

数据结构 学习指导

刘凌波 主编

河海大学出版社



数据结构学习指导

主 编 刘凌波
副主编 顾韵华
李 贞
简艳春

河海大学出版社

图书在版编目(CIP)数据

数据结构学习指导 / 刘凌波主编. —南京: 河海大学出版社, 2000.12

ISBN 7-5630-1517-5

I. 数… II. 刘… III. 数据结构-高等教育-自学考试-自学参考材料 IV. TP311.12

中国版本图书馆 CIP 数据核字 (2000) 第 88459 号

书 名 / 数据结构学习指导

书 号 / ISBN 7-5630-1517-5

责任编辑 / 代江滨

封面设计 / 张世立

出 版 / 河海大学出版社

地 址 / 南京西康路 1 号(邮编: 210098)

电 话 / (025)3737852(总编室) (025)3722833(发行部)

经 销 / 江苏省新华书店

印 刷 / 南京金阳彩色印刷厂

开 本 / 787 毫米×1092 毫米 1/16 14.875 印张 368 千字

版 次 / 2000 年 12 月第 1 版 2000 年 12 月第 1 次印刷

印 数 / 1-5000 册

定 价 / 18.00 元

前 言

数据结构是一门计算机专业基础课,它培养了学生在分析研究现有数据的特性的基础上,选择合适的数据结构和相应的算法,提高编写程序软件的能力。数据结构是学习操作系统、编译原理、数据库系统等课程的重要基础。

本书是在陈本林、陈珮珮、吉根林编著、南京大学出版社出版的《数据结构》一书的基础上,根据自学考试大纲编写的自学考试学习指导书,它适合于参加计算机专业(本科)自学考试人员和计算机等相关专业的学习者使用。

本书共九章,第一章介绍了数据类型、数据结构和算法的基本概念;第二章至第六章详细介绍了线性表、栈、队列、串、树和图等几种基本的数据结构类型,以及在這些数据结构上的操作处理方法;第七章至第九章介绍了在程序设计中经常遇到的排序和查找问题,分析了解决这些问题的方法及其算法实现。在本书的每一节中包括“本节要点”和“习题及分析”两部分,“本节要点”中介绍了这一节中的主要内容,在部分难点处进行了详细的分析,并举出了一些例题;在“习题及分析”中,根据自学考试大纲和题型配有本节相应的习题练习和答案,对一些有一定难度的习题还给出了提示信息,以便于学习者自我检测和提高。

本书由刘凌波、顾韵华、李贞、简艳春分工执笔,刘凌波对全书进行了修改和统稿。由于水平所限,书中难免会出现错误和不足,恳请广大读者及同仁给予批评和指正。

编 者

2000 年 11 月

目 录

前言	(1)
第一章 概论	(1)
1.1 数据类型和数据结构	(1)
1.2 基本数据结构	(4)
1.3 算法及算法分析	(12)
第二章 表	(17)
2.1 抽象数据类型表	(17)
2.2 表的实现	(22)
2.3 其他表结构	(35)
2.4 稀疏矩阵的链表表示	(43)
2.5 表的应用—动态存储管理	(47)
第三章 栈和队列	(52)
3.1 抽象数据类型栈	(52)
3.2 抽象数据类型队列	(55)
3.3 栈和队列的应用举例	(58)
3.4 递归数据结构	(62)
第四章 串	(65)
4.1 抽象数据类型串	(65)
4.2 串的实现	(68)
4.3 串的模式匹配	(77)
第五章 树	(81)
5.1 树的基本概念	(81)
5.2 二叉树	(84)
5.3 二叉树的遍历	(93)
5.4 线索树	(105)
5.5 树和森林	(113)
5.6 树的应用	(117)
第六章 图	(131)
6.1 图的概念	(131)
6.2 图的存储表示	(134)
6.3 图的遍历	(141)
6.4 最小代价生成树	(145)

