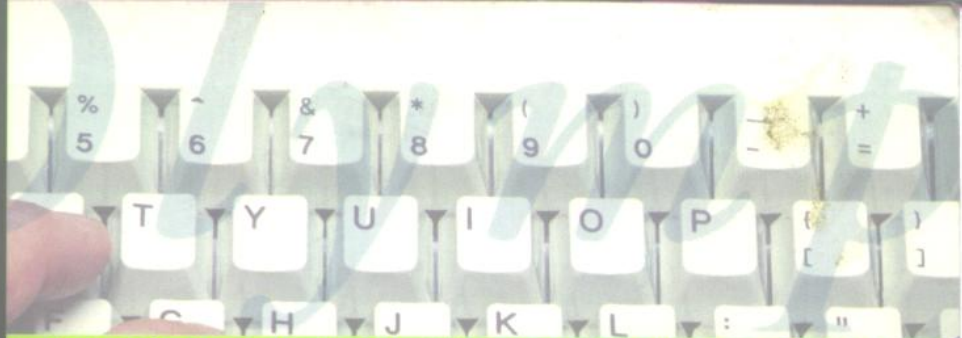




全国青少年信息学（计算机）奥林匹克丛书

数据结构 及其应用

中国继续教育联合学院推荐用书

郭嵩山 黄叶亭 黄国洪 阮志远 编著
陈树楷 吴文虎 王晓敏 审校

Computer

Computer



人民邮电出版社
PEOPLE'S POSTS &
TELECOMMUNICATIONS
PUBLISHING HOUSE

Olympic

TP311.12
GSS/1

全国青少年信息学(计算机)奥林匹克丛书
中国继续教育联合学院推荐用书

数据结构及其应用

郭嵩山 黄叶亭 编著
黄国洪 阮志远
陈树楷 吴文虎 王晓敏 审校



人民邮电出版社

047723

全国青少年信息学(计算机)奥林匹克丛书

中国继续教育联合学院推荐用书

数据结构及其应用

JSSB2/30

◆ 编 著 郭嵩山 黄叶亭 黄国洪 阮志远

◆ 审 校 陈树楷 吴文虎 王晓敏

◆ 责任编辑 陈 昇

◆ 人民邮电出版社出版发行 北京崇文区夕照寺街 14 号

北京鸿佳印刷厂印刷

新华书店总店北京发行所经销

◆ 开本:787×1092 1/16

印张:13

字数:330 千字

1998 年 7 月第 1 版

印数:1-5 000 册

1998 年 7 月北京第 1 次印刷

ISBN 7-115-07157-8/TP·702

定价:19.00 元

本书共分为 8 章,系统地介绍了数据结构的有关知识。其中第 1 章介绍了数据结构的有关概念和术语;第 2 章介绍了数组、线性表、堆栈、队列和串等数据结构及相应的操作;第 3 章介绍了递归的概念及其在程序设计中的应用;第 4 章介绍了线性链表及其在程序中的应用;第 5、6 章分别介绍了两种典型的非线性数据结构——树和图;第 7 章着重介绍四种典型排序算法原理及其应用,并在此基础上介绍了堆排序;第 8 章列举了 8 个典型的数据结构的综合应用例子。

本书叙述简明扼要、深入浅出。全部例子用 Turbo Pascal 编写,而且所有程序均上机调试通过并附有测试数据及运行结果,使读者更易阅读。本书可作为广大中小学计算机教师继续教育的培训材料及信息学奥林匹克竞赛活动用书,也可作为大专院校有关计算机专业师生的参考书。

