数据结构技术

王本颜　方蕴昌　编著

清华大学出版社

数据结构技术
王本颜　方蕴昌　编著

数据结构技术

王本颜　方蕴昌　编著

清华大学出版社

内 容 提 要

本书介绍计算机科学中的数据结构理论和方法。共分五章，内容有表、排序、检索、树和复杂数据结构等。其中，介绍的排序方法较多、平衡树的插入和删除、穿线平衡树的插入、B^+树、B*树、前缀B树、二分B树等内容在国内同类书籍中尚不多见。

书中各部分内容均用通俗的语言和算法描述，所有例子和程序框图均用 BASIC 程序语言调试通过。书后附有习题，供读者练习。

本书可作为高等工科院校工程专业及中等学校计算机科学专业的教材，也可供从事计算机科学与软件工作的教师、科技工作者和其他有关工程技术人员学习参考。

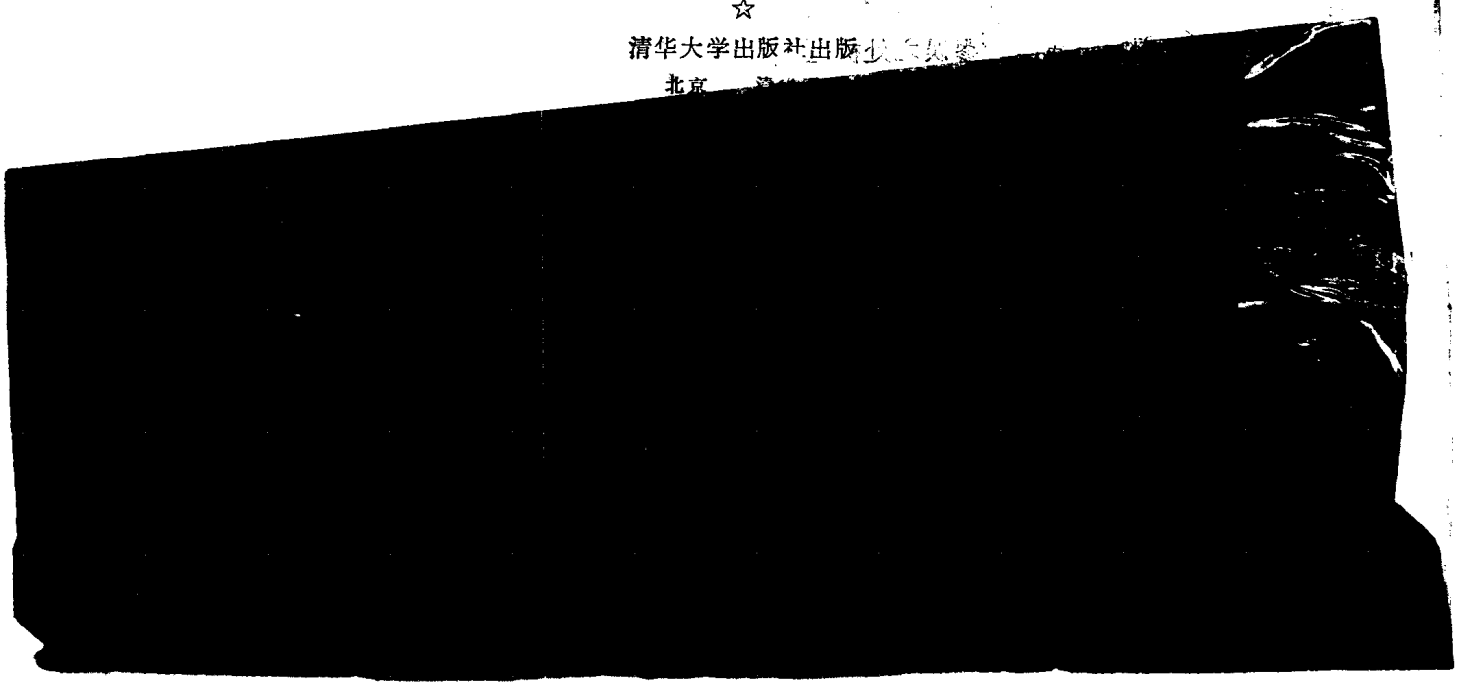
数 据 结 构 技 术

王本颜 方蕴昌 编著

☆

清华大学出版社出版

北京



前　言

当今的世界已经进入了信息时代，随着计算机科学硬件和软件的迅猛发展，计算机的应用已逐步深入到科研和生产的各个领域。除此以外，计算机在企业部门和经济部门的管理工作中也发挥了巨大的作用。如情报检索、企业管理、数据处理、数据库、网络通讯、人工智能等。在这些领域中，数据结构是重要的基础和有效的手段。因此，数据结构不论在系统软件或应用软件中都得到了广泛的应用。尤其对非计算机科学的广大应用软件科技人员，都应很好掌握。目前，国内外出版的同类书籍虽已不少，但对初学者和从事计算机系统软件应用的工程技术人员来说，深望能得到一本“深浅适度，通俗易懂，运用方便”的有关数据结构方面的参考书，以满足教学、科研和应用的需要。为此，我们着手编写了这本书，希望能达到上述目的，帮助读者提高理论和运用的水平。

本书在介绍数据结构基本理论的基础上，尽可能多地把近年来数据结构范畴内的新成就奉献给读者。本书是笔者根据几年来教学上的实践和体会，并参考了国内外有关较新的文献编写而成。为使全书结构避免松散和易于教学，本书既详尽地介绍了各种数据结构的理论；又提出了具体实现的方法。对有关算法的分析，并没有繁复的数学推导和证明，而是依照由浅入深和循序渐进的原则编排各章节，力求使全书衔接自然，系统全面。