6.5	最短路径	(151)
6.6	拓扑排序	(155)
6.7	关键路径	(158)
第七章	内排序	(161)
7.1	插入排序	(161)
7.2	冒泡排序	(166)
7.3	选择排序	(169)
7.4	希尔排序	(172)
7.5	快速排序	(174)
7.6	堆排序	(179)
7.7	基数排序	(182)
7.8	归并排序	(186)
第八章	查找	(192)
8.1	表的查找	(192)
8.2	散列技术	(194)
8.3	二叉查找树	(197)
8.4	平衡二叉树	(202)
8.5	Tries	(203)
8.6	B 树和 B ⁺ 树	(204)
第九章	外排序	(207)
9.1	外存储器	(207)
9.2	初始归并段的生成	(207)
9.3	磁带归并模式	(208)
9.4	磁盘归并技术	(209)
1999 年(下)江苏省高等教育自学考试数据结构(本)试卷及参考答案		(210)
2000 年(下)江苏省高等教育自学考试数据结构(本)试卷及参考答案		(220)

第一章 概 论

1.1 数据类型和数据结构

【本节要点】

一、数据类型(Data Type)

数据类型是一个与数据结构密切相关的概念,最先在高级语言中提出,用来描述运算对象的特性。

1. 数据类型的含义

数据类型是指由一个值集和定义在该值集上的一组操作构成的总体。

在高级语言中,每个常量、变量及表达式等都有一个确定的类型,该类型确定了这些对象可以取值的范围以及可以被施与的操作。例如 PASCAL 语言的整型 integer 的值集为整数集合的一个子集,具体的取值范围与计算机有关,操作集包括加、减、乘、除等等。

2. 原子类型和结构类型

我们将不可再分解的值称为原子项,将可以再分解为更基本值的值称为组合项。例如学生数据包括学号、年龄和各门考试成绩,其中学号和年龄是原子项,各门考试成绩则是组合项。

高级语言的数据类型分为原子类型和结构类型。原子类型指值是不可分解(原子项)的数据类型,例如 PASCAL 语言的整型、实型、字符型、布尔型和枚举类型等;结构类型指值是由若干成分(原子项或组合项)按照特定结构组成的数据类型。例如 PASCAL 语言的数组、记录等都是结构类型。

二、抽象数据类型

1. 数据抽象

数据抽象是程序设计的核心之一,通过抽象使得程序员将精力集中在数据及其操作的理想模型上,而不必关心数据存储及操作的实现,从而减轻程序员的负担,提高效率。数据类型就是数据抽象的手段与结果。例如程序员在高级语言 PASCAL 中定义三个整型变量 x 、 y 、 z ,为 x 、 y 赋值,并将 x 与 y 相加,结果赋给变量 z ,程序片段为:

```
...  
var x,y,z:integer;  
...  
x:=2;  
y:=5;  
z:=x+y;  
...
```

高级语言程序经过编译、链接转变为目标程序,目标程序执行时,完成对 x 、 y 的赋值, x 与 y 相加及结果赋予 z 。下层的这些操作对高级语言程序员来说是透明的。两台计算机可能有完

全不同的指令系统,可能采用截然不同的方法实现这些操作,但高级语言程序却不必修改源代码,而且程序员也可以不必知道它们的差别。数据抽象减少了程序设计的复杂性,增加了程序的可移植性。当然,要注意以下情况,某些变量的某些特定值可能在一些计算机上会溢出,此时程序在这些计算机上不能正常执行。例如在早期的 16 位计算机上,整型变量取值不能为 2^{20} ,所以若程序中有语句 $x := 2^{20}$,程序执行将出错。这种程序可移植性问题不是由数据抽象引起的,而是由数据类型的具体实现的限制而引起的。

2. 抽象数据类型(ADT, Abstract Data Type)

(1) 抽象数据类型的基本思想。

抽象数据类型是一种描述用户与数据之间接口的抽象模型,它指一个包含数据值集合以及该值集合上的一组操作的软件模块。

抽象数据类型的基本思想是将数据类型的使用与其表示和实现分开,通过将数据类型的表示及操作封装在一个程序模块中而对用户隐蔽数据表示和操作过程的细节,用户只要直接使用数据和操作即可。