全国青少年信息学(计算机) 奥林匹克丛书编委会

顾 问： 陈树楷

吴文虎

潘懋德

主 任： 曹志超

副主任： 徐修存

李树岭

孙彦德

编 委： 李树奎

马启洲

李立新

蒋新儿

蔡绍稷

王晓敏

郭嵩山

张 明

孔 军

吴再陵

张桂林

黄叶亭

黄国洪

阮志远

吕晓春

今日世界,无论是科技竞争、经济竞争,还是综合国力竞争,归根到底是人才竞争。

世界各国都把科学技术作为综合国力竞争的经济热战主要武器,争先占领 21 世纪的技术制高点。各国政府越来越重视科学技术的作用,愈来愈重视人才的培养。许多国家相继制定了加强大学本、专科教育,抓好中小学教育和成人教育的措施,并提出了以做好继续教育为重要手段的人才第一战略,积极开发人才资源。

党中央和各级政府历来重视教育、培养人才工作。中央提出了科教兴国和可持续发展的战略方针,并一再强调科学教育的发展关系到全民素质的提高和中国现代化的前途。邓小平同志说过:“改革经济体制,最重要的、我最关心的,是人才。改革科技体制,我最关心的还是人才。”江泽民主席提出“严重的问题在于教育干部”深得人心。“要把培训教育作为一种开发,作为一种投入,是一种适应竞争需要的学习竞赛,要树立终身培训的观念。”已成为许多部门领导的共识。

随着我国社会主义市场经济的发展和与国际市场的逐步接轨,经济体制从传统的计划经济体制向社会主义市场经济体制转变,经济增长方式从粗放型向集约型转变这两个根本转变,急需大量专业人才。这使得人们对继续教育给与更大的重视,各类培训中心如雨后春笋般地冒了出来,既处于有利的发展机遇,又面临着十分繁重的任务。我国专业技术人员 2700 多万人,仅占我国人口总量的 2.3%,与经济建设关系密切的自然科学技术人员只占人口总量的 1.5%,比中等发达国家 80 年代的水平还差 2-3 倍。我国专业技术人员在产业间分布不合理,作为社会财富直接创造者的企业和农业领域尚未成为人才资源配置的重点,知识结构与现代经济发展需要差距较大,相当一部分专业技术人员老化,缺乏创新意识和创造能力,也缺乏市场经济知识与管理能力。高新技术人才、高级管理人才以及新兴行业的专业技术人员严重短缺,学术带头人年龄偏大。这种状况直接影响着我国国民经济与社会发展,直接影响着我国跨世纪宏伟目标的实现。要改变这种状况,继续教育起着十分关键的作用。总之,为实现我国社会主义发展宏伟战略目标,为造就千百万高素质的专业技术人员队伍,继续教育任重而道远。

信息科学技术学的需求更加突出。专业技术人员、广大教师都需要信息科学技术学科方面的继续教育,同时与对青少年计算机普及教育相联系,责任就更加重大。我们的工作符合专业技术人员的要求,符合市场经济对人才的需求,也符合现代化建设的需求。中国科协及其所属学会开展的学术交流、科学普及等各项工作,说到底,还是为了培养人才。从这一角度上看,直接培养人才的继续教育工作对学会来说,更是至关重要。

中国计算机学会有了 10 多年继续教育工作的实践,从 1986 年开办各类专门课程培训班,至今已有 12 年历史,累计开办了 20 多种培训班,接受培训人数约为 10000 人次。编辑出版丛书共 5 套 26 本,发行 20 多万册。还开展了《奥林匹克竞赛与计算机教学改革》、《计算机促进结构性教育改革》等软课题研究。

本套丛书的系列定名为:《全国青少年信息学(计算机)奥林匹克丛书·中国继续教育联合学院推荐用书》是有意义的。该丛书是在已使用了两年的内部讲义基础上,经进一步修改审定后,先出版四本,另外两本待再版时补齐,这是中国计算机学会普及工作委员会对质量审慎的表现。

我们期望着这套丛书能对广大各课师资参加计算机继续教育的学习有所帮助;能对于广大青少年参加全国计算机奥林匹克分区联赛有所帮助。我还希望读者能对这套丛书提出宝贵修改意见,指出不当之处。

