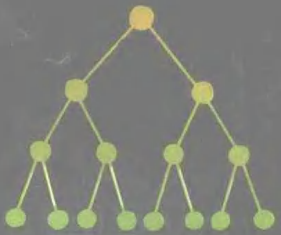
高等学校
工科电子类 规划教材

数据结构

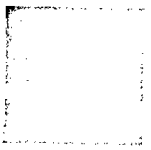
潘道才 陈一华



电子科技大学出版社

数据结构

潘道才　陈一华



电子科技大学出版社
·1994·

[川]新登字016号

内 容 提 要

本书是1988年由电子工业部"大专计算机教材编审委员会专业基础课编审小组"组织评选和推荐出版的教材。此次是在其基础上进行修编的，在保持原书特点和优点的基础上，考虑到本学科的发展和大专教育的特点，注意突出重点，削枝强干，加强应用和重视实践能力的培养，对原书作了一定的增删和改写。

主要内容：线性表的存储结构和插入、删除运算；栈、队列的存储结构及应用；串的基本运算；数组、稀疏矩阵的存储和有关操作；树、二叉树的性质、存储结构、遍历及应用；图的存储结构和遍历、生成树、最短路径、拓扑排序；顺序表、树表和哈希查找；几种排序方法；数据结构应用示例。

本书可作为大专类计算机软、硬件专业的教材，也可作为使用计算机的广大科技人员的自学参考书。

数据结构

潘遂才　陈一华

*

电子科技大学出版社出版

（中国成都建设北路二段四号）邮编 610054

电子科技大学出版社印刷厂印刷

四川省新华书店经销

*

开本 787×1092　1/16　印张 17.125　字数 416千字

版次　1994年5月第一版　印次　1994年5月第一次印刷

印数 1—8000册

中国标准书号　ISBN 7-81016-750-2/TP·55

定价：14.50元

出版说明

根据国务院关于高等学校教材工作的规定，我部承担了全国高等学校和中等专业学校工科电子类专业教材的编审、出版的组织工作。由于各有关院校及参与编审工作的广大教师共同努力，有关出版社的紧密配合，从1978～1990年，已编审、出版了三个轮次教材，及时供给高等学校和中等专业学校教学使用。

为了使工科电子类专业教材能更好地适应“三个面向”的需要，贯彻国家教委《高等教育“八五”期间教材建设规划纲要》的精神，“以全面提高教材质量水平为中心，保证重点教材，保持教材相对稳定，适当扩大教材品种，逐步完善教材配套”，作为“八五”期间工科电子类专业教材建设工作的指导思想，组织我部所属的九个高等学校教材编审委员会和四个中等专业学校专业教学指导委员会，在总结前三轮教材工作的基础上，根据教育形势的发展和教学改革的需要，制订了1991～1995年的“八五”（第四轮）教材编审出版规划。列入规划的，以主要专业主干课程教材及其辅助教材为主的教材约300多种。这批教材的评选推荐和编审工作，由各编委会或教学指导委员会组织进行。

这批教材的书稿，其一是从通过教学实践、师生反映较好的讲义中经院校推荐，由编审委员会（小组）评选择优产生出来的，其二是在认真遴选主编人的条件下进行约编的，其三是经过质量调查在前几轮组织编写出版的教材中修编的。广大编审者、各编审委员会（小组）、教学指导委员会和有关出版社，为保证教材的出版和提高教材的质量，做出了不懈的努力。

限于水平和经验，这批教材的编审、出版工作还可能有缺点和不足之处，希望使用教材的单位、广大教师和同学积极提出批评和建议，共同为不断提高工科电子类专业教材的质量而努力。

机械电子工业部

电子类专业教材办公室

前　言

本教材系按机械电子工业部工科电子类专业教材 1991－1995 年编审出版规划，由机械电子部教材办公室、中国电子工业总公司教育局全国大专计算机专业教材编审委员会专业基础课编审小组推荐出版。责任编委为赵晓彬同志。