在软件中使用抽象数据类型可提高软件的重用度。因为在抽象数据类型内部定义数据表示和操作细节,而在模块外部使用的只是抽象的数据和抽象的操作。

(2) 抽象数据类型组成。

抽象数据类型需给出数据及可对数据进行的操作的描述,它由说明(specification)和实现(implementation)两个部分构成。

抽象数据类型的说明部分定义了一个应用程序怎样使用该抽象数据类型,包括语法和语义,语法指明操作名称、参数特性及返回值类型;语义指明数据类型的每种操作的定义、对每个可能的输入将返回的值。

抽象数据类型的实现部分包括表示和算法。表示指明抽象数据类型的值的表示,我们在高级语言层次上讨论抽象数据类型的表示,故利用高级语言提供的数据类型来建立新的结构,用高级语言已有的操作建立新的操作。算法描述抽象数据类型操作的实现。

例 1.1,本例使用 C++ 语言定义一个抽象数据类型 time,它具有操作 init、list。

```
#include <iostream.h>
class time{
    private:
        int year,month,date,hour,minute,second;
        /* 定义数据对象 */
    public:
        void init(int,int,int,int,int,int);
        /* 操作结果:将 6 个参数值分别赋予数据对象 year,month,date,hour,minute,
        second */
        void list();
        /* 初始条件:数据对象 year,month,date,hour,minute,second 已存在。
        操作结果:显示数据对象 year,month,date,hour,minute,second 的值。 */
};
//以上是定义模块
```

```

void time::init(int y,int mon,int d,int h,int min,int s)
{
    year=y;
    month=mon;
    date=d;
    hour=h;
    minute=min;
    second=s;
}
void time::list()
{ cout<<"/"<<year<<"/"<<month<<"/"<<date<<"/"<<hour<<":"<<minute<<":"<<second
<<"\n";
}
//以上是实现模块

```

本例中定义模块中列出了可供外部模块使用的函数名及每个函数的语法和语义。由于C++语言的语法规则,抽象数据类型的值的表示也在这部分定义,即私有数据成员(private)定义,它对应用程序是隐蔽的。操作实现的细节在实现模块中定义,它对应用程序也是隐蔽的。以下是使用上面两个模块的例子。

```

void main()
{
    time t;
    t.init(2000,8,18,20,50,18);
    t.list();
}

```

【习题及分析】

一、填空题

1. 数据类型是指由_____和_____构成的总体。

答案:一个值集,定义在该值集上的一组操作

2. 数据类型分为_____和_____。

答案:原子类型,结构类型

3. 抽象数据类型指_____。

答案:一个包含数据值集合以及该值集合上的一组操作的软件模块

二、简答题

1. 简述抽象数据类型的基本思想。

答案:抽象数据类型的基本思想是将数据类型的使用与其表示和实现分开,通过将数据类型的表示及操作封装在一个程序模块中而对用户隐蔽数据表示和操作过程的细节,用户只要直接使用数据和操作即可。

2. 简述抽象数据类型的组成。

答案:抽象数据类型由说明和实现两个部分构成。抽象数据类型的说明部分定义一个应用程序怎样使用该抽象数据类型,包括语法和语义,语法指明操作名称、参数特性及返回值类型;语义指明数据类型的每种操作的定义、对每个可能的输入将返回的值。抽象数据类型的实现部分包括表示和算法。表示指明抽象数据类型的值的表示,利用高级语言提供的数据类型来建立新的结构,用高级语言已有的操作建立新的操作。算法描述抽象数据类型操作的实现。

三、设计球体的抽象数据结构,数据包括球的半径,操作包括初始化、返回球的半径和体积。

答案:用 C++ 语言定义。

```
class global{
private:
    int radius;
    /* 定义数据对象 */
public:
    void init(int r);
    /* 操作结果:将参数值赋给数据对象 radius */
    int getR();
    /* 初始条件:数据对象 radius 已存在。
       操作结果:返回球体半径值。 */
    int getD();
    /* 初始条件:数据对象 radius 已存在。
       操作结果:返回球体体积值。 */
};
void global::init(int r)
{
    radius=r;
}
int global::getR()
{
    return radius;
}
int global::getD()
{
    return 4/3 * 3.14 * radius * radius * radius;
}
```