最后,我还要感谢诸位顾问、编委和人民邮电出版社有关人员的辛勤工作,使得这套丛书能在极短的时间与广大读者见面。

中国科协副主席
中国继续教育联合学院院长



1998 年 2 月 12 日于北京

90年代以来计算机技术已经在我国广泛应用于社会的各个领域,计算机已成为现代化建设必不可少的重要工具。掌握计算机知识,提高计算机应用能力,是时代对每个青少年提出的要求。不懂得计算机的应用,将难以承担21世纪国家现代化建设的重任。

为了更好地在广大青少年中普及计算机知识,培养适应21世纪建设需要的人才,首先要不断提高广大中小学计算机教师的水平,帮助他们不断进行知识的更新。这套由中国计算机学会组织的计算机教师继续教育的培训教材,正是为适应这种需要而编写的。

本书作为这套教材的一部分,系统地介绍数据结构方面的有关知识。其中第1章介绍数据结构有关概念和术语,数据结构的分类和学习数据结构的意义;第2章重点介绍了数组、线性表、堆栈、队列和串等数据结构及相应的操作;第3章介绍递归的概念及其在程序设计中的应用;第4章介绍了线性链表,并对单向、双向线性链表、循环链表等链表结构及其应用作了详细讲述;第5、6章分别介绍了两种典型的非线性数据结构树和图,其中第5章着重介绍了二叉树,并详细讲解了二叉树的有关操作及其在程序设计中的应用,第6章着重介绍了图的概念及其存储结构,并用较多篇幅介绍了在图运算中应用比较广泛的深度优先搜索和广度优先搜索两种典型算法;第7章着重介绍4种典型排序算法原理及其应用,并在此基础上介绍了堆排序;第8章举了8个典型例子,作为全书前几章数据结构的综合应用例子。

本书的一个重要特点是对有关概念及结构的描述简明扼要、深入浅出,并通过比较多的例题分析,帮助读者加深对原理的理解。全书的例题全部用Turbo Pascal编写,所有程序均上机通过。并附有测试数据及运行结果。为了帮助读者更易阅读程序清单,我们除了在解题思路及题目分析方面指出其所用算法外,还对程序清单作了详细的注释。此外,我们在第8章的综合应用例题的算法描述中采用了逐步求精的方法来描述,以便于读者理解和学习。

本书由郭嵩山主编,黄叶亭、黄国洪、阮志远编写,其中郭嵩山负责全书的组

织策划及第 1 章编写,阮志远负责第 2-3 章编写,黄叶亭负责第 4-6 章编写,黄国洪负责第 7-8 章编写。全书最后由郭嵩山全面修改、补充及统稿。

本书可作为广大中学计算机教师继续教育的培训教材,也可作为大专院校有关计算机专业学生学习相关课程的教学参考书及信息学奥林匹克竞赛活动用书。

本书曾以讲义形式使用,受到读者的好评和鼓励,我们在原稿基础上再作了详细修改和补充。

由于编者水平所限,有疏漏及不当之处,敬请读者批评指正。

编者

1998 年 2 月

●第1章 概述	1
1.1 什么是数据结构	1
1.2 数据结构的分类	2
1.3 学习数据结构的意义	3
习题	4
●第2章 线性表	5
2.1 数组	5
2.2 线性表的顺序存储结构和基本操作	7
2.3 栈	13
2.4 队列	15
2.5 串	18
2.6 应用实例	26
习题	45
●第3章 递归	47
3.1 递归定义与类型	47
3.2 递归应用举例	48
3.3 递归过程的模拟	52
习题	56
●第4章 线性链表	57
4.1 单向线性链表	57
4.2 双向线性链表	66
4.3 循环线性链表	71
4.4 链表的应用	73
习题	79
●第5章 树	81
5.1 树及其存储结构	81
5.2 二叉树	83
5.3 树的应用	91
习题	102