本教材由上海科技高等专科学校潘道才同志、陈一华同志主编，西安电子科技大学朱倩荣同志主审。

本教材是在 1988 年由大专计算机专业教材编审委员会征稿并推荐出版的《数据结构》（潘道才主编）教材的基础上进行修编的。

本课程的参考学时数为 72 学时，其主要内容为数据的逻辑结构、物理结构以及对每种结构所定义的运算及应用。其中包括线性表、串、数组、广义表、树、图、文件等数据结构。对于同一种逻辑结构的数据，讨论了不同的存储结构及相应的有关算法和 PASCAL 源程序。本教材偏重于数据结构的应用，所以除在各章节中通过实例介绍数据结构的应用外，还专门讨论了查找和排序的方法，在第十章中举例讨论了综合应用数据结构的例子。在每章中都为学生提供了一定份量的习题并标明了应上机调试的题目。

本次修编中，在保持原书特色和优点的基础上，考虑到本学科的发展及大专教育的特点，删去了原书中“文本编辑”、“二叉排序树中删除结点”、“判定树”、“连通分量”、“银行业务活动的模拟”等内容，增加了“抽象数据类型”、“串的模式匹配”、“三维图形信息的压缩存储”、“构造最小生成树的普里姆算法”、“分块查找”、“递归和回溯”等内容及其他一些应用例子，并对“树的应用”、“构造最小生成树的克鲁斯卡尔算法”、“每一对顶点间的最短路径”、“B 树”、“什么是哈希法”等内容作了改写。本次修编，注意突出重点、加强应用、重视实践能力培养。前面加有“＊”号的章节供读者阅读，可不作为讲授内容。

本教材对原书增删及修改工作主要由陈一华同志担任，潘道才同志统编全稿。参加审阅工作的还有袁荣喜同志、杜舜国同志，他们都为本书提出许多宝贵意见，这里表示诚挚的感谢。由于编者水平有限，书中难免还存在一些缺点和错误，殷切希望广大读者批评指正。

编　者

一九九二年十一月

目　录

第一章 绪 论

1.1 数据结构课程的形成和发展

"数据结构"(Data structure)是一门随着计算机科学的发展而逐渐形成的新兴学科。

早期的电子计算机主要用于数值计算。后来,随着计算机软、硬件的发展,计算机的应用范围越来越广泛,其处理的对象也从简单的数字、字符发展到文字、图象、声音等各种复杂的具有一定结构关系的数据。同时,要处理的信息量也越来越大。让我们看以下两个例子。

例 1 教材的计算机管理问题

高校的教材科要负责全校教材的预订、采购、发放和管理等工作。所以,经常要查询教材的库存量等信息和作各种统计,有关部门和教师有时也需了解某专业所用的全部教材名称或某本教材的诸如作者、单价、出版日期等信息。如何利用计算机来解决这些问题呢?首先要考虑的就是对教材的各种信息如何加以组织和存储?一种常用的方法是建立一个表(文件),每本教材的各种信息占表的一行,为一个数据元素。如图 1.1 所示。这样,上面的问题就归结为计算机对一个表的有关操作,用户可以根据需要,按各种不同的项目进行查询或统计,从而快速、准确地得到结果。

编号	教材名	作者	出版日期	出版社	单价	适用专业	库存量

图 1.1 教材数据元素

例 2 考虑一个古典问题。设有 n 个正常人和 n 个精神病患者要过一条河,现只有一条能容纳 c(c<2n)人的小船,为防止患者伤害正常人,要求无论在河的哪一边,正常人的个数不得少于患者的个数(除非正常人个数为 0)。又设每个人都会划船,试设计一个过河的最佳(小船来回次数最少)方案。

这样的问题是比较复杂的,我们先构造出相应的数学模型。用一个三元组(x,y,t)表示渡河过程中的某个状态。其中,x 表示起始岸上正常人的个数,y 表示起始岸上患者的个数,t 表示小船的位置:

$$t=\begin{cases}1 & \text{表示小船在起始岸边}\\ 0 & \text{表示小船在目的岸边}\end{cases}$$

我们构造一个图,图中每一个顶点代表一个合法状态(不难推得,合法状态所对应的三元组(x,y,t)必须满足:x=0 OR x=n OR x=y),图中的边则表示该边所依附的两个顶点(即两个合法状态)间可由一次划船而相互交换。

