Broadview
www.broadview.com.cn

希赛IT在线教育
www.educity.cn

全国硕士研究生入学统考专用辅导丛书

# 全国硕士研究生入学统一考试

## 计算机学科专业基础综合考点分析与全真模拟：

## 数据结构（分册）

希赛教育研究生院 陈暄 桂阳 主编

新大纲
权威辅导

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

希赛 IT在线教育
www.educity.cn

全国硕士研究生入学统考专用辅导丛书

# 全国硕士研究生入学统一考试

# 计算机学科专业基础综合考点分析与全真模拟：数据结构（分册）

希赛教育研究生院 陈暄 桂阳 主编

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

## 内 容 简 介

本书由希赛教育研究生院组织编写，作为全国硕士研究生入学统一考试计算机学科专业基础综合考试辅导指定教材。本书特点：紧密围绕最新的考试大纲，着重对考试大纲规定的内容有重点地细化和深化，内容涵盖了考试大纲的所有知识点；采取考点分析与真题详解的形式，使读者的学习更具针对性；把可能要考的知识点按实际考试的真题方式组织成模拟试卷，精辟地指出题型的特点，阐述解题思路，使读者更好地了解考试题型，以及试题的解答方法和技巧。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

全国硕士研究生入学统一考试计算机学科专业基础综合考点分析与全真模拟. 数据结构分册 / 陈暄，桂阳主编. —北京：电子工业出版社，2010.10
（全国硕士研究生入学统考专用辅导丛书）
ISBN 978-7-121-11895-1

Ⅰ. ①全… Ⅱ. ①陈… ②桂… Ⅲ. ①数据结构－研究生－入学考试－自学参考资料 Ⅳ. ①TP3

中国版本图书馆 CIP 数据核字（2010）第 187901 号

责任编辑：李利健
印　　刷：
装　　订：北京中新伟业印刷有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编 100036
开　　本：860×1092　1/16　印张：17.75　字数：454 千字　插页：1
印　　次：2010 年 10 月第 1 次印刷
印　　数：4000 册　　定价：36.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。
质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。
服务热线：（010）88258888。

# 前　言

根据教育部文件要求，全国硕士研究生入学统一考试计算机学科专业基础综合全国联考，初试科目调整为4门，分别是政治理论（100分）、外语（100分）、数学一（150分）、计算机专业基础综合（150分）。其中计算机专业基础综合考试内容涵盖数据结构、计算机组成原理、操作系统和计算机网络等学科专业基础课，要求考生比较系统地掌握上述专业基础课的概念、基本原理和方法，能够运用所学的基本原理和基本方法分析、判断和解决有关理论问题和实际问题。

## 内容超值，针对性强

在全国硕士研究生入学统一考试计算机学科专业基础综合考试大纲中，所规定要考查的4个学科知识范围比较广。根据希赛教育研究生院（www.csaiky.com）的调查，考生希望得到一本“精装”书，以便在短时间内对考试大纲所规定的知识点进行快速地回顾和掌握，轻松考出高分。该书既能涵盖考试大纲的所有知识点，同时又很精炼；既能对考试大纲规定的知识点进行解析，又能提供实战练习。

为了帮助考生熟练地掌握考试大纲所规定的知识点，使考生能举一反三，希赛教育研究生院组织有关专家，在电子工业出版社的大力支持下，编写和出版了本书，作为全国硕士研究生入学统一考试计算机学科专业基础综合考试辅导指定教材。本书紧密围绕最新的考试大纲，着重对考试大纲规定的内容有重点地细化和深化，内容涵盖了考试大纲的所有知识点。采取考点分析与真题详解的形式，使读者的学习更具针对性。把可能要考的知识点按实际考试的真题方式组织成模拟试卷，精辟地指出题型的特点，阐述解题思路，使读者更好地了解考试题型，以及试题的解答方法和技巧。根据希赛教育研究生院的计算机专业考研培训学员反馈的经验，通过习题形式来学习知识，能更加容易地掌握知识。同时，通过阅读本书，考生还可以清晰地把握命题思路，掌握知识点在试题中的变化，以便在研究生入学统一考试中洞察先机。

## 作者权威，阵容强大

希赛教育（www.educity.cn）专业从事人才培养、教育产品开发、教育图书出版，在职业教育和基础教育方面具有极高的权威性。特别是在在线教育方面，稳居国内首位，希赛教育的远程教育模式得到了国家教育部门的认可和推广。

