①数… Ⅱ. ①董… ②赵… Ⅲ. ①数据结构－高等学校－教学参考资料 Ⅳ. ①TP311.12

中国版本图书馆CIP数据核字(2013)第078935号

责任编辑：袁勤勇　徐跃进
封面设计：傅瑞学
责任校对：时翠兰
责任印制：沈　露

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：三河市李旗庄少明印装厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：15.25　　**字　　数**：367千字
版　　次：2013年6月第1版　　**印　　次**：2013年6月第1次印刷
印　　数：1～3000
定　　价：33.00元

产品编号：050833-01

出版说明

我国高等学校计算机教育近年来迅猛发展，应用所学计算机知识解决实际问题，已经成为当代大学生的必备能力。

时代的进步与社会的发展对高等学校计算机教育的质量提出了更高、更新的要求。现在，很多高等学校都在积极探索符合自身特点的教学模式，涌现出一大批非常优秀的精品课程。

为了适应社会的需求，满足计算机教育的发展需要，清华大学出版社在进行了大量调查研究的基础上，组织编写了《高等学校计算机专业教材精选》。本套教材从全国各高校的优秀计算机教材中精挑细选了一批很有代表性且特色鲜明的计算机精品教材，把作者们对各自所授计算机课程的独特理解和先进经验推荐给全国师生。

本系列教材特点如下。

(1) 编写目的明确。本套教材主要面向广大高校的计算机专业学生，使学生通过本套教材，学习计算机科学与技术方面的基本理论和基本知识，接受应用计算机解决实际问题的基本训练。

(2) 注重编写理念。本套教材作者群为各校相应课程的主讲，有一定经验积累，且编写思路清晰，有独特的教学思路和指导思想，其教学经验具有推广价值。本套教材中不乏各类精品课配套教材，并力图努力把不同学校的教学特点反映到每本教材中。

(3) 理论知识与实践相结合。本套教材贯彻从实践中来到实践中去的原则，书中的许多必须掌握的理论都将结合实例来讲，同时注重培养学生分析、解决问题的能力，满足社会用人要求。

(4) 易教易用，合理适当。本套教材编写时注意结合教学实际的课时数，把握教材的篇幅。同时，对一些知识点按教育部教学指导委员会的最新精神进行合理取舍与难易控制。

(5) 注重教材的立体化配套。大多数教材都将配套教师用课件、习题及其解答，学生上机实验指导、教学网站等辅助教学资源，方便教学。

随着本套教材陆续出版，相信能够得到广大读者的认可和支持，为我国计算机教材建设及计算机教学水平的提高，为计算机教育事业的发展做出应有的贡献。

清华大学出版社

前　言

“数据结构”是计算机科学与技术专业的专业基础课，也是十分重要的核心课程，同时是后续多门课程的基础，例如，操作系统中设备管理用到的线性链表，最短作业优先服务用到的队列；编译原理中表达式求解用到的栈，符号管理技术用到的哈希查找；数据库存储用到的 B^+ 树等。所有的计算机系统软件和应用软件都要用到各种类型的数据结构。

随着计算机应用范围的不断扩大，学习和使用计算机的人群已经不限于计算机专业，非计算机信息类或理工类专业也把数据结构作为一门必修的基础课，以便有效地使用计算机、充分发挥计算机的性能。因此，学好“数据结构”，对于学习计算机及其相关专业的学生，具有十分重要意义。

数据结构主要分析研究的是计算机处理的数据对象的特性以及数据元素之间的关系，以便为应用涉及的数据选择适当的逻辑结构、存储结构和相应的算法，并初步掌握算法的时间分析和空间分析的技术，但课程内容涉及数据的组织、存储和运算的一般方法，其原理和算法比较抽象，难度较大，常常觉得难以理解，对部分习题的解答感到无从下手，造成学习上的困难。为此，作者将多年的教学经验做了系统的总结，从初学者的学习心理出发，对课程内容做层次化处理，从低到高，逐步提升，做到高度概括、简单易懂，以便读者能够更好地学习和掌握课程内容，为专业的学习打下良好的基础。