1.2 基本数据结构

【本节要点】

一、基本概念和术语

数据是指用符号表示的客观事物。计算机数据指能够被计算机接受和处理的符号集合,在

本书中将其简称为数据。

数据元素是数据的基本单位,数据元素由数据项组成。例如表 1-1 列出了学生数据,学生数据的数据元素是学生记录,学生记录又由学号、姓名、年龄等数据项组成。

表 1-1 学生信息表

学号	姓名	年龄	性别	系 别	专 业
0001	王刚	19	男	计算机	计算机及应用
0002	王洁	18	女	计算机	计算机及应用
0003	刘丽	20	女	计算机	计算机及应用
0004	李立	20	男	计算机	计算机及应用
	...	...	...	...	...

上表中,第一行是表目行,给出了该表的结构,其后的每行为一条数据记录,即数据元素,表示一个学生的信息。每条记录都由 6 个数据项构成。不同记录的同一数据项的值可能不同,例如第一条和第二条记录的“姓名”数据项的值分别为“王刚”和“王洁”。

二、数据结构

关于数据结构概念至今还没有一个公认的标准定义。一般认为,数据结构指数据之间的相互联系。数据结构概念包括 3 个方面:数据间的逻辑关系、数据在计算机中的存储方式和数据的运算,分别称为数据的逻辑结构、物理结构和数据的运算。

1. 数据的逻辑结构

数据的逻辑结构描述数据元素之间的逻辑关系,其形式化定义为:数据的逻辑结构是一个二元组 $B=(K,R)$,其中 K 是数据元素的有限集合, R 是 K 上二元关系的有限集合,即 $K=\{k_i|1\leq i\leq n,n\geq 0\}$, $R=\{r_j|1\leq j\leq m,m\geq 1\}$, n 是数据元素的个数, m 是 K 上关系的个数,本书只讨论 $m=1$ 的情况。 K 上的二元关系 r 是序偶的集合, r 中的序偶记为 $\langle a,b\rangle(a,b\in K)$, a 、 b 分别称为序偶的第一、第二元素,称第一元素为第二元素的直接前驱(简称前驱),称第二元素为第一元素的直接后继(简称后继)。

例 1.2,对表 1-1 中给出的学生信息四条记录,因每个学生的学号必不同,故该数据项可以唯一标识一个记录,称这样的数据项为关键字。用学号来代表整个记录,构造对应于表 1-1 数据的一种数据结构为: $S1=(K,R)$,其中:

$$K=\{0001,0002,0003,0004\}$$

$$R=\{r\}$$

$$r=\{\langle 0001,0002\rangle,\langle 0002,0003\rangle,\langle 0003,0004\rangle\}$$

逻辑结构还可用图形直观地表示,每个数据元素用图中的结点表示,关系中的每个序偶用连接表示两个数据元素的结点的有向边表示。如对应于例 1.1 构造的数据结构的图形表示如图 1-1 所示。



图 1-1 逻辑结构的图形表示

从表 1-1 所表示的数据的含义可以看出, r 是数据元素按学号升序排列的关系。

根据数据元素间逻辑关系的特点,数据的逻辑结构分为四类:(1)集合:结构中的数据元素属于同一个集合。(2)线性结构:结构中的数据元素间是一一对一关系。例 1.2 给出的即为线性结构。(3)树形结构:结构中的数据元素间是一对多关系。(4)图结构:结构中的数据元素间是多对多关系。集合、树型结构和图结构合称为非线性结构,故又可认为逻辑结构分为线性结构和非线性结构两大类。

2. 数据的物理结构

数据的物理结构又称存储结构,指数据的逻辑结构在计算机中的实现,包括数据元素和关系的实现。

最常用的存储结构有顺序结构和链接结构。

(1)顺序结构。顺序结构使用计算机存储器中连续的存储空间实现数据结构,以存储单元间的位置相邻关系表示数据元素间的逻辑关系。这种存储结构主要用于线性结构,也可以用于某些具有特殊性质的非线性结构。

例 1.3,对例 1.2 中定义的线性结构采用顺序存储结构。假设每个数据元素占用 6 个存储单元,第一个数据元素存放在地址为 1000 的单元,那么该线性结构的存储实现如图 1-2 所示。