例如,当 n=2,c=2 时,各合法状态及其变换如图 1.2 所示。

于是,过河方案的求解就转换成一个图的搜索问题,即搜索一个图,找出从起始顶点(n,n,1)到目的顶点(0,0,0)的一条包含边数最少的通路(若该通路存在,否则给出无解的信息)。

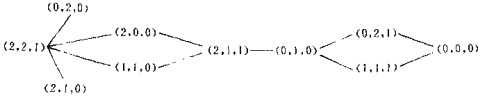


图 1.2 各合法状态及其变换图

对于图 1.2,下面的通路是最佳方案之一:

(2,2,1)⟶(1,1,0)⟶(2,1,1)⟶(0,1,0)⟶(0,2,1)⟶(0,0,0)

易知,最佳方案不唯一。

从上述的两个例子可见,许多实际问题是不能用数值计算来解决的,而且,它们的数据有时是大量的。所以,要使计算机高效正确地解决问题,对数据如何表示、组织、存储以及如何操作等问题的研究就显得越来越迫切,越来越重要了。

六十年代初期,还没有独立的"数据结构"课程,但有关的内容已散见子操作系统、编译原理和表处理语言等课程之中。1968 年,美国一些大学的计算机科学系的教学计划中,确定"数据结构"为一门课程,但对课程的内容范围仍没有作明确规定。当时,数据据结构几乎和图论,特别是表和树的理论是同义语。以后,数据结构的概念被不断扩充,包括网络、集合代数论,关系等现在称之谓"离散数学结构"的内容,它与现在数据结构的某些内容合在一起,被称为"数据结构"。由于数据的加工处理是在计算机中进行的,因此,除了研究数据本身的数学性质外,还必须考虑数据的存储结构,这就进一步扩大了数据结构的内容。自从 1968 年,美国计算机科学的著名教授 D. E. Knuth 所著的"计算机程序设计技巧"(The Art of Computer Programming)问世以后,逐渐将数据的数学概念及性质等内容独立出来,形成了现在的"离散数学结构",而把数据的逻辑结构、存储结构以及对每种结构所定义的运算组成了"数据结构"的主要内容。此后,各大学纷纷开设"数据结构"课程,我国高校在七十年代后期开始陆续开设该课程。

"数据结构"与数学、计算机软、硬件,特别是计算机软件有着密切的关系,它是计算机专业的一门核心课程,是"编译原理"、"操作系统"、"数据库"等课程的基础,也是设计和实现系统程序及大型应用程序的重要基础。

数据结构在计算机科学中的地位是十分重要的,随着计算机科学的飞速发展,特别是人工智能、图形和自然语言的计算机处理等领域研究的不断深入,"数据结构"这门新兴的学科将更显出勃勃生机,得到进一步的发展。

1.2 数据结构与算法

著名的计算机科学家,PASCAL 语言的发明者 N. 沃思(Niklaus Wirth)教授曾提出一个有名的公式:

算法+数据结构=程序

它清楚地揭示了算法和数据结构这两个计算机科学的重要支柱的重要性和统一性。我们不能离开数据结构去抽象地分析求解问题的算法,也不能脱离算法去孤立地研究程序的数据结构。N. 沃思教授还说:不了解施加于数据上的算法就无法决定如何构造和组织数据,反

之，算法的选择常常在很大程度上要依赖于作为基础的数据结构。

1.2.1 什么是数据结构

在计算机科学中，数据(data)是计算机程序加工处理的对象。抽象地说，数据是对客观事物所进行的描述，而这种描述是采用了计算机系统能够识别、存储和处理的形式来进行的。例如，一个计算几何图形面积的程序，其加工处理的数据是实数和整数；一个编译程序所处理的数据是字符串，因此，对计算机而言，数据的含义极为广泛，如数字、字符、图形、色彩、声音等都是数据。