本书分为学习辅导、上机实验、课程设计和模拟试卷 4 个部分。第一部分学习辅导共有绪论、线性表、栈和队列、串、数组、树与二叉树、图、查找、内部排序 9 章，每章包括知识结构分析、重难点解析、学习方法指导、典型例题、习题与答案等内容。知识结构分析主要是对各章知识结构和相互关系进行归纳总结；重难点解析主要是对各章重点和难点进行讲解，力求言简意赅；学习方法指导是对本章涉及的知识点进行多种形式的启发、引导式的要点讲解；典型例题是作者根据各章内容精心编选的，既包括根据多年教学经验自编的例题，也包括部分高校考研试题、高级程序员考试试题等，并给出了详尽的分析与解题过程；习题与习题答案，是通过做题进行强化训练，以巩固所学知识。第二部分为上机实验，给出了包括线性表、栈和队列、字符串和数组、树状结构、图状结构、查找、排序等实验，每个实验都给出了能够运行的程序，并在此基础上提出实验题目，是对每章涉及的知识点进行进一步的扩展和对学生实践能力的培养。第三部分为课程设计，包括航空客运订票系统、迷宫问题、社区医院选址问题、基于赫夫曼树的编码/译码系统等 8 个综合性实验，以加强学生对数据结构算法实际应用技能的训练，提高分析问题和解决问题的能力。第四部分的模拟试卷是为了满足教学和各类学生学习与考前复习的需要，给出了两套模拟试卷与答案，便于读者学习后进行自我测试，检查效果。

本书第一部分的第 1、2、6、7 章由董洁编写，第 3、4、5、8、9 章由赵明编写，第二部分由任义编写，第三、四部分由韩子扬编写，孙雪阳和王守金完成了部分资料的收集，由董洁对全文进行通审和定稿。

由于作者水平有限，书中难免有错误和疏漏之处，恳请广大读者指正。

编　者

2013 年 1 月

目　录

第一部分　学习辅导

第 1 章　绪论 …… 1

1.1　学习指导 …… 1

1.1.1　知识结构分析 …… 1

1.1.2　重难点解析 …… 1

1.1.3　学习方法指导 …… 4

1.2　典型例题 …… 7

1.3　习题 …… 8

1.4　习题参考答案 …… 11

第 2 章　线性表 …… 14

2.1　学习指导 …… 14

2.1.1　知识结构分析 …… 14

2.1.2　重难点解析 …… 14

2.1.3　学习方法指导 …… 18

2.2　典型例题 …… 21

2.3　习题 …… 24

2.4　习题参考答案 …… 28

第 3 章　栈和队列 …… 34

3.1　学习指导 …… 34

3.1.1　知识结构分析 …… 34

3.1.2　重难点解析 …… 34

3.1.3　学习方法指导 …… 37

3.2　典型例题 …… 38

3.3　习题 …… 41

3.4　习题参考答案 …… 47

第 4 章　串 …… 52

4.1　学习指导 …… 52

4.1.1　知识结构分析 …… 52

4.1.2　重难点解析 …… 52

4.1.3　学习方法指导 …… 54

4.2 典型例题 …… 56
4.3 习题 …… 57
4.4 习题参考答案 …… 58

第 5 章 数组和广义表 …… 60
5.1 学习指导 …… 60
5.1.1 知识结构分析 …… 60
5.1.2 重难点解析 …… 60
5.1.3 学习方法指导 …… 65
5.2 典型例题 …… 66
5.3 习题 …… 67
5.4 习题参考答案 …… 68

第 6 章 树和二叉树 …… 71
6.1 学习指导 …… 71
6.1.1 知识结构分析 …… 71
6.1.2 重难点解析 …… 72
6.1.3 学习方法指导 …… 77
6.2 典型例题 …… 81
6.3 习题 …… 85
6.4 习题参考答案 …… 90