希赛教育研究生院是全国计算机专业考研专业课辅导的权威机构，多年对计算机考研专业课考试的跟踪与分析。根据考试大纲，组织权威专家编写和出版了考试教材、考试串讲、习题解答、考前冲刺与

全真模拟等4个系列的辅导书籍，录制了考试培训视频教程、历年考试真题解析视频教程等2个系列的考研视频。在辅导方面，希赛教育研究生院实行个性化辅导，家教式服务，名师亲自制定辅导计划和批改作业，博士团队在线辅导。希赛教育研究生院具有自成体系的辅导资料，使学习更具系统性，复习更具针对性。实时的网络课堂和答疑系统，学员能与名师在线交流。希赛教育研究生院组织相关专家编写了高质量的模拟试题，能有的放矢地帮助学员备考。

本书由希赛教育研究生院陈暄和桂阳担任主编，参加编写的人员还有张友生、胡钊源、王勇、施游、刘毅、朱小平、周玲、李雄、王冀等。希赛教育首席专家张友生博士审核了所有的稿件。

## 在线测试，心中有数

上学吧（www.shangxueba.com）在线测试平台为考生准备了在线测试，其中有数十套全真模拟试题和考前密卷，考生可选择任何一套进行测试。测试完毕，系统自动判卷，立即给出分数。

对于考生做错的地方，系统会自动记忆，待考生第二次参加测试时，可选择“试题复习”。这样，系统就会自动把考生原来做错的试题显示出来，供考生重新测试，以加强记忆。

如此，读者可利用上学吧在线测试平台的在线测试系统检查自己的实际水平，加强考前训练，做到心中有数，考试不慌。

## 诸多帮助，诚挚致谢

在本书出版之际，要特别感谢国家教育部的命题专家们，编者在本书中引用了部分考试原题，使本书能够尽量方便读者的阅读。在本书的编写过程中，参考了许多相关的文献和书籍，编者在此对这些参考文献的作者表示感谢。

感谢电子工业出版社田小康编辑，他在本书的策划、选题的申报、写作大纲的确定，以及编辑、出版等方面，付出了辛勤的劳动和智慧，给予了我们很多的支持和帮助。

感谢参加希赛教育研究生院辅导和培训的学员，正是他们的想法，汇成了本书的原动力，他们的意见使本书更加贴近读者。

由于编者水平有限，且本书涉及的内容很广，书中难免存在错漏和不妥之处，编者诚恳地期望各位专家和读者不吝指正和帮助，对此，我们将十分感激。

## 互动讨论，专家答疑

希赛教育研究生院（www.csaiky.com）是中国最大的计算机专业考研在线教育网站，该网站论坛是国内人气最旺的计算机专业考研社区，在这里，读者可以和数十万考生进行在线交流，讨论有关学习和考试的问题。希赛教育研究生院拥有强大的师资队伍，为读者提供全程的答疑服务，在线回答读



试读结束，需要全本PDF请购买 www.ertongbook.com

者的提问。

有关本书的意见反馈和咨询，读者可在希赛教育研究生院论坛（bbs.csaiky.com）“考研教材”版块中的“希赛教育研究生院”栏目上与作者进行交流。也可在“书评在线”版块中的“希赛教育研究生院”栏目与我们交流，我们会及时地在线解答读者的疑问。

希赛教育研究生院

# 目　录

# 第1章 线性表

线性表是数据结构的核心考点之一，也是数据结构中最基础、最常用的内容。其他的数据结构最终都可以用线性表来描述，理解了线性表对学习其他数据结构有事半功倍的功效。例如，栈和队列可以理解为输入、输出端受限制的线性表，数组可以理解为顺序存储的线性表，而树、森林和图可以理解为若干条线性表的集合。

根据考试大纲，本章要求考生掌握以下知识点：

① 线性表的定义和基本操作。

② 线性表的实现，包括顺序存储结构、链式存储结构、线性表的应用。

## 1.1 线性表的定义和基本操作

线性表就是一串元素的有序排列，它是最基础的数据结构，应用非常广泛，能够进行初始化、查找、插入、删除、更新和遍历等多种操作。

### 1.1.1 线性表的逻辑定义与特征

线性表是一种典型的线性结构，由 $n(n\geq 0)$个有序的数据元素（结点）组成。例如，从 2001 年到 2010 年的年份按顺序排列就是一个线性表（简称“年份线性表”）：