我们把组成数据的基本单位称为**数据元素**(data element)，数据结构要研究的数据不是一、二个孤立的数据，而是大量的相互关联的数据，数据元素之间存在的相互关系称为结构。数据元素之间抽象化的相互关系称为数据的逻辑结构，这种相互关系可用一组运算及相应的运算规则来描述。通常，把这种数据逻辑结构简称为数据结构。例如：有一本电话号码本，我们相应地构造一个一维数组，它的每个数据元素是一个数对(a_i,b_i)。其中，a_i 和 b_i 分别表示第 i 个用户的名称和对应的电话号码。对于这个一维数组：

$(a_1,b_1),(a_2,b_2),\cdots,(a_i,b_i),\cdots,(a_n,b_n)$其中，n 为用户个数。我们可以定义若干运算及相应的运算规则。如进行查找，插入或删除某个数据元素等。这样，就构成了上述电话号码本的数据结构。

在设计算法和程序时，不仅要考虑数据的逻辑结构及其运算，而且要考虑这些数据在计算机存储器中的存储方式。这就是数据的物理结构或称存储结构。一种逻辑结构可以通过映象得到与它相应的存储结构。本书中将要讨论的逻辑结构有线性的，包括栈、队列和串等；也有非线性的，包括树和图。它们分别通过顺序和非顺序的映象，可以得到不同的存储结构。

我们从集合论的观点出发，可以对数据结构作出形式化的描述。数据结构是一个二元组，即 D.S=(D,R)

其中，D 是数据元素的集合，R 是 D 上关系的集合。两者的有机结合，就是数据结构。"数据结构"要研究的不仅是数据的逻辑结构和物理结构，还要研究相应结构上数据的有关运算。

在数据结构中，往往涉及数据类型(data type)和数据对象(data object)的概念。数据类型是指某种程序设计语言所允许使用的变量种类。如在 PASCAL 语言中，有整型、实型等简单数据类型，还有数组、记录等复杂数据类型和指针类型。一个数据类型不仅定义了相应变量可以设定的值的集合和存储方法，而且还规定了对变量允许进行的一组运算及其规则。所以，我们可以把数据类型看作是程序设计语言中已经实现了的数据结构。数据对象是指某种数据类型的数据元素的集合，是数据的一个子集。如整数的数据对象是集合 I={0，±1，±2，…}，英文字母的数据对象是集合 C={A，B，…，Z}。

随着计算机科学的不断发展，特别是面向对象的程序设计语言的研究和发展，提出了抽象数据类型(abstract data type，简记为 ADT)的概念。通俗地说，抽象数据类型是程序设计语言中数据类型概念的推广，是定义了一组运算的数学模型。

在设计复杂软件时，往往要采用抽象的思维方法，即抓住问题的本质而忽略其细节和其它非本质的东西，以便从宏观上把握全局。一个软件系统可以看作是由一些数据、操作过程和接口控制所组成，当使用自顶向下的方法设计软件时，要对它们进行抽象，即数据抽象，过

程抽象和控制抽象，抽象数据类型不仅包含数学模型，还包含了模型上的运算，所以，它不但发展了数据抽象的概念，而且将数据抽象和过程抽象结合起来。一个抽象数据类型确定了一个模型，但将构成模型的具体细节加以隐蔽；它定义了一组运算，但将运算的实现过程又隐蔽了起来。

运用抽象数据类型的概念设计软件的过程可用图 1.3 来表示：

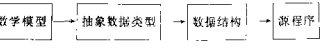


图 1.3　软件设计过程

抽象数据类型概念的引入，降低了大型软件设计的复杂性，并使软件设计中普遍遵循的模块化，信息隐蔽、代码共享等思想得到更充分的体现。所以，抽象数据类型是计算机科学的一个重要的新概念和重大的成果。

1.2.2　算法的概念和特性

算法(algorithm)的意义非常类似于处方、过程、方法和规程，通俗地说，所谓算法是指为解决给定问题而需施行的有穷操作过程的描述。在计算机科学中，算法则是描述计算机解决给定问题的操作过程。