第 7 章 图 …… 97
7.1 学习指导 …… 97
7.1.1 知识结构分析 …… 97
7.1.2 重难点解析 …… 97
7.1.3 学习方法指导 …… 99
7.2 典型例题 …… 111
7.3 习题 …… 115
7.4 习题参考答案 …… 121

第 8 章 查找 …… 127
8.1 学习指导 …… 127
8.1.1 知识结构分析 …… 127
8.1.2 重难点解析 …… 127
8.1.3 学习方法指导 …… 133
8.2 典型例题 …… 134
8.3 习题 …… 138
8.4 习题参考答案 …… 142

第 9 章　内部排序 …… 146
9.1　学习指导 …… 146
9.1.1　知识结构分析 …… 146
9.1.2　重难点解析 …… 146
9.1.3　学习方法指导 …… 150
9.2　典型例题 …… 152
9.3　习题 …… 154
9.4　习题参考答案 …… 158

第二部分　上机实验

第 10 章　实验一　顺序表 …… 165
10.1　实验目的 …… 165
10.2　实验内容 …… 165
10.3　实验要求 …… 167
10.4　实验拓展 …… 167

第 11 章　实验二　线性链表 …… 169
11.1　实验目的 …… 169
11.2　实验内容 …… 169
11.3　实验要求 …… 172
11.4　实验拓展 …… 172

第 12 章　实验三　栈与队列 …… 173
12.1　实验目的 …… 173
12.2　实验内容 …… 173
12.2.1　栈的实验部分 …… 173
12.2.2　队列的实验部分 …… 175
12.3　实验要求 …… 176
12.4　实验拓展 …… 176

第 13 章　实验四　串与数组 …… 178
13.1　实验目的 …… 178
13.2　实验内容 …… 178
13.2.1　串的实验部分 …… 178
13.2.2　数组的实验部分 …… 180
13.3　实验要求 …… 181
13.4　实验拓展 …… 181

第 14 章　实验五　树和二叉树 …… 183
14.1　实验目的 …… 183
14.2　实验内容 …… 183
14.3　实验要求 …… 185
14.4　实验拓展 …… 185

第 15 章　实验六　图 …… 186
15.1　实验目的 …… 186
15.2　实验内容 …… 186
15.3　实验要求 …… 189
15.4　实验拓展 …… 189

第 16 章　实验七　查找 …… 191
16.1　实验目的 …… 191
16.2　实验内容 …… 191
16.3　实验要求 …… 193
16.4　实验拓展 …… 193

第 17 章　实验八　内部排序 …… 194
17.1　实验目的 …… 194
17.2　实验内容 …… 194
17.3　实验要求 …… 195
17.4　实验拓展 …… 196