●第6章 图	105
6.1 图的基本概念	105
6.2 图的存储结构	109
6.3 图的运算及其应用	115
习题	147
●第7章 排序	149
7.1 插入排序	150
7.2 选择排序	152
7.3 冒泡排序	154
7.4 快速排序	156
7.5 堆排序	159
习题	164
●第8章 综合应用	165

1.1 什么是数据结构

从 60 年代末开始,程序设计已从技巧发展成一门科学。与程序设计联系最密切的课题有数据结构、算法分析与设计和程序设计方法。著名的计算机科学家沃斯(N. Wirth)提出了“算法 + 数据结构 = 程序”的著名论点,简洁地概括了程序的组成。那么,什么是数据结构呢?让我们先从数据说起。

数据是程序加工的原始材料,它可能是数字、字符或由它们组成的字符串;它也可能是采样后的物理量,例如电压、电流等电信号通过模-数转换器(A/D)输出变成计算机可以接受的数字信息;或是从磁带、磁盘或光盘上读出一串二进制数表示的数字、字符或图形的信息;或是从调制解调器(MODEM)上将电话的声信号转换成计算机可以接受的格式;或是通过按键,从键盘上输入到计算机的信息……简言之,数据是描述客观事物的数字、字符以及所有能输入到计算机、能被计算机进行处理的信息的集合。也就是说,数据是符号的集合,是计算机要处理的信息的集合。从本质上来说,数据是客观事物表示的一种抽象结果,而数据结构这门课程就是研究如何把客观世界待处理的信息逐层抽象成计算机可以接受的某种形式。通俗地说,数据结构就是研究数据之间的相互关系,也就是数据的组织形式的一门科学。

所谓算法是指解题方法的精确描述,它是有穷处理动作的序列。

数据结构和算法有着密切的联系,数据结构是建立在算法的基础上,而选择什么样的数据结构对于程序设计来说,是至关重要的决策,它直接影响到程序的效率。选择一个合适的数据结构便很容易形成一个简洁有效的算法;否则,如果数据结构选择不好,除了影响程序开发速度之外,更重要的是影响设计出来程序的运行效率。

下面介绍一些常用的基本术语。

数据(Data)是信息的载体,它能够被计算机识别、存储和加工处理。

数据元素(Data Element)是数据的基本单位。有时候,一个数据元素可以由若干个数据项(data item,或称字段、域)组成,数据项是具有独立含义的数据最小单位。

数据结构(Data Structure)可以看作是相互间存在着某种(或多种)特定关系的数据元素的集合。这些特定关系包括:

- (1) 数据元素之间逻辑关系,即数据的逻辑结构(Logical Structure)。
- (2) 数据元素及其关系在计算机存储器中的存储方式,即数据的存储结构(Storage Structure),也就是数据元素的物理结构。

(3) 施加在该数据上的操作,即数据的运算。常用的运算有:检索、插入、删除、更新、排序等,这些运算实质上是在抽象的数据上所施加的一系列的抽象的操作。

数据类型(Data Type)是一个值的集合和定义在这个值集上的一组操作的总称。数据类型最早出现在高级语言中,例中,PASCAL 语言中的整型、实型、字符型、布尔型、枚举类型、子界类型和指针类型,以及构造类型中的记录类型等。

数据结构研究的对象是各种数据的表示、数据的类型以及它们之间关系的集合。它研究的范围主要包括:各种数据结构的性质,即它们的逻辑结构和物理结构以及施加在其上的操作。由于数据结构和算法有着十分密切的关系,所以,我们在本书中除主要介绍数据结构的有关概念及其基本操作外,还介绍了在这些数据结构上设计的有关算法。

1.2 数据结构的分类

数据结构的类型可以从不同的角度来划分。若从数据元素的值在使用时具有不可分割的性质或是它可以由更基本成份所组成这个角度来划分,数据结构可分成简单类型和构造类型两大类;而如果从数据所占据的内存空间在程序执行期间是否会发生变化这个角度来划分,又可以分成静态和动态结构两大类。表 1.1 列出了常见的数据结构的分类。