通常，一个算法必须具备以下五个重要特性：

1. 有穷性

有始有终是算法最基本的特征。一个算法必须在它所涉及的每一种情形下，都能在执行有穷步操作之后结束。有的运算过程，操作步骤看似有限，但不能保证它在动态环境下实际执行次数的有穷性。因此，判断一个算法的有穷性应对算法所涉及的各种情形作动态分析，从而作出判断。看下面两个例子。

例 1　一个非算法的计数过程

(1)　开始

(2)　n←0

(3)　n←n＋1

(4)　重复(3)

(5)　结束

粗粗一看，这个过程仅有五个步骤，是有穷的，但事实上，该过程一开始，就永不休止。因而在动态执行中它并不具备有穷性，它不是一个算法。当然，只要对上述过程稍加修改，限定其计数之上限，即可使之成为一个算法。这就是下面的例 2。

例 2　不超过一万次的计数算法

(1)　开始

(2)　n←0

(3)　n←n＋1

(4)　若 n＝10000，则顺次执行(5)，否则重复(3)

(5)　输出 n

(6) 结束

2. 确定性

算法的每一步操作，其顺序和内容都必须确切定义，而不得有任何歧义性。

3. 数据输入

一个算法有 $n(n\geqslant 0)$ 个初始数据的输入。

4. 信息输出

算法是用来解决给定问题的，所以，一个算法必须将人们所关心的信息输出。也就是说，一个算法必须有一个或多个有效信息的输出，它是同输入数据有某种特定关系的信息。

5. 能行性

一个算法的任何一步操作都必须是可行的。即必须是可以付诸实施并能具体实现的基本操作。也就是说，每步操作均能由计算机(或人们用纸和笔)操作有限次即可完成。

1.2.3 数据结构与算法的关系

数据结构和算法是计算机科学的两个重要支柱。它们是一个不可分割的整体。

计算机求解问题的过程，从某个角度来说，无非是

初始数据输入——→计算机处理——→结果输出

对于一个给定问题的初始数据，如何组织、在计算机内如何存储，以节约存储空间和便于计算机处理，同时，如何选择合适的算法以提高求解问题的可靠性和效率，这都是至关重要的。我们不能强调了一面而轻视另一面，也不能把两者分开来作孤立的讨论。

比如前面所举的教材的计算机管理问题，我们用一个表来存储初始数据，现要查询《数据结构》教材的库存量以决定是否要征订，如何进行呢？如果初始数据在表内的存放是随机的，即没有什么规律的，那么，我们的查询算法只能是在表中顺序地逐行逐行地查找，显然其效率是不高的。如果想提高效率，则必须对数据的组织和存储作相应的调整和改进，比如，可以让教材按教材名的字典顺序排列或按专业归类存储，并建立索引表，同时选择更有效的算法。这样，就可大大提高查询的速度。

由此可知，不同的算法必须选用相适应的数据结构才能发挥作用，也就是说，数据结构的不同直接影响算法的选择和效率。

所以，对于一个特定问题，究竟采用何种数据结构和选用什么算法，不能一概而论，需要具体问题具体分析。要看问题的具体要求和现有的各种条件，把数据结构和算法这二者有机地结合起来考虑，这样才能设计出较好的计算机程序。

1.3 关于算法描述和分析的说明

数据结构是一门具有较强实践性的学科。通过该课程的学习，应使学生能运用数据结构的知识和技巧更好地进行算法和程序的设计。所以，我们在讨论各种数据结构的基本运算时，都给出了相应的算法，对于算法的描述，我们力求做到通俗易懂和适于自学，所以采用文字框图进行图示，读者在掌握和理解了框图所示的设计思想后，可以较方便地使用自己熟悉的算法语言来编制源程序。另外，考虑到用 PASCAL 语言来编写程序可以较好地体现结构程序设计的一些原则，并且简明易懂，便于教学，且具有实用价值，所以本书中大部分算法在给

出文字框图的同时还给出了用 PASCAL 语言编写的源程序片断。其中包括常量说明、类型说明以及用过程或函数形式给出的子程序。但对一些较长的程序则收在附录中。(源程序中出现的<=即≤,<>即≠,>=即≥)。书中虽然仅给出程序片断,但程序本身已在计算机上调试通过。限于时间和作者的水平,算法的实现并不一定精练。