第三部分　课 程 设 计

第 18 章　项目一　迷宫问题 …… 199
18.1　设计内容 …… 199
18.2　实现提示 …… 200
18.3　选作内容 …… 202

第 19 章　项目二　社区医院选址问题 …… 203
19.1　设计内容 …… 203
19.2　实现提示 …… 203
19.3　选作内容 …… 204

第 20 章　项目三　航空客运订票系统 …… 205
20.1　设计内容 …… 205
20.2　实现提示 …… 206
20.3　选作内容 …… 206

第 21 章　项目四　赫夫曼编码译码系统 …… 207
21.1　设计内容 …… 207
21.2　实现提示 …… 208

第 22 章　项目五　电话号码查询系统 …… 209
22.1　设计内容 …… 209
22.2　实现提示 …… 210
22.3　选作内容 …… 210

第 23 章　项目六　汽车牌照信息快速查找 …… 211
23.1　设计内容 …… 211
23.2　实现提示 …… 212
23.3　选作内容 …… 212

第 24 章　项目七　回文的判定 …… 213
24.1　设计内容 …… 213
24.2　实现提示 …… 213
24.3　选作内容 …… 214

第 25 章　项目八　八皇后问题 …… 215
25.1　设计内容 …… 215
25.2　实现提示 …… 215
25.3　选作内容 …… 216

第四部分　模 拟 试 卷

第 26 章　数据结构自测题(一) …… 217

第 27 章　数据结构自测题(二) …… 221

第 28 章　数据结构自测题(一)答案 …… 225

第 29 章　数据结构自测题(二)答案 …… 229

第一部分　学习辅导

第1章　绪　　论

目前，计算机已经广泛而深入地应用到了人类社会的各个领域，计算机处理的对象也由单纯的数值对象发展到字符、声音、图像等各种各样具有不同结构的数据，为了有效地组织和管理数据，设计出高质量的程序，必须深入分析和研究数据对象自身的特性，以及各数据对象之间的关系，这正是数据结构这门课程形成和发展的背景。

1.1　学习指导

1.1.1　知识结构分析

本章从数据结构的定义和有关术语入手，对数据结构所包含的逻辑结构、存储结构、数据的运算三方面内容进行了分析；介绍了抽象数据类型的含义和描述方法、算法的概念、性质以及效率度量的方法。在本章学习的概念和方法，是贯穿于全书的学习基础。为了便于学习，先将本章的知识点进行归类，列出的知识结构图如图 1.1 所示。

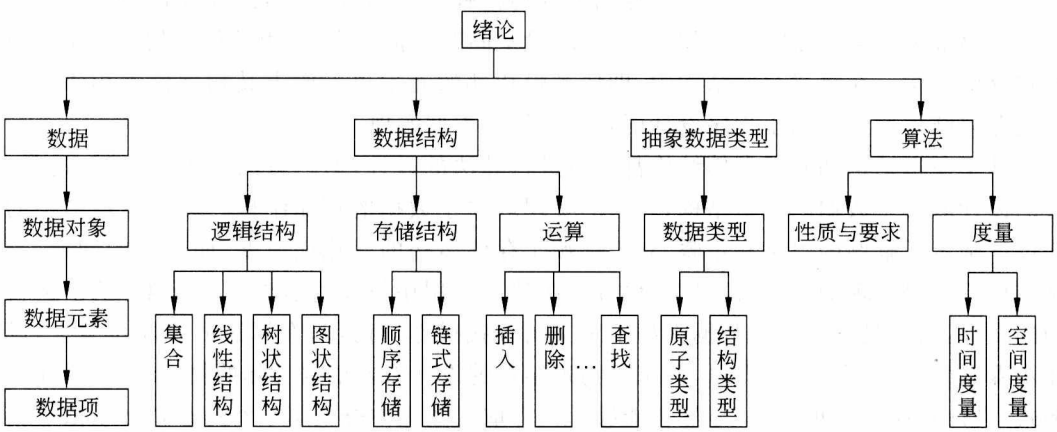


图 1.1　绪论知识结构图

1.1.2　重难点解析

1. 数据结构定义及其内容

在实际应用中，数据元素通常都不是孤立存在的，彼此之间存在着这样或那样的关系，

这种数据元素之间的关系称为结构。而数据结构就是指相互之间存在一种或多种特定关系的数据元素的集合，或者简称为带结构的数据对象。

根据数据元素间关系的不同特性，通常有下列四类基本的结构：

(1) 集合。在集合中，数据元素间的关系是属于同一个集合，别无其他关系。集合是元素关系极为松散的一种结构。

(2) 线性结构。在线性结构中，数据元素的直接前驱和直接后继之间存在着一对一的关系。

(3) 树状结构。在树状结构中，数据元素的前驱和后继之间存在着一对多的关系。

(4) 图状结构。在图状结构中，数据元素的前驱和后继之间存在着多对多的关系，图状结构也称为网状结构。

上述的数据结构有两个要素：一个是数据元素的集合，另一个是关系的集合。因此在形式上，数据结构通常可以采用二元组来表示，定义形式如下：

$$\text{Data\_Structure}=(D,R)$$

其中，D 是数据元素的有限集，R 是 D 上关系的有限集。

在这些结构中描述的关系是指数据元素之间的逻辑关系，因此称为逻辑结构。为解决实际问题，上述逻辑结构的信息应该在计算机中标识(又称映像)出来，此时体现的是数据结构在计算机中的实现方法，包括数据结构中元素的表示及元素间关系的表示，称为数据的物理结构，或称为存储结构。数据的存储结构可采用顺序存储或链式存储两种方法。