（2001，2002，2003，2004，2005，2006，2007，2008，2009，2010）

**1．线性表的定义**

线性表可定义为 $n$ 个数据元素组成的序列：

$$(a_1,\ a_2,\ a_3,\ \ldots,\ a_n)$$

① 数据元素的个数 $n$ 为线性表长。如果 $n=0$，即线性表中无数据元素，则称此线性表为空表。

② 每个数据元素在线性表中都有一个位置，线性表可以看成是由“位置＋数据元素”组成的集合，这个位置决定了各数据元素在线性表中的前驱元素和后继元素。定义 $a_i(1\leq i\leq n)$为线性表中第 $i$ 个位置上的元素，或者称为线性表的第 $i$ 个元素。

试读结束，需要全本PDF请购买 www.ertongbook.com

线性表中数据元素的位置随着线性表插入（或删除）元素的操作而发生变化。当插入数据元素后，原线性表中插入位置及其之后的所有元素在新表中的位置都增加 1。同理，当删除数据元素后，原线性表中删除位置之后的所有元素在新表中的位置都减少 1。

③ 若线性表中存在两个相连的数据元素组成的序偶对$<a_{i-1}, a_i>$，则称 $a_{i-1}$ 是 $a_i$ 的直接前驱（元素），$a_i$ 是 $a_{i-1}$ 的直接后继（元素）。

④ 数据元素 $a_i(1 \leqslant i \leqslant n)$只是一个抽象符号，它既可以是单数据项，也可以由多个数据项组成。例如，表 1-1 所示的学籍成绩花名册线性表就是一个多数据项元素线性表的例子，它的每个数据元素由学号、姓名、语文、数学、英语、总分和平均分 7 个数据项组成。

**表 1-1　学籍成绩花名册线性表示例**

| 学　号 | 姓　名 | 语　文 | 数　学 | 英　语 | 总　分 | 平 均 分 |
|---|---|---|---|---|---|---|
| 1 | 张三 | 90 | 100 | 70 | 260 | 86.67 |
| 2 | 李四 | 80 | 80 | 65 | 225 | 75.00 |
| 3 | 王五 | 85 | 60 | 90 | 235 | 78.33 |
| 4 | 姚六 | 50 | 90 | 55 | 195 | 65.00 |
| …… | …… | …… | …… | …… | …… | …… |

**2．线性表的特征**

线性表或者为空，或者具有如下特性：

① 有且仅有一个“头”数据元素。例如，年份线性表的“2001”。

② 有且仅有一个“尾”数据元素。例如，年份线性表的“2010”。

③ 除“头”数据元素外，线性表中的每个元素有且仅有一个直接前驱。例如，年份线性表中的“2009”的直接前驱是“2008”。

④ 除“尾”数据元素外，线性表中的每个元素有且仅有一个直接后继。例如，年份线性表中的“2007”的直接后继是“2008”。

**3．特殊的线性表**

特殊线性表有空表和长度为 1 的表。

① 空线性表。没有数据元素的线性表是空线性表，此时，线性表既没有头数据元素，也没有尾数据元素。

② 长度为 1 的线性表。线性表中只有一个数据元素，此元素既是该线性表的头数据元素，也是其尾数据元素，它没有前驱，也没有后继。

### 1.1.2　线性表的基本操作

线性表具有很多操作，按照其对数据元素的操作可分为以下几种。在下面的讨论中，设 L 为线性表结构变量。

（1）void InitList(&L)

初始条件：线性表 L 不存在。

操作结果：构造一个空的线性表 L。

（2）void ClearList(&L)

初始条件：线性表 L 存在。

操作结果：将线性表 L 重置为空表，同时释放原线性表所占用的存储空间。

（3）void DestroyList(&L)

初始条件：线性表 L 存在。

操作结果：销毁线性表 L，同时释放原线性表占用的内存空间。

（4）Bool ListEmpty(&L)

初始条件：线性表 L 存在。

操作结果：判断线性表是否为空表操作。如果线性表 L 是空表，则返回真，否则返回假。

（5）int ListLength(&L)

初始条件：线性表 L 存在。

操作结果：求线性表长操作。返回线性表 L 中数据元素的个数。

（6）int LocateElem(&L,$x$)