鉴于我国目前BASIC语言较普及，我们对本书中介绍的数据结构内容，都给出了直观易读的流程框图，并用BASIC程序调试通过。读者可以方便地验证，加深理解，并可供实际工作中参考和移植。因此，本书有一定的实用价值。本书可用作高等工科院校工程专业及中等专科学校计算机专业的教材，也可供从事计算机科学与软件工作的教师、科技工作者和其他有关工程技术人员学习参考。

本书的第一章由翁志强编写，第二章和第三章由王忠萍和方蕴昌编写，第四章由高幸和王本颜编写，第五章由王本颜编写，并由王本颜和方蕴昌统编。最后，由施伯乐教授对全书作了审定。在本书的编写过程中，曾得到了上海市物资局计算站许多同志的大力支持。借此机会，谨向上述同志及为出版本书付出了辛劳的所有同志表示衷心的感谢。

由于我们学识浅薄，[illegible]所望读者不吝赐教。

编　者

1985年10月于上海

目　　录

绪　论

§1　数据结构的发展

早在六十年代初期，计算机专业尚无这门课。仅在《编译原理》、《操作系统》和《程序设计》等课程中涉及到一些数据结构的有关内容。六十年代中期，国外某些大学开始设立了类似的课程，当时多半称为表处理语言（List Processing Language），主要研究几种表处理系统。如 J.Weizenbaum 的简单表处理系统 SLIP，A.Newell、C.Shaw 以及 H. Simon 的信息处理语言 IPL-V 系统，J.Mc Carthy 的表处理语言 LISP 和 D.Farber 等人设计的串处理语言 SNOBOL 系统等。1968 年，计算机科学的巨人 D.E.Knuth 在其著作的“The Art of ComputerProgramming”中，系统而详尽地讨论了各种数据结构。此后，各大学才纷纷开设了数据结构这门课。我国从七十年代末期开始也有了数据结构这一课程。

随着计算机科学的发展，一方面由于硬件的发展，制造出体积小、价格低的各种小型机和微型机，给计算机在各领域普及应用带来了可能和方便。另一方面由于软件和软件应用的发展，使得系统程序和应用程序规模庞大、结构复杂。要想设计出效率高、可靠性好的程序，必然导致对算法思想和数据处理方法的优化进行研究。不但要研究程序的结构和算法，而且要研究程序加工的对象、数据的物理结构和逻辑结构。