顺序存储方法是借助数据元素在存储器中的相对位置来表示数据之间的逻辑关系，由此得到的存储结构称为顺序存储结构。顺序存储结构是一种最基本的存储方式，特别是线性结构，如果采用顺序存储方法，那么逻辑相邻的数据元素，物理位置也相邻。顺序存储结构通常借助于程序设计语言中的数组实现。

链式存储方法是借助指示元素存储地址的指针来表示数据元素之间的逻辑关系，因此除了存储数据元素所必需的空间外，需要附设空间存放指针字段，由此得到的存储结构称为链式存储结构。在链式存储结构中，即使逻辑相邻的元素也不要求其物理位置相邻，显然，链式存储结构通常借助于程序设计语言中的指针类型实现。

基于逻辑结构和基于存储结构都可以进行算法的设计，但二者的着重点不同，通常，算法的设计取决于数据的逻辑结构，算法的实现取决于数据的存储结构。基于逻辑结构的算法设计确定了解题的思路和步骤，它不依赖于存储结构的变化而改变，而同样的问题，当存储结构不同时实现的策略就不同。例如：在某一应用中需要查找第 i 个元素，基于逻辑结构的设计，通常是根据抽象数据类型中定义的查找操作，直接写在算法的相应位置上，说明此步骤需要进行查找操作即可，而基于物理结构的设计需要依据实际的物理结构(顺序还是链式)设计相应的实现算法，显然在能够随机存取的数组中查找第 i 个元素和必须依靠指针指示的逻辑关系进行查找的链式方式上，二者的实现有很大的差别。

除了通常采用的顺序存储方法和链式存储方法外，有时为了操作的方便，还采用索引存储方法和散列存储方法。

2. 数据类型与抽象数据类型

数据结构定义了一组按某些关系结合在一起的数据元素。在程序设计语言中，每一个数据元素都属于某种数据类型。数据类型是一个值的集合和定义在这个值集上的一组操作

的总称，可以认为数据类型是在程序设计中已经实现的数据结构，而现实中的数据类型不足以直接表达数据结构中的所有元素类型，因而提出了抽象数据类型，抽象数据类型是指一个数学模型以及定义在该模型上的一组操作。在程序设计过程中，抽象数据类型的实现总是借助于编程语言所提供的数据类型。所以数据类型和抽象数据类型本质上是一个概念，但抽象数据类型的范畴更广。数据结构正逐步使用抽象数据类型描述，以便与面向对象的程序设计紧密地结合起来。

3. 算法的概念

对现实问题的解决通常需要如下的过程：对具体的问题进行分析，从中提取出操作的对象，找出操作对象间的关系，并用数学的语言加以描述——即建立数学模型，然后设计该数学模型的算法，最后编出程序，进行调试、测试，直至得到结果，而其中的算法就是指对特定问题求解步骤的一种描述，它是指令的有限序列，其中每一条指令表示一个或多个操作。数据结构中的算法通常用高级语言来描述和实现，例如目前比较流行的是 C 语言、C++ 语言、Java 等，但使用的方式与标准语言有一定的区别，它更着重于算法设计思想的描述，省略了很多的细节，所以算法不是程序，算法往往要经过必要的修改才能变成程序在计算机上运行。

4. 算法的度量

同一个问题往往具有多种解决算法，在这些算法中选择最终的解决方案需要对算法进行度量和评价，通常从“时间”和“空间”两个方面来分析和评价算法的效率。

由于影响程序运行时间的因素很多，所以估算算法的效率不能采用诸如秒、微秒、纳秒等具体时间，而是利用算法与输入规模之间的关系进行评估，即用一定规模的数据作为程序的输入，描述此时运行程序所需要的基本操作的数量级作为时间效率的度量——即算法的渐进时间复杂度，简称时间复杂度。例如：如果数据规模 n 与运行时间 t 存在一个线性关系，$t=c\times n$，那么当输入数据增加后，运行时间也增加同样的倍数，此时称时间复杂度为线性阶，记为 $O(n)$。