初始条件：线性表 L 存在。

操作结果：查找数据元素操作。在线性表 L 中查找首次出现值为 $x$ 的数据元素，并返回该元素在 L 中的位置，查找成功。如果线性表中没有数据元素 $x$，则返回一个特殊值 0，表示查找失败。

（7）void GetElem(&L,i,&e)

初始条件：线性表 L 存在，且 $1 \leqslant i \leqslant$ ListLength(L)。

操作结果：获取数据元素取值操作。读取线性表 L 中的第 $i$ 个数据元素的取值，并将之存储到 e 中。

（8）void ListInsert(&L,i,x)

初始条件：线性表 L 存在，且 $1 \leqslant i \leqslant n+1$（$n$ 为表长）。

操作结果：在线性表 L 的第 $i$ 个位置上插入一个值为 $x$ 的新元素，这样使原序号为 $i$，$i$+1，…，$n$ 的数据元素的序号变为 $i$+1，$i$+2，…，$n$+1，插入元素成功后，线性表 L 的长度增加 1（为 $n$+1）。

（9）void ListDelete(&L,i,&e)

初始条件：线性表 L 存在，$1 \leqslant i \leqslant n$（$n$ 为表长）。

操作结果：在线性表 L 中删除序号为 $i$ 的数据元素，删除后使序号为 $i$+1，$i$+2，…，$n$ 的元素变为 $i$，$i$+1，…，$n$-1，删除元素成功后，线性表 L 的长度减 1（为 $n$-1），被删除的数据元素存储到 e 中。

（10）void ListLink(&L1,&L2)

初始条件：线性表 L1 和 L2 都存在。

操作结果：将线性表 L2 接到线性表 L1 之后构成一个新线性表，返回线性表 L1 的地址。

（11）void ListMeger(&L1,&L2)

初始条件：两个有序线性表 L1 和 L2 都存在。

操作结果：将两个有序线性表 L1 和 L2 归并成一个新的有序线性表（放到 L1 中），并返回新线性

表的地址。

（12）void visit(&L)

初始条件：线性表 L 已存在。

操作结果：对指定元素做规定的操作。

（13）void ListTraverse(&L)

初始条件：线性表 L 已存在。

操作结果：按次序访问线性表中的每个元素一次且只访问一次。

（14）void ListReverse(&L)

初始条件：线性表 L 已存在。

操作结果：将线性表元素的顺序变成与原来的次序恰好相反，称为就地逆置。

（15）void ListCopy(&L1,&L2)

初始条件：线性表 L1 已存在，L2 不存在。

操作结果：将线性表 L1 复制到线性表 L2 中。

## 1.2 线性表的实现

线性表的存储结构有多种方式，包括顺序存储结构的无序顺序表和有序顺序表，链式存储结构的单链表、双链表和循环链表等。

### 1.2.1 顺序存储结构

采用顺序存储结构存储的线性表简称为顺序表，它采用一组地址连续的存储单元依次存放线性表中的数据元素，如图 1-1 所示。

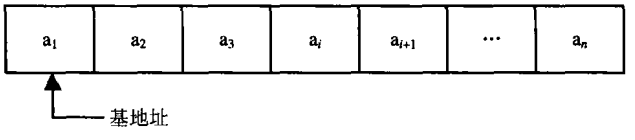


图 1-1 线性表顺序存储示意图

顺序表具有以下 3 个特点：

① 所有的数据元素存储在同一个区域内，而此区域也只存储该线性表的数据元素。

② 数据元素的存储地址与其在线性表中的顺序相同。假设线性表中存在元素 $a_i$ 和 $a_j$（$i<j$），即线性表中 $a_i$ 在 $a_j$ 之前，则 $a_i$ 的存储地址也将在 $a_j$ 之前。线性表的头元素存储在所有数据元素的最前面，尾元素存储在所有数据元素的最后面。

③ 线性表中逻辑上相连的两个数据元素的物理存储地址也相连。假设线性表中存在元素 $a_i$ 和 $a_{i+1}$（$1<i<n$），即 $a_i$ 是 $a_{i+1}$ 的直接前驱，$a_{i+1}$ 是 $a_i$ 的直接后继，那么 $a_i$ 与 $a_{i+1}$ 的物理存储地址相连，即 $a_i$ 后面直接存储 $a_{i+1}$。