除此之外，人类社会的高速发展，使得各种信息与日俱增，有人称现今世界信息量是“信息爆炸”。如何用计算机处理这些大量的信息，这给计算机在非科学计算问题上的应用带来了广阔的前途。今天，计算机已主要应用于诸如情报检索、企业管理、文件系统、数据库等方面。这类问题的共同特点是把大量的数据抽象和归纳成某种物理结构和逻辑结构。进而，用计算机对它们进行“加工处理”。综上所述，不论从事系统软件设计，还是应用软件设计的人员，都必须对数据的物理结构和逻辑结构，即数据结构，进行深入地探讨和研究。

§2　数据结构的定义

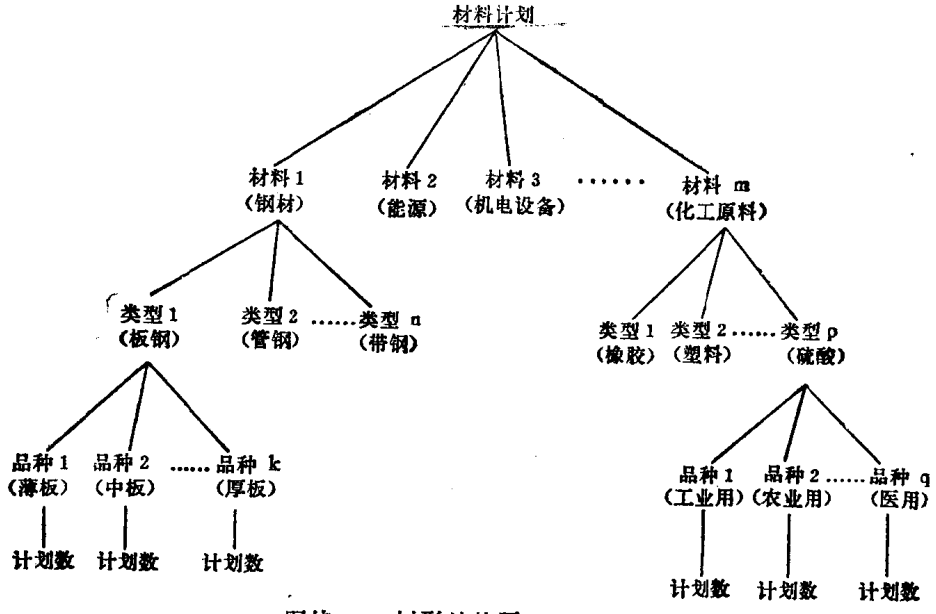
读者也许会问，究竟什么是数据结构呢？为了回答这个问题，让我们先来看这样两个例子。

例 1：体育部门对各国优秀运动员的成绩要作记录，并进行统计和分析。设有一张跳高的记录表，它记载了 m 个世界著名跳高运动员的名字、最佳成绩和身高。这实际上是由一系列的三元式：(a_1, b_1, c_1)，(a_2, b_2, c_2)，…，(a_m, b_m, c_m) 所组成的一个三元表。其中，a_i，b_i，c_i $(i=1, 2, \cdots, m)$ 分别表示某运动员的姓名、最佳成绩和身高。我们可以根据不同的要求来组织这张表的结构，访问表中的数据。若把三元式按顺序逐个记录于表中，那么，要随意地检索某个运动员的最佳成绩和身高，只能用顺序检索的方法，逐个地比较运动员的姓名、成绩和身高。显然，这不是好的办法。在这种数据结构（顺序结构）条件下，无法用更好的检索方法。这就说明这种顺序结构不是好的结构。为了便于检索，我们不

妨把运动员的姓名按字典序排列，重新组织这个记录表（排序结构）。那么，该表中的姓名有了关系，给检索带来了方便，减少了比较的次数。

本例表明，对数据进行不同的组织，形成不同的结构，可以直接影响到处理问题的效率和算法的设计。这是一个数据结构问题。

例 2：在企业管理中，如经济计划部门在制定生产计划时，要分门别类地列出各种原材料的计划。如何很好地组织这些材料（数据），使人能一目了然，这也是一种数据结构问题。通常，人们都以树形结构来组织这些材料（数据）。具体的数据结构图形如图绪 -1 所示。