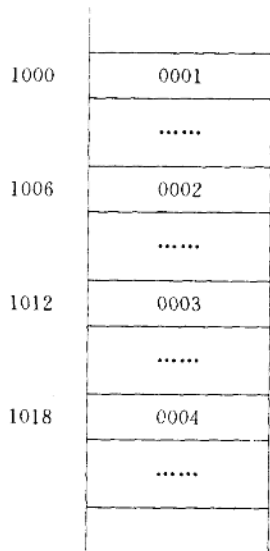


图 1-2 顺序存储结构

(2)链接结构。链接结构利用指针表示数据元素之间的联系,即链接结构中结点所占的存储单元包括两部分:存储数据元素值(结点值)的部分和存储该结点的后继结点地址的部分(指针域)。注意由于一个结点可能有多个后继结点,所以结点的指针域可以包含多个指针。结点结构如图 1-3 所示。

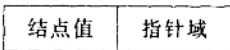


图 1-3 链接存储的结点结构

链接结构既可用于存储线性结构,又可用于存储非线性结构。

例 1.4,对例 1.2 中定义的线性结构采用链接存储结构,如图 1-4 所示。

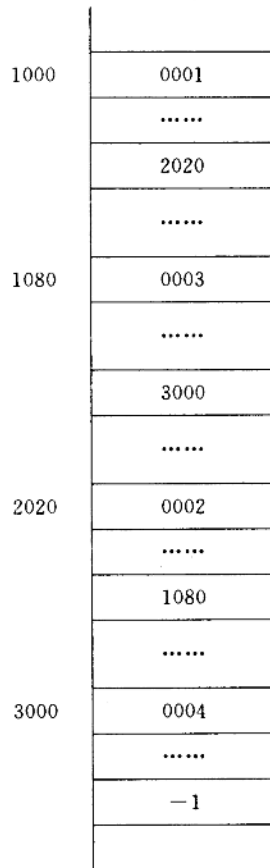


图 1-4 用链接结构存储的线性表

注:指针域为-1 表示该结点没有后继

(3) 顺序结构与链接结构比较。顺序结构需要占用连续的存储空间,而链接结构不必;顺序结构通常需事先分配存储空间,属静态存储分配,链接结构可在需要使用时再分配存储空间,属动态存储分配;顺序结构无须使用额外的存储空间来表示数据元素间的联系,故存储效率高,链接结构需使用一定的额外存储空间来表示数据元素间的联系(称之为结构性开销),存储效率低于顺序结构。

除顺序结构和链接结构外,数据的存储还可使用散列(hash)方法、索引方法等。

3. 数据的运算

数据的运算指对数据进行的操作。数据的运算定义于逻辑结构,实现于物理结构。数据的运算随逻辑结构的不同而有较大的不同。大多数数据结构都有的运算包括查找、插入、删除等。数据运算的描述即算法。

三、三种基本数据结构

1. 数组

数组是包含有限个具有相同数据类型的元素的序列。PASCAL 语言中,数组类型的定义为:

array $[T_1]$ of T_2 ;

其中 T_1 表示下标类型,可以是除整型和实型以外的其它类型(即可以是字符型、布尔型、枚举型及子域型), T_2 表示元素的类型。假设类型 T_1 包含有 N 个不同值,那么该数组类型的每个数组变量均包含 N 个元素。

数组元素是顺序排列的,即第 $i(1 \leq i \leq N)$ 个元素排在第 $i-1$ 元素之后、第 $i+1$ 个元素之前,显然这是一种线性关系,因此其逻辑结构是线性结构。每个元素具有一个索引,指明其在数组中的位置,索引又称“下标”,通过下标可以直接存取数组元素。

数组在存储器中占用连续空间,每个元素占用相等的存储单元数,是顺序存储结构。假设每个数组元素占用 l 个存储单元,数组的首地址为 $LOC(a_1)$,那么数组第 $i(1 \leq i \leq N)$ 个元素的地址可由下式计算:

$$LOC(a_i) = LOC(a_1) + (i-1) * l$$

例如 var arr:array $[1..100]$ of integer; 说明了一个名为 arr 的数组变量,它的元素类型是整型,有 100 个元素,下标值范围为 $1-100$ 。数组元素的引用方法为:数组名[下标],如引用数组 arr 的第 10 个元素的记法是 arr[10]。