在顺序表的连续地址存储空间中存在一个首地址，称为该线性表的基地址，它存储了线性表中的第一个数据元素。

### 1．顺序存储结构的定义

在高级语言中，由于数组具有连续空间和随机访问的特性，因此，一般采用数组来描述顺序存储的线性表，如下定义了一个数据类型为 Elemtype 的顺序存储线性表的例子：

```
#define LISTMAXSIZE 100
typedef struct{
    Elemtype Elem[LISTMAXSIZE];      //存储数据元素
    int nLength;                     //线性表中数据元素的数量
}LIST;
```

在上述定义中，成员变量 Elem 存储了线性表的数据元素信息，其中数组名 Elem 描述了连续地址空间的首地址，成员变量 nLength 描述了线性表的长度。

**【例 1-1】** 已知线性表如下所示，采用结构 LIST 存储，请画出其示意图：

（2000，2001，2002，2003，2004，2005，2006，2007，2008，2009）

【解析】如图 1-2 所示，Elem 是线性表的基地址，nLength 即表明线性表中当前数据元素数为 10，又指向下一个数据元素存储的位置 Elem[10]（C 语言数组从 0 开始记数，Elem[10]表示数组中第 11 个元素）。

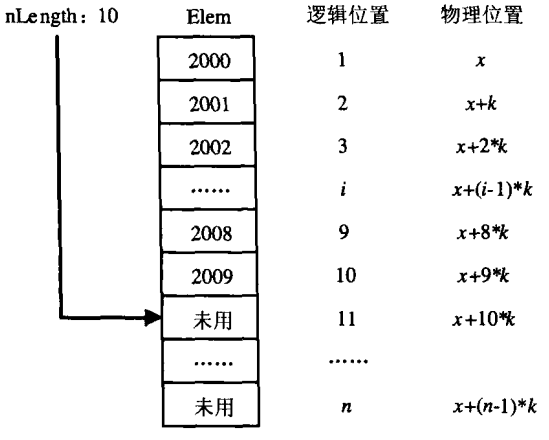


图 1-2　顺序存储线性表示例图

结构 LIST 采用静态数组简单地描述一个线性表，但也有一个很大的缺陷，就是其描述线性表的最大元素的个数为定值（LISTMAXSIZE），不能随意增加，这将造成线性表元素的无限增长与线性表最大空间有限之间的矛盾，从而影响了其应用范围。

### 2．顺序存储结构的元素寻址

线性表中每个元素的物理位置与其逻辑位置存在关联关系，定义 LOC($a_i$)为数据元素 $a_i$ 的物理地址，那么基地址的取值为 LOC($a_1$)，即元素 $a_1$ 的物理地址。设每个数据元素占用 $k$ 个字节空间，那么每个元

素的物理地址如表 1-2 所示。

表 1-2　顺序存储线性表数据元素物理地址

| 元素序号 | 物理地址 | 与基地址偏移 |
|---|---|---|
| 第 1 个元素 | LOC($a_1$) | 0 |
| 第 2 个元素 | LOC($a_1$)+$k$ | $k$ |
| 第 3 个元素 | LOC($a_2$)+$k$=LOC($a_1$)+2×$k$ | 2×$k$ |
| …… | …… | …… |
| 第 $i$ 个元素 | LOC($a_{i-1}$)+$k$=LOC($a_1$)+($i$-1)×$k$ | ($i$-1)×$k$ |
| …… | …… | …… |
| 第 $n$-1 个元素 | LOC($a_{n-2}$)+$k$=LOC($a_1$)+($n$-2)×$k$ | ($n$-2)×$k$ |
| 第 $n$ 个元素 | LOC($a_{n-1}$)+$k$=LOC($a_1$)+($n$-1)×$k$ | ($n$-1)×$k$ |

从表 1-2 中可以看出，线性表中每个元素相对于线性表基地址的物理偏移量与其逻辑位置成正比。假设线性表基地址为 LOC($a_1$)，每个数据元素占用 $k$ 个字节的空间，那么线性表中逻辑第 $i$ 个元素 $a_i$ 的物理地址的推导公式为：

$$\begin{aligned}\text{LOC}(a_i) &= \text{LOC}(a_{i-1})+k = \text{LOC}(a_{i-2})+k+k = \text{LOC}(a_{i-3})+3\times k\\ &= \text{LOC}(a_{i-4})+4\times k = \ldots\\ &= \text{LOC}(a_1)+(i-1)\times k\end{aligned}$$