图绪 -1　树形结构图

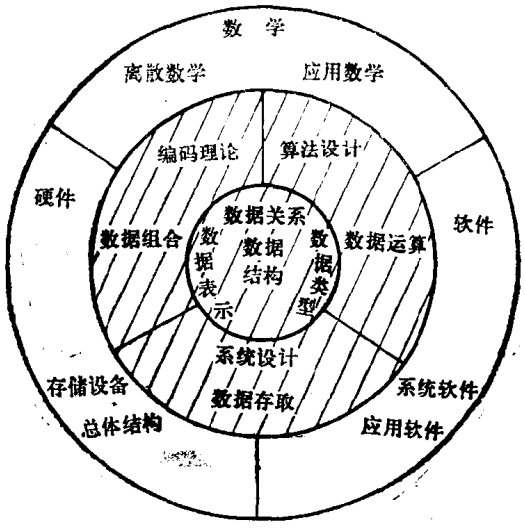
用这种方法组成的材料计划树，层次清晰，类别分明。如对厚钢板的计划数进行检索，首先可根据它的属性，从钢材中找，再根据类别，可在钢板类内找，最后就可按品种检索到厚钢板的计划数。本例表明，这种结构便于检索。

上述二例仅告诉我们什么是数据结构的概念。至于如何去存入、取出这些数据，如何去组织数据的逻辑结构和物理结构，对某个实际问题如何去选择合适的数据结构、设计算法、编制程序等问题，就是我们这门数据结构课所要研究的内容。由此，我们可以看到，数据结构最主要的是研究数据的逻辑关系和数据表示的一门学科。至此，我们可以给出数据结构的确切定义：数据结构 B 是一个二元组 B = (K, R)，其中 K 是结点的有限集合，R 是集合 K 上的关系的有限集合。两者的有机结合，就是数据结构。

§3　数据结构的重要性

计算机是帮助人们进行计算的一种现代化工具，计算机科学是一门研究信息表示和处理的科学。对计算机科学稍有了解的人都知道，计算机必须通过程序（包括系统程序和应用程序）才能进行信息处理。要使计算机有较高的效率，就需要设计出较好的程序。有人曾概括成这样一个公式：程序 = 算法 + 数据结构。当前，数据结构和算法已成为计算机程序设计技术的二大支柱。这就是说，要设计出好的程序，除了要设计出好的算法之外，还必须采用合适的数据结构。所以，数据结构是计算机专业一门重要的课程。

数据结构与数学、计算机硬件和软件之间有着相当密切的关系（如图绪-2所示）。它是计算机科学系的一门重要专业课程，也是程序设计、算法分析、编译原理、操作系统、数据库和人工智能等课程的基础。



图绪-2 数据结构作用图

数据结构研究的对象是各种数据表示、类型和关系的集合，它研究的范围主要包括：研究各种数据结构的性质，不仅要研究数据的逻辑结构和物理结构，而且要研究如何对各种结构的数据定义各种运算，设计各种算法，并分析算法的效率（亦称为算法分析）。同时，还要研究各种数据结构在计算机科学及软件工程中的某些应用，探讨数据排序和检索等方面的技术。在本书中涉及的数据类型包括表、串、栈、队列、数组、树、图和文件等。本书注重理论与实践的结合，致力于在系统地介绍数据结构理论的同时，也给出了各种实践的方法。鉴于我国BASIC语言较普及，我们对本书中所介绍的内容，均用BASIC程序调试通过。为便于读者用各种语言编程和验证，加深理解，我们对各种方法均给出通俗易懂的流程框图，可供读者在工作中参考和移植。

本书内容较丰富，读者可以按工作与学习条件的需要，适当地选择学习内容。书中每章均配备了适量的习题，供读者练习。凡带有*号的章节和习题，都有一定的难度，初学者可以略去不顾。预期读者学完本书之后，将对数据结构能有一个较全面、较系统的认识。并能独立地进行有关领域内的科研与工程实践，使计算机科学更好地为人类社会服务。