对算法时间效率进行度量的另外一种方法是频度，频度是指某条语句的运行次数。例如：

```
for(i=0,s=0;i<n;i++){s=s+i;}
```

其循环体的执行次数为 n 次，显然，重复执行的次数与时间复杂度是成正比的。

算法中语句的频度有时不仅与问题规模有关，还与所输入的元素值相关。例如起泡排序中待排序的数据序列的正序和逆序，时间复杂度差别较大。

时间复杂度按数量级递增排列依次为：$O(1)<O(\log_2 n)<O(n)<O(n\log_2 n)<O(n^2)<O(n^3)$。

指数时间的排列关系依次为：$O(2^n)<O(n!)<O(n^n)$。

算法的空间复杂度是对算法所需要的存储空间的量度。

对于同等复杂程度的程序，一般时间效率越高的程序，占用的内存就越大，空间效率就越低；反之时间效率低的程序，占用的内存相对就小，空间效率就越高，两者具有一定的矛盾性。

算法的时间复杂度和空间复杂度合称为算法复杂度。随着内存容量的不断增大，往往会牺牲空间性能来提高时间性能，因此在目前计算机硬件条件下，决定哪种算法更合适的主要因素是算法的时间复杂度。

1.1.3 学习方法指导

1. 数据、数据元素、数据对象

数据就是指能够被计算机识别、存储和加工处理的符号的总称，例如数字、文字、声音和图像等。

数据元素是数据的基本单位，在程序中作为一个整体进行处理。不同的问题数据元素的复杂程度不同，例如判断一个数是否为质数，数据元素只是一个整数；设计“五子棋”程序，数据元素是所有的棋盘格局，此时数据元素由若干个数据项组成，分别表示棋盘落子状态、落子位置等，较为复杂。

数据对象是性质相同的数据元素的集合，是数据的一个子集。

例如，在图书管理系统的“读者借书”子系统中，涉及三个数据对象：图书、读者、管理员，每个数据对象中都有不同的数据元素，每个数据元素包含不同的数据项，图书的数据项包含图书编号、书名、作者、出版社、出版时间、价格等，读者的数据项包含读者编号、读者姓名、办证时间等，管理员包括管理员编号、姓名、密码、权限等，数据项是数据不可分割的最小单位。

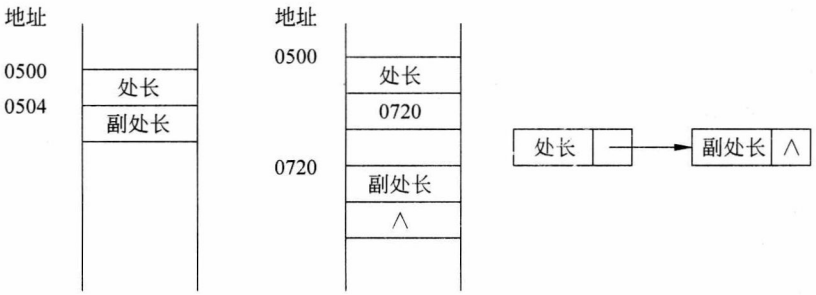
2. 对比分析逻辑结构和存储结构的差别

数据结构包括数据的逻辑结构、数据的存储结构和数据的运算三个方面，逻辑结构描述的是数据之间相互联系的方式，存储结构描述逻辑结构中涉及的数据以及数据之间的关系在计算机中存储的方法，数据的运算是对数据进行的操作处理。例如父子关系可以看成是一种逻辑关系，但二人不一定生活在同一个地方，他们所处的位置就属于存储结构，找到他们的住处就是一个操作。