其中，LOC($a_1$)是基地址，所以，逻辑第 $i$ 个元素 $a_i$ 的物理地址为：

物理地址 = 线性表基地址+(逻辑序号−1)×单个元素空间

【例 1-2】某顺序存储的线性表，最多可存储 100 个数据元素，其基地址为 10000，数据元素为整型，每个数据元素占用 4 个字节空间，求其逻辑第 1、2、$i$ 个和最后一个数据元素的地址。

【解析】可根据元素物理地址计算公式求解。

第 1 个元素的物理地址：LOC($a_1$) = 10000

第 2 个元素的物理地址：LOC($a_2$) = 10000+4

第 $i$ 个元素的物理地址：LOC($a_i$) = 10000+($i$-1)×4

最后一个元素的物理地址：LOC($a_{100}$) = 10000+(100-1)×4

### 3．顺序存储结构的操作算法

由于数组的随机访问特性，采用数组存储的顺序结构线性表的元素定位和读取操作显得特别简单，例如，第 $i$ 个元素 $a_i$ 可表示为“L.Elem[$i$-1]”。

① 初始化、销毁和置空算法。对于采用静态数组（例如，LIST 结构）描述的线性表，只需将其数据元素数量变量（即表长变量）强行置为 0 即可：

```
L.nLength=0;
```

而对于采用动态数组（例如，LISTEXT 结构）描述的线性表，则还需要申请或释放空间和清零等

操作。例如，初始化代码如下：

```
Status InitList(LISTEXT &L){          //线性表初始化算法
   if ((L.Elem = (Elemtype *)malloc(sizeof(Elemtype)*LISTINITSTEP)) return ERROR;
   //申请线性表存储空间
   L.nSize = LISTINITSTEP;            //线性表最大容量为 LISTINITSTEP
   L.nLength = 0;                     //线性表长度为 0
   return OK;
}
```

② 获取长度算法。直接读取长度变量 nLength 即可：

```
return (L.nLength);
```

③ 获取元素取值算法。利用数组特性直接获取，以下代码为读取顺序线性表中第 $i$ 个元素（从 0 开始计数）的取值：

```
Status GetElem(LIST L, int i, ElemType &e){
   if (i<0 || i>=L.nLength) return ERROR;
   e = L.Elem[i-1];
   return OK;
}
```

可以看出，顺序存储线性表中随机查找数据元素操作的时间复杂度是常数，表示为 $O(1)$。

④ 插入元素算法。在顺序存储结构的线性表中插入元素需要移动数据。在第 $i$ 个位置插入，则需要将原线性表中编号为 $n$，$n$-1，…，$i$+1，$i$ 的元素依次先后移动一个位置，变为编号为 $n$+1，$n$，…，$i$+2，$i$+1 的元素。

**【例 1-3】**已知线性表（2001，2002，2004，2005，2006），现需在其第 3 个元素位置处插入数据 2003，请画出插入前后的示意图。

【解析】插入前线性表如图 1-3（a）所示。插入时，首先将原线性表中第 3 个元素之后（含第 3 个元素）的所有数据元素依次向后移动一位，如图 1-3（b）～图 1-3（d）所示，再将新线性表中第 3 个元素赋值为 2003，最后把线性表长 nLength 加 1 即可，如图 1-3（e）所示。插入后的线性表为（2001，2002，2003，2004，2005，2006）。

⑤ 删除元素算法。在顺序存储结构的线性表中删除元素需要移动数据。在第 $i$ 个位置删除，则需将原线性表中编号为 $i$+1，$i$+2，…，$n$ 的元素依次向前移动一个位置，变为编号为 $i$，$i$+1，…，$n$-1 的元素。

**【例 1-4】**已知线性表（2001，2002，2003，2004，2005，2006），现删除其中第 3 个元素，请画出删除前后的示意图。

【解析】删除前线性表如图 1-4（a）所示。删除时，首先将原线性表中第 3 个元素之后（不含第 3 个元素）的所有数据元素按照“从前到后”的顺序依次“向前”移动一位，最后把线性表长（nLength）



试读结束，需要全本PDF请购买 www.ertongbook.com