第一章　表

在计算机科学与应用中，数据是加工处理的主要对象。要使计算机对数据的处理速度加快、效率提高，就必须研究数据在机内的存放形式以及数据与数据之间的组合结构。基于这个目的，我们将散乱的数据集合转化成结点的集合。根据结点与结点之间的不同组合关系，可以分成线性的与非线性的关系。结点间的线性组合关系又称为表，而非线性关系又称为树、图等。

线性表是应用最广泛的数据结构之一，根据表中结点的不同存放形式，可分为顺序关系与链接关系；若按结点在表中的存取方式来分，又有链、栈、队等不同的方式。为了提高计算机的使用效率和节省存贮单元，我们还必须研究数据结点在机内的合理存贮方法。本章除了介绍表的各种形式及其处理方法和应用技巧以外，还介绍了数据在机内的各种存贮技巧，如压缩存贮、索引存贮、散列存贮以及矩形数组和 m 维数组的各种存贮方法。在本章的最后，还专门介绍了串的概念及其各种处理方法，为读者提供了一系列处理线性表数据的有效方法和理论基础。

§1.1　表的概述

数据是电子计算机处理的对象。无论是科学计算，或是数据处理，过程控制以及文件系统中信息的贮存和检索等等，都是对数据进行一系列加工处理的过程。为了使计算机对每一数据的处理和存贮变得方便，我们将数据转化成结点的形式，即以结点来表示某一数据集合中的每一个数据的普遍特征。结点及结点间结构关系的不同组合而形成了如表、树、图等各类不同的数据结构。表是本章讨论的主要内容。在给出表的一般性概念之前，首先对本书中所研究的各类数据结构的基本元素——结点作一些说明。

结点的存贮形式如图 1-1 所示。

在存贮区开辟一个有 n 个结点的集合 K 的存贮区域 z，每一结点 $k_i \in K$，$1 \leqslant i \leqslant n$。

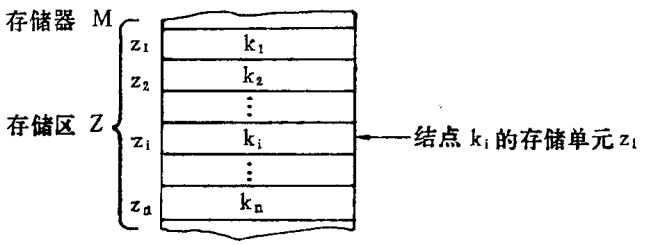


图 1-1　结点的存贮形式

在数据结构 B = (K, R) 中，K 是结点的有限集合，R 是 K 上关系的有限集合，结点的一般表示形式如图 1-2 所示：

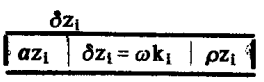


图 1-2　结点的表示形式

其中：

z_i——结点的存贮单元

az_i——存贮单元地址

δz_i——给定地址的存贮单元，也称该结点存贮单元的数据场。数据场 ωk_i 可以有多个分量，如 $\omega_1 k_i$，$\omega_2 k_i$，……

ρz_i——结点存贮单元的指针场。ρz_i 也可有多个分量，如 $\rho_1 z_i$，$\rho_2 z_i$，……

一个结点在存贮器中占有一个或若干个存贮单元。在高级语言中，一般是用一个或几个数组联合起来表示结点。其中每个数组表示结点集合中结点的某一个场；而每个数组的第 i 个分量则表示结点集合中第 i 个结点的某一个场。如表 1-1 所示：

表 1-1 学生登记表

地址 I	姓　名 D$（　）	年　龄 D1（　）	总　分 D2（　）	实现各种关系的指针场 R1（　），R2（　），…，Rn（　）
1	HH	20	480	
2	JJ	21	500	
3	AA	21	410	
⋮	⋮	⋮	⋮	

在表 1-1 中 D$（　），D1（　）和 D2（　）分别表示结点集合中结点的姓名、年龄和总分。当 i=3 时，D$（3）=AA，D1（3）=21，D2（3）=410。它们分别表示第 3 个结点的姓名、年龄、总分三个项中的数值，这样，即用 i=3 所指的各项的值来表示一个结点。该结点在存贮器中的存贮情况如图 1-3 所示。