PASCAL 语言中数组类型的定义中允许成分类型为数组类型,因此可以通过数组类型的嵌套定义来定义二维及更高维数的数组。例如:

array $[1..5]$ of array $[1..4]$ of integer;

定义了一个 5×4 的二维数组类型,其逻辑结构仍是线性结构;该线性结构共有 5 个元素,每个元素是一个包含 4 个元素数组。二维数组的存储结构采用顺序结构,如图 1-5 所示。

假设 $m \times n$ 二维数组的首地址为 $LOC(a_{11})$,每个元素占用 l 个存储单元,那么元素 $a_{ij}(1 \leq i \leq m, 1 \leq j \leq n)$ 的地址 $LOC(a_{ij})$ 可由下式计算:

$$LOC(a_{ij}) = LOC(a_{11}) + (i-1) * n * l + (j-1) * l$$

需说明的是以上所讨论的二维数组的存储结构是“行优先”存储,即先存储数组的第一行,接下去再存储数组的第二行,依次类推。对二维数组较常用的存储方法还有“列优先”存储,即先存储数组的第一列,接下去再存储数组的第二列,依次类推。

2. 记录

记录是包含有限个元素的序列,元素的数据类型可以不同。PASCAL 语言中使用关键字 record 来说明记录类型或变量,格式为:

record

 <域名 1>: <类型 1>;

 <域名 2>: <类型 2>;

 <域名 n>: <类型 n>

end;

类型 1~类型 n 可以是任一种简单类型或结构类型。

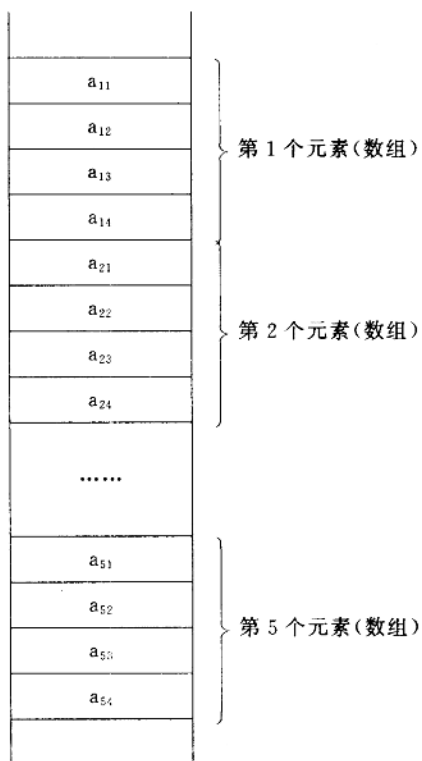


图 1-5 二维数组的存储结构

注：用 a_{ij} 表示二维数组元素， i 为第一维下标值， j 为第二维下标值

例如描述学生情况的记录，包括学号、姓名、年龄、性别、系别和专业 6 个数据项，其 PASCAL 语言说明为：

{先定义学生记录类型}

type

student=record

num;integer;

name;array [1..10] of char;

age;integer;

sex:Boolean;

department;array [1..40] of char;

speciality;array [1..40] of char

end;

{再说明 student 类型的变量}

var s1;student;

{若需处理一组学生信息，则需说明数组变量}

s;array [1..100] of student;

记录的成分(称为字段)的引用方法为：记录变量名. 字段名。例如 s1.name, s[2].age 等。

记录的字段在逻辑上是顺序排列的,其逻辑结构是线性结构;其物理存储采用顺序存放的方式,即物理结构是顺序结构。

3. 指针

指针是动态数据结构的基础。指针类型是以存储单元地址为值的数据类型,一个指针变量的值是地址值或空值。PASCAL 语言中指针类型的定义用保留字符 $\uparrow$,格式为:

$\uparrow$ (类型标识符);