按照数据元素的前驱与后继的关系，逻辑结构有四种类型：集合、线性结构、树状结构和图状结构，分别对应着数据元素之间“没有任何关系”、“一对一关系”、“一对多关系”和“多对多关系”，揭示了数据之间相互关系的一般性质。存储结构建立了从逻辑关系到空间关系的一种映射，分为顺序存储映射和链式存储映射，所以逻辑上相邻的数据可以存储到连续的空间，也可以存储到不相邻的空间，但数据之间的内在关系保持不变。例如：某单位有处长和副处长各一人，在逻辑上属于线性结构，但他们的办公地点(即存储结构)可以是相邻的两个办公室，也可以不相邻，即将二人表示到计算机中可以有顺序和链式两种方法，在顺序存储方法中，可以开辟一个包含两个元素的数组，存放两个元素，其逻辑关系与存储的物理位置关系相同，如图 1.2(a)所示；也可以采用链式存储结构，存放到不同的位置上，如图 1.2(b)所示，此时每个元素都要额外开辟指针空间，利用指针表示其逻辑关系，如图 1.2(c)所示。

3. 通过数据类型的概念深刻理解抽象数据类型

数学中有整数，程序设计语言中也有“整型”数据类型，二者含义虽然相同，但后者却有更细的分类，以 C 语言为例，有短整型、整型和长整型三种，不同的整型数据占用的空间不同，也意味着表达数据的范围不同，1 个字节的短整型只能表示 256(即 2^8)个整型数据，而 2 字节的整型数据能表达 65 536(即 2^{16})个整型数据，4 个字节的长整型能表示 2^{32} 个整型数据，所有的整型都能进行加、减、乘、除、取余操作。与此相似，程序设计语言中的“实型”数据也分为单精度、双精度等多种，每种类别所表示的数据范围不同，所有的实型数据都能进行



(a) 顺序存储方式　　(b) 链式存储方式　　(c) 链式存储逻辑表示

图 1.2　存储示意图

加、减、乘、除操作，因此数据类型实质上是一个值的集合以及在这些值集上定义的一组操作的总称。

数学中有“复数”类型，但大多数程序设计语言中并没有复数类型，所以不能直接定义变量进行相关运算，如果要设计进行复数运算的程序，就必须先定义“复数类型”。对于这个“创造”出来的数据类型，需要先对其进行必要的说明——既要说明复数在程序设计中表达的方式和表示的范围(对应数据类型中“值的集合”)，还要设计进行的运算(对应数据类型中“操作的总称”)。因此抽象数据类型(Abstract DataType，简称 ADT)是指一个数学模型以及定义在该模型上的一组操作。

抽象数据类型和数据类型实质上是一个概念，但抽象数据类型的范围更广泛，可以用已有的数据类型实现抽象数据类型。实际上抽象数据类型往往不会只由一个基本类型数据组成，因而抽象数据类型也可以看做是“由一种数据结构和定义在其上的一组操作”组成，而数据结构又包括数据元素及元素间的关系，因此抽象数据类型一般可以由数据元素、数据关系及基本操作三种要素组成的(D,S,P)三元组表示，其中，D 是数据对象，S 是 D 上的关系集，P 是对 D 的基本操作集。形式如下：

```
ADT 抽象数据类型名 {
    数据对象：<数据对象的定义>
    数据关系：<数据关系的定义>
    基本操作：<基本操作的定义>
} ADT 抽象数据类型名
```

其中，数据对象和数据关系的定义用伪代码描述，基本操作的定义格式如下：

```
基本操作名(参数表)
    初始条件：<初始条件描述>
    操作结果：<操作结果描述>
```

抽象数据类型的特征是使用与实现相分离，实行封装和信息隐蔽。这与面向对象的程序设计思想是一致的。

4. 算法和程序的差别

进行程序设计时必须先分析问题，然后提出解决问题的步骤，以求 $n!$ 为例，求解步骤如下：

```
long  fact(int n)
{
    i=2; t=1;
    while(i<=n)
    {
        t=t * i;
        i++;
    }
    return t;
}
```