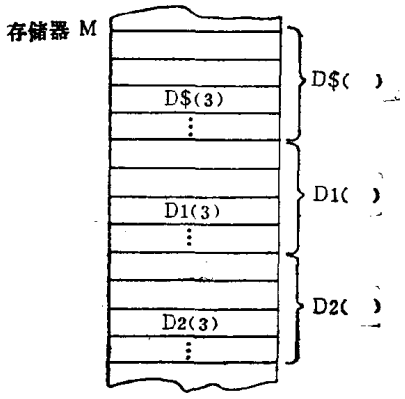


图 1-3　结点的存贮情况

由于数据结构主要研究的是结点与结点之间的关系，而不是着重研究结点本身。为了标识结点引入了关键字。当某一个结点有多个数据项时，可以选定一个或几个数据项来表示该结点的特征，以区别结点集合中的其他结点，这样选定的数据项称为该结点的关键字。在表 1-1 所示的学生登记表中，姓名、年龄和总分均是结点的数据项，其中姓名可选作结点的关键字（若不存在相同的姓名），而年龄、总分则不能选作关键字。

§1.2 线　性　表

线性表定义：

长度为 n 的线性表是一种数据结构 B=(K，R)，其中 K 由 n 个结点组成，R 只有一种关系 N，且它确定了一个结点序列 k_1，k_2，…，k_n，并有：

$N=\{(k_{i-1}, k_i) \mid 2 \leqslant i \leqslant n\}$

则称结点序列 k_1，k_2，…，k_n 为线性表。

表 1-2 所示的学生登记表就是一个线性表。

1.2.1 线性表的顺序存贮

如果数据结构 B=(K，R) 中的关系 N 通过结点的顺序存贮来实现，则称该线性表的存

表 1-2 学 生 登 记 表

地址 I	姓　名 D$（ ）	年　龄 D1（ ）	总　分 D2（ ）	指针场 R（ ）
1	HH	20	480	2
2	JJ	21	500	3
3	AA	21	410	4
4	BB	22	420	5
5	CC	20	430	6
6	DD	21	440	7
7	FF	21	460	8
8	GG	20	470	9
9	EE	20	450	10
10	II	20	490	0

贮是顺序存贮。顺序存贮即在存贮区中开辟若干数组（或串组），把结点一个个顺序存入。在表 1-3 中，以姓名作关键字，根据输入的次序，顺序地把每一结点存入存贮器。由于是顺序存放，可以省去指针场。这样，线性表的实际存贮情况如表 1-3 所示：

表 1-3 学 生 登 记 表

地址 I	姓　名 D$（ ）	年　龄 Di（ ）	总　分 D2（ ）
1	HH	20	480
2	JJ	21	500
3	AA	21	410
4	BB	22	420
5	CC	20	430
6	DD	21	440
7	FF	21	460
8	GG	20	470
9	EE	20	450
10	II	20	490