例如说明一个 student 类型的变量指针:var p: $\uparrow$ student;此时 p 的值是地址值,p $\uparrow$ 则是 student 类型的变量,可以引用记录的字段,例如 p $\uparrow$.name,表示引用 p 所指的 student 类型记录变量的 name 字段值。

【习题及分析】

一、填空题

1. 数据(计算机数据)是指_____。

答案:能够被计算机接受和处理的符号集合

2. 数据元素是指_____,它由_____组成。

答案:数据的基本单位,数据项

3. 数据结构概念包括三个方面,即_____,_____,_____,它们分别称为_____,_____,_____。

答案:数据间的逻辑关系,数据在计算机中的存储方式,数据的运算,逻辑结构,物理结构,数据运算

4. 数据的逻辑结构分为四类,即_____,_____,_____,_____。

答案:集合,线性结构,树型结构,图结构

5. 数据的物理结构主要有_____,_____。

答案:顺序结构,链接结构

二、简答题

1. 简述数据结构三方面的含义。

答案:数据的逻辑结构描述数据元素之间的逻辑关系;数据的物理结构指数据的逻辑结构在计算机中的实现,包括数据元素和关系的实现;数据的运算指对数据进行的操作。

2. 简要说明抽象数据类型与数据结构的关系。

答案:从本章第一节和第二节对二者的讨论可知,抽象数据类型与我们所定义的数据结构所描述的内容都包括了对数据逻辑特性、实现和数据操作,所以二者在概念上基本是等同的。也有一些文献对数据结构给出了与本书不同的定义,定义数据结构是抽象数据类型的实现,而抽象数据类型仅指数据逻辑特性描述和运算定义,这样二者在概念上就是不同的,在这样的定义下,数据结构是 ADT 的物理实现。

3. 试比较顺序结构和链接结构。

答案:顺序结构需要占用连续的存储空间,而链接结构不必;顺序结构通常需事先分配存储空间,属静态存储分配,链接结构可在需要使用时再分配存储空间,属动态存储分配;顺序结构无须使用额外的存储空间来表示数据元素间的联系,故存储效率高;链接结构需使用一定的额外存储空间来表示数据元素间的联系(称之为结构性开销),存储效率低于顺序结构。

4. 试比较数组和记录两种基本数据结构。

答案:数组和记录逻辑上都是线性结构,物理上都是顺序结构。区别为:

- (1) 数组元素必须是同一类型的,而记录允许成分是不同的类型;
- (2) 每个数组元素占用的存储单元数相同,每个记录字段占用的存储单元数可以不同;
- (3) 元素访问方式不同,数组元素通过下标值直接访问,而访问记录的字段必须指明字段名。

5. 设 $m \times n$ 二维数组采用“列优先存储”,数组首地址为 $LOC(a_{11})$,每个元素占用 l 个存储单元,试写出计算元素 a_{ij} ($1 \leq i \leq m, 1 \leq j \leq n$) 的存储地址 $LOC(a_{ij})$ 的公式。

答案: $LOC(a_{ij}) = LOC(a_{11}) + (j-1) * m * l + (i-1) * l$

分析:参见第三题图。

三、试画出 5×4 二维数组“列优先”存储的示意图。

答案:如图 1-6 所示。

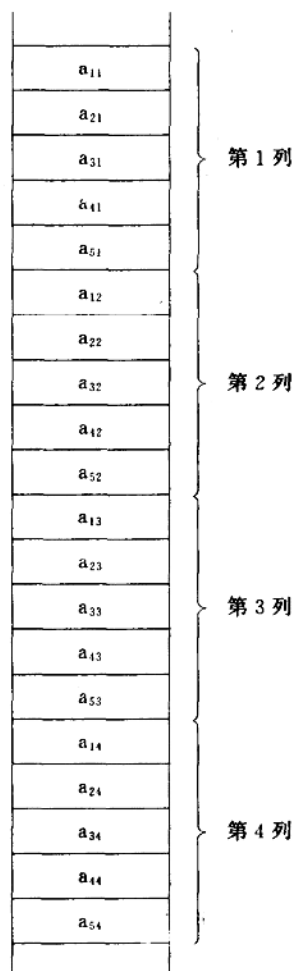


图 1-6 二维数组“列优先”存储示意图