表 1.1 数据结构的分类

简单类型	整型 实型 字符型 布尔型 枚举型 子界型 指针型	静态结构
构造类型	数组 记录 集合	
	文件 线性表 栈 队列 链表 树 图	动态结构

通常高级程序设计语言都提供各种简单类型数据结构和静态的构造类型的数据结构。例如,PASCAL 就提供了 11 种类型的定义,这 11 种除了文件属于动态结构的构造类型外,其余 10 种都属于简单类型和静态构造类型。

在表 1.1 的数据结构中,像数组、线性表、栈、串和队列等数据结构属于线性的数据结构,而像树和图这样的动态数据结构属于非线性的数据结构。使用线性数据结构来表示各结点之间的联系,这些结点内在的存放显得比较简单;而采用非线性数据结构往往更能形象地反映出各结点之间的层次关系。

在以下各章中,我们将分别介绍数组、线性表、栈、串、队列、链表、树和图的概念、基本操作及其在程序设计中的应用。

1.3 学习数据结构的意义

“数据结构”是计算机专业学生的专业基础课之一,是十分重要的核心课程。计算机的所有系统软件和应用软件都要用到各种类型的数据结构。要想更好地运用计算机来解决实际问题,仅仅学习计算机语言而缺乏数据结构的知识是远远不够的,而打好“数据结构”这门课程的扎实基础,对于学习计算机专业的其他课程,如操作系统、编译原理、数据库管理系统、软件工程、人工智能等都是十分有帮助的。

随着计算机应用领域不断扩大,非数值计算问题占据了当今计算机应用的绝大多数,简单的数据类型已经远远不能满足需要,各数据元素之间的复杂联系已经不是普通数学方程式所能表达的了。因此,掌握好数据结构方面的知识,对于提高我们解决实际问题的能力将会有莫大的帮助。实际上,一个好的程序无非是选择一个合适的结构和好的算法,而好的算法的选择很大程度上取决于描述实际问题的数据结构的选取。所以,学好数据结构,将是进一步提高我们程序设计水平的关键之一。通常我们在程序设计时,所遇到的首要问题是:选择什么样的数据结构才合适呢?这个问题十分关键。下面,我们以本书各章例题中所举的一些典型题目用到的数据结构作一导引。

例如,第二章例 2.1 是一个矩阵乘法的问题。由于矩阵中每个数据都具有相同的类型,我们便想到可以用一个二维数组来存放矩阵中的元素。但在分析题目时我们会发现,这两个矩阵 $A(M \text{ 行 } N \text{ 列})$ 和 $B(N \text{ 行 } H \text{ 列})$ 中 M 、 N 、 H 的值需要由键盘输入确定。如果我们将本来用二维数组比较直观地存放矩阵中元素改为用一维数组来存放,就可以使程序设计更为简化。又如对例 2.2 的学生成绩表进行插入、修改、删除、追加等操作,解题时我们将整个学生成绩表看成一个线性表,将每个学生记录看作是表中的一个元素,这样设计的数据结构也使我们所编的程序更简洁。对于像表达式计算类的题型(如例 2.3),我们通常选择栈这种数据结构,利用栈的后进后出的特点,将中缀算术表达式转换成后缀表达式。在实际设置时,需要建立两个栈,一个是数栈,用来存放运算的对象(操作数),而另一个是符号栈,用来存放运算符及左括号。在例 6.5 和例 8.6 中都举了一个要求找出象棋棋盘上“马”的周游路线的例子。对于这些题,我们也是采用栈这种数据来记录“马”跳过的每个方格的坐标位置,而最后入栈的方格位置就是用回溯时应退回的位置;在例 8.6 中我们用解答树列出“马”可能跳到的所有位置(结点)。又如对银行营业活动模拟、医生诊病模拟、机场航机等类型题目,通常都可以通过设置队列这种数据结构来求解,例 2.5 就是这类活动模拟的一个例子。例 4.10 是一个著名的约瑟夫问题,对这类题,我们采用单向循环链表来处理;像例 8.1 的纸牌游戏也属于这类问题。对于处理字符串等类型的题目(如例 4.13),通常使用链表来存放字符串,这样便于对字符串进行插入和删除操作。又如对于像例 4.14 的多项式运算类型题目,可以采用单向链表这种数据结构,其每个结点包含有三个域,它们分别用来存放多项式的每项的系数、指数及指示下一结点(后继结点)的指针。有的经典题目,例如称假金币问题(例 5.4),通常用判定树来处理;又如博弈问题(例 5.5),通常是建立一棵博弈树;对于像例 8.2 这类单词排序问题,采用二叉排序树来求解。此外,分析问题时还经常利用树形结构列出排列组合类问题所有可能的解,这就是所谓的解答树,例如,例 5.7 的根据各人工作效率分配工作问题和例 5.8 的背包问题就是运用解答树帮助我们在编程时能够不遗漏地找出所有可能的解。在图的应用方面,像例 6.9 的求最