结点为 k_1，k_2，…，k_n 的线性表顺序存贮时，第 i 个结点的地址可由下式求得：

$$az_i = az_1 + S\ (i-1)$$

其中：

i——第 i 个结点

S——每个结点占用的存贮单元数

az_1——第一个结点的地址

下面以学生登记表为例研究线性表顺序存贮的若干操作：

1．求线性表中的结点总数；

2．按关键字检索线性表中某结点；

3．检索线性表中的第 i 个结点；

4．修改线性表中的第 i 个结点；

5．删除线性表中的第 i 个结点；

6．在第 i 个结点后插入一个新结点；

7．在第 i 个结点前插入一个新结点；

8．将两个或两个以上线性表合并成一个；

9．将一个线性表拆成两个或多个线性表；

10．将线性表中的结点重新排序。

几点说明：

1．在本章中只介绍线性表的 1 至 7 的操作，8 至 10 的操作在后面排序的章节中再作介绍。

2．对线性表的操作中为方便起见，仅对结点的一个数据项即关键字进行操作。应注意，结点的其他数据项实质上存在。

下面给出顺序存贮的线性表的 1 至 7 的操作的程序框图。算法的开始先将表 1-3 存入存贮器。程序框图 1-1 如下：

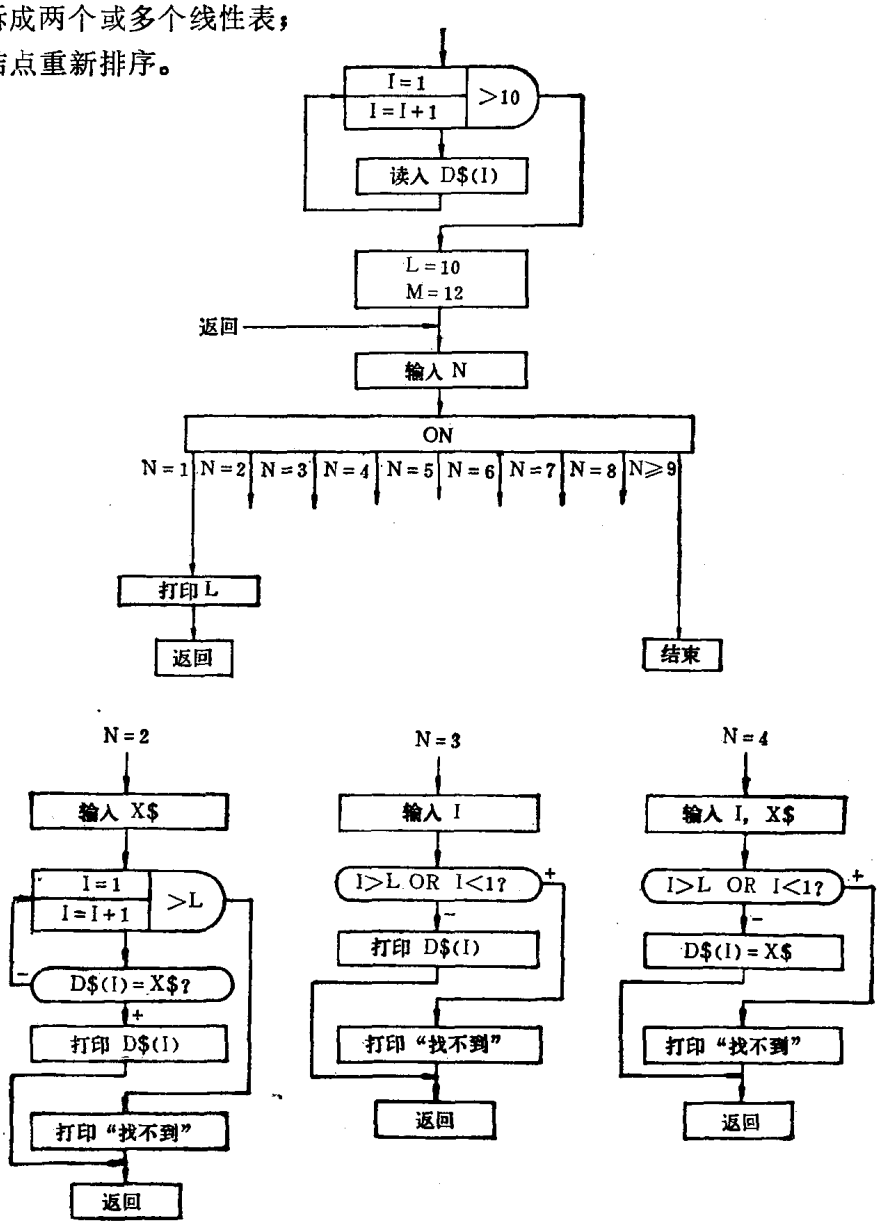
变量说明：

D$(　)——存贮的结点的关键字

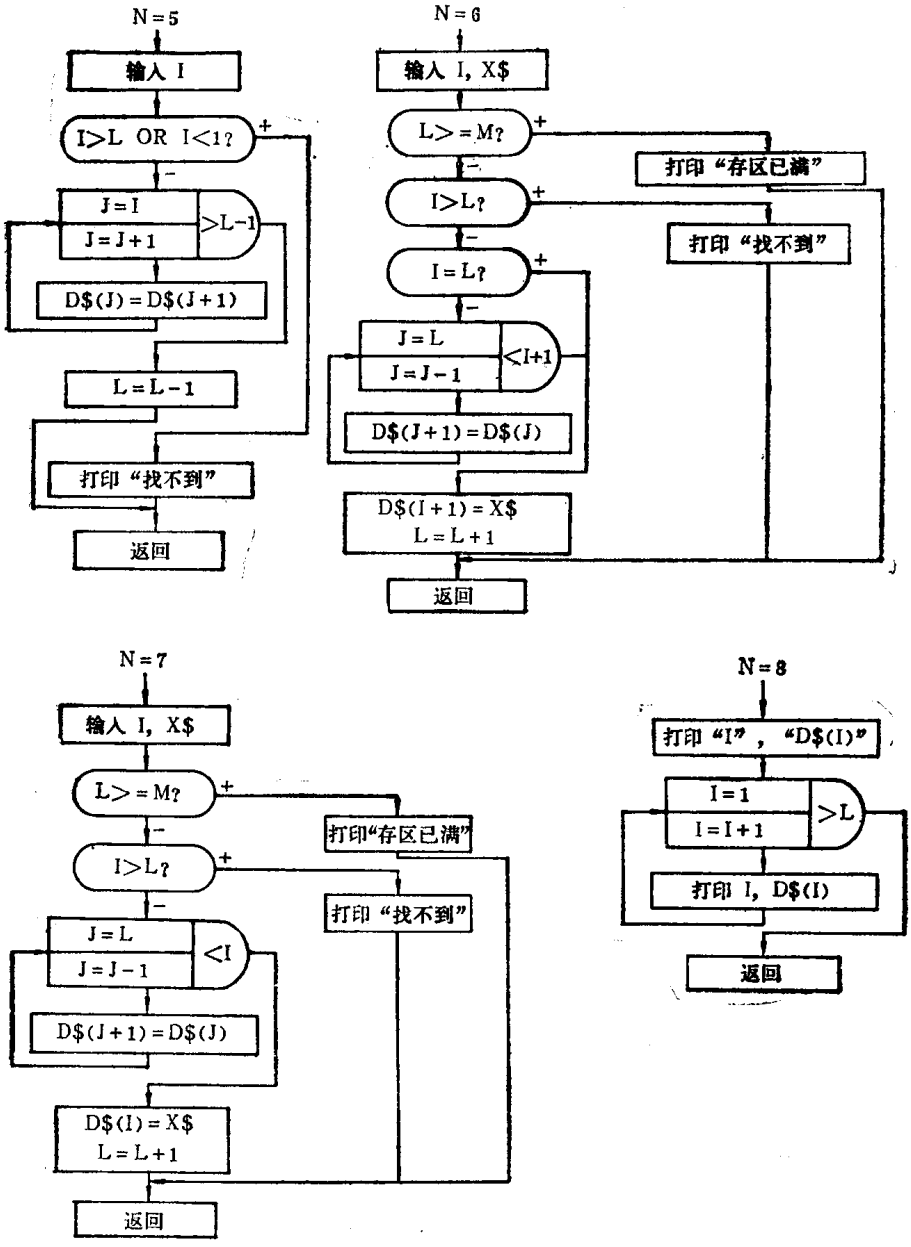
X$——新结点或需检索结点的关键字

I——地址

L——表中已存



框图 1-1　顺序表的基本操作(1)



框图 1-1　顺序表的基本操作(2)

放的结点总数

M——存区容量

J——循环变量

N——分支用变量

N = 1：求线性表中结点个数并打印

N = 2：按关键字检索并打印

N = 3：按第 I 个结点检索并打印

N = 4：修改第 I 个结点的关键字

N = 5：删除第 I 个结点

N = 6：在第 I 个结点后插入一新结点