上面这段类 C 语言代码就是算法，算法是描述特定问题求解步骤的计算机指令的有限序列，即解决该问题的步骤和方法，可以用自然语言描述，也可以用伪计算机语言描述，不必完全符合计算机语言的规范。

程序则是根据分析出的算法编写出符合编程语言语法规范的代码，用来编译执行而得到结果。例如：根据上段算法，编写的 C 语言程序如下：

```
#include <stdio.h>
long  fact(int n)
{
    int i;
    long t;
    i=2; t=1;
    while(i<=n)
    {
        t=t * i;
        i++;
    }
    return t;
}
```

实际上，换用 Pascal、Fortran 等其他编程语言实现同样的功能时，算法依然相同，只不过是不同程序设计语言的语法规则不同，表现形式不同，所以算法是程序设计的核心。通常算法应该满足如下五个特性。

(1) 有穷性：任何一条指令都只能执行有限次，即算法必须在执行有限步后结束。

(2) 确定性：算法中每条指令的含义必须明确，不允许有二义性。

(3) 可行性：算法中待执行的操作都是可以实现的。

(4) 输入：一个算法的输入可以包含零个或多个数据。

(5) 输出：算法有一个或多个输出。

算法的性质不一定适用于程序，例如：操作系统程序，只要没有外力中断，可以无限的运行下去。

设计一个“好算法”通常有四个目标。

(1) 正确性：算法应能满足设定的功能和要求。

(2) 可读性：思路清晰、层次分明、易读易懂。

(3) 健壮性：输入非法数据时应能作适当的反应和处理。

(4) 高效性和低存储量：高效性(时间复杂度)指解决问题时间越短，算法的效率就越高；低存储量(空间复杂度)指完成同一功能，占用存储空间应尽可能少。

1.2 典型例题

例 1(解析题) 设数据的逻辑结构定义为 $T=(D,R)$其中：

$D=\{a,b,c,e,f,g,h\}$

$R=\{(a,b),(a,c),(b,e),(b,f),(b,g),(c,h)\}$

请画出该结构的逻辑图。

分析：本题比较简单，主要考查数据结构中逻辑结构的概念，其中 D 是数据元素的集合，R 是元素间关系的集合，需要注意的是集合中关系的表达为无序。

解：如图 1.3 所示。

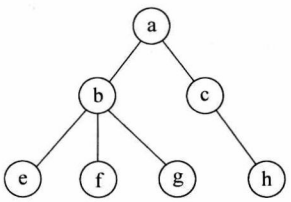


图 1.3 逻辑结构图

例 2(解析题) 定义复数的抽象数据类型。

分析：抽象数据类型是在已有数据类型的基础上，定义一组数据和施于这些数据上的一组操作，其定义包括数据对象、数据关系和基本操作三方面的内容；另一方面复数由实部和虚部组成，可以分别用两个实型数据描述，两个实数逆序时代表不同的复数；抽象数据类型的基本操作通常都包括初始化和销毁，根据复数的特点，还可以设计取实部数据、虚部数据、复数相加、相减、相乘、相除等操作。

解：

```
ADT Complex {
  数据对象：D={e1,e2|e1,e2∈RealSet}
  数据关系：R1={<e1,e2>|e1 是复数的实数部分，e2 是复数的虚数部分}
  基本操作：
    InitComplex(&Z, v1, v2)
      操作结果：构造复数 Z，其实部和虚部分别被赋以参数 v1 和 v2 的值。
    DestroyComplex(&Z)
      操作结果：复数 Z 被销毁。
    GetReal(Z,&RealPart)
      初始条件：复数已存在。
      操作结果：用 RealPart 返回复数 Z 的实部值。
    GetImag(Z,&ImagPart)
      初始条件：复数已存在。
      操作结果：用 ImagPart 返回复数 Z 的虚部值。
    Add(Z1,Z2,&Sum)
      初始条件：Z1、Z2 是复数。
      操作结果：用 Sum 返回两个复数 Z1、Z2 的和值。
    Sub(Z1,Z2,&Result)
      初始条件：Z1,Z2 是复数。
```