短路径问题、例 8.5 的邮递员问题等,都是图运算中典型问题。图的运算通常又与搜索算法紧密相连,因此,如何建立图的存储结构,对于提高搜索效率是十分重要的。例如,例 6.6 的填数问题、例 6.7 的商人和生番渡河问题采用图的深度优先搜索算法来处理;像例 6.10 的分酒问题、例 6.11 的八数码问题则采用图的广度优先搜索算法来处理。

从以上分析不难得出下面的结论:学好“数据结构”这门课程,灵活运用数据结构的知识,对于提高我们自身的程序设计水平是十分重要的。

习 题

1-1 解释下列术语:

数据 数据元素 数据项 数据结构 算法

1-2 试述数据结构的分类。

1-3 试扼要说明学习数据结构的意义。

线性表

线性表(Linear List)是最常用且比较简单的一种数据结构,它是由有限个数据元素组成的有序集合,每个数据元素由一个或多个数据项组成。比如:26个英文字母的字母表(A,B,C,……Z)是一个线性表,表中每个数据元素由单个字母字符组成数据项。又如图2.1所示的学生成绩表,它的每一行就是一个数据元素,它反映一个学生的成绩情况,而每个学生成绩情况又由姓名、语文、数学、总分、平均分五个数据项组成。

姓 名	语 文	数 学	总 分	平 均 分
王小冲	70	89	159	79.5
陈 红	80	75	155	77.5
⋮	⋮	⋮	⋮	⋮
张立红	90	89	179	89.5

图 2.1 学生成绩表

线性表具有如下的结构特点:

① 均匀性:虽然不同数据表的数据元素可以是各种各样的,但对于同一线性表的各数据元素必定具有相同的数据类型和长度。

② 有序性:各数据元素在线性表中的位置只取决于它们的序号,数据元素之间的相对位置是线性的,即存在唯一的“第一个”和“最后一个”的数据元素,除了第一个和最后一个外,其它元素前面均只有一个数据元素(直接前趋)和后面均只有一个数据元素(直接后继)。

在实现线性表数据元素的存储方面,一般可用顺序存储结构和链式存储结构两种方法。链式存储结构将在本书第四章中介绍,本章主要介绍用数组实现线性表数据元素的顺序存储及其应用。另外栈、队列和串也是线性表的特殊情况,又称为受限的线性结构。

2.1 数 组

几乎所有的程序设计语言都定义数组为一种固有的数据类型,事实上数组本身就具有线性表的性质。比如,一维数组,它的每个数组元素对应于一个下标;二维数组,它的每个数组元素对应于一对下标;对于 n 维数组,它的每个元素对应一组 n 个下标。而数组元素的数据类

型总是相同的,元素的有序性可由下标的顺序关系来确定。但是数组与一般的线性表不同,数组一般不作插入和删除操作,而通常只定义两种操作,即对于给定的下标读取和修改相应的数组元素的值。

例如:设 A 为一个一维数组,数组类型为 $atype$,

$var A : array[c..d] \text{ of } atype$

一维数组的每个数组下标对应线性表的一个元素,假设 $i \in [c..d]$,则有 $A[c]$ 为线性表的第一个元素, $A[d]$ 为线性表的最后元素, $A[i]$ 的直接前趋为 $A[i-1]$,直接后继为 $A[i+1]$,其中第一个元素只有直接后继,最后元素只有直接前趋。

它的每个元素在存储空间中的地址可通过下式计算:

$$loc(a_i) = loc(a_c) + (i - c) L$$

其中 $loc(a_i)$ 表示第 i 个元素 $A[i]$ 的地址, $loc(a_c)$ 为第一个元素 $A[c]$ 的地址,称为“基地址”, L 为每个数组元素所占用的存储单元数。设 $n = d - c + 1$,则各元素对应的存储单元的关系如图 2.2 所示。

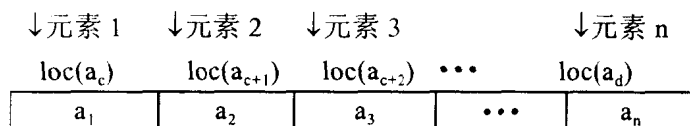


图 2.2 一维数组各元素位置与存储单元关系

又如:设 A 为一个二维数组,数组类型为 $atype$,

$var A : array[c_1..d_1, c_2..d_2] \text{ of } atype;$

其中 c_1, d_1 表示行下标的上界和下界, c_2, d_2 表示列下标的上界和下界。上式可理解为以行为主序用一维数组的形式定义的二维数组:

$var A : array[c_1..d_1] \text{ of } array[c_2..d_2] \text{ of } atype;$

此时, $A[c_1, c_2]$ 为首元素, $A[d_1, d_2]$ 为尾元素,若 $i \in [c_1, d_1], j \in [c_2, d_2]$,当 $c_2 < j < d_2$ 时 $A[i, j]$ 的直接前趋为 $A[i, j-1]$,直接后继为 $A[i, j+1]$;而 $A[i, c_2]$ 的直接前趋为 $A[i-1, d_2]$,直接后继为 $A[i, c_2+1]$; $A[i, d_2]$ 的直接前趋为 $A[i, d_2-1]$,直接后继为 $A[i+1, c_2]$,也即某行第一列元素的直接前趋为上一行最后一列的元素,而某行最后一列元素的直接后继为下一行第一列的元素,其中首元素只有后继,尾元素只有前趋。

如果每个数组元素占用 L 个存储单元,则二维数组任一元素的存储空间地址可通过下式计算:

$$loc(a_{i,j}) = loc(a_{c_1,c_2}) + [(d_2 - c_2 + 1)(i - c_1) + (j - c_2)]L$$

其中 $loc(a_{c_1,c_2})$ 为基地址。设 $n = d_1 - c_1 + 1, m = d_2 - c_2 + 1$,则各元素对应的行序存储关系如图 2.3 所示:

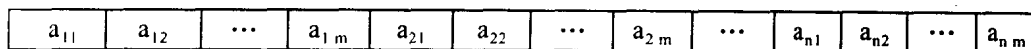


图 2.3 二维数组各元素的行序存储方式

例如:有一个 5×3 的矩阵,存放于数组 $A[3..7, 2..4]$ 中,矩阵中每个元素用 2 个字节存储,并设第一个元素的地址为 100,求 $A[5, 3]$ 元素的地址值。

分析:我们可以把数组 $A[3..7, 2..4]$ 按行序方式存储,从而构成一线性表,如图 2.4 所