关于 PASCAL 语言的内容,读者可以参阅有关书籍,这里不作介绍。本书框图中使用的图形符号见图 1.4。它们各自的意义是

1. 椭圆框:框中用文字标明算法的"开始"或"结束"。

2. 矩形框:框内描述某些操作,如赋值、组织循环等。统称为操作框。

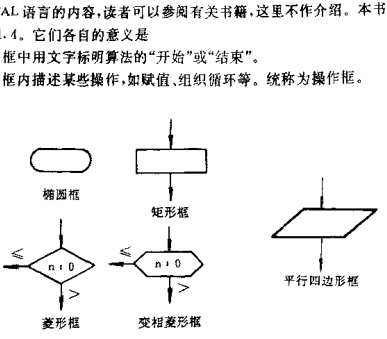


图 1.4 框图所用图形

3. 平行四边形框:表示输入/输出操作,即提供运算所需的数据或记录运算的结果。

4. 菱形框(包括变相菱形框):是判别框。框中符号":"表示比较。例如 n ? 0 表示 n 与 0 相比较,比较的结果写于框外连接线条的旁边。非箭头的线条表示算法或程序走向,写于其旁的判断结果就是算法或程序的分支走向应满足的条件。

另外,对算法(或程序)的分析和评价通常较复杂。一般需考虑正确性、最优性、可读性、可维持性、运算量及占用存储量等诸多因素。为了简化讨论,我们仅采用其中的两条标准。一是用问题的某个参数的函数来估算其存储量,其二是算法实现所需的运算量。假设问题的参数为 n。此参数可以是矩阵的阶、线性表的长度、图的顶点数等显示该问题规模大小的参数。那末,所选数据结构上执行某种操作所需要的存储量及运算量是 n 的什么函数呢?我们引进记号"O",对这些函数作数量级的估算。如,对于下面三个简单程序段:

```
(1)  x := x+1;
(2)  FOR i := 1 TO n DO
         x := x+1;
(3)  FOR i := 1 TO n DO
         FOR j := 1 TO n DO
             x := x+1;
```

设程序段 1 中,语句 x := x+1 不包含在任何循环体之中,则此语句只执行一次,运算量可记为 O(1)。在程序段 2 中,上述赋值语句在 FOR 循环之中,所以要执行 n 次,其执行时

间和 n 成正比，运算量可记为 O(n)。在程序段 3 中，x := x+1 语句要执行 n×n 次，其执行时间和 n^2 成正比，故运算量可记为 $O(n^2)$。然而，要事先对一个算法的运算量作仔细的分析很复杂，而且也不是本课程的主要内容，所以书中对一些算法的性能评价只根据算法中执行次数最多的语句来估算该算法的运算量的数量级，对于算法或程序所需的存储量，本书也作类似处理。

习 题 一

1. 举例说明什么叫数据、数据结构和数据类型？
2. 数据结构研究的主要内容是什么？
3. 分析下列程序的运算量

```
PROGRAM EX;
VAR i,j,p:integer;
BEGIN
  FOR i := 1 TO 9 DO
    write(i:4);
  FOR i := 1 TO 2 DO
    writeln;
  FOR i := 1 TO 9 DO
    BEGIN
      FOR j := 1 TO i DO
        BEGIN
          p := i * j;
          write(p:4)
        END;
      FOR j := 1 TO 2 DO
        writeln;
    END
END.
```

第二章　线性表

线性表（linear list）是一种最基本、最常用的数据结构。栈和队列则是线性表的特例，是加有某种限制条件的线性表。本章介绍它们的逻辑结构和两种主要的物理结构——向量和线性链表。并在此基础上，进一步介绍循环链表和双向链表的概念及应用。

插入和删除元素，这是线性表最基本、最重要的操作。本章重点讨论对于不同物理结构的线性表的插入和删除运算。最后一节讨论单元多项式的存储和相加这个线性表应用的典型例子。

2.1　基本概念和常用术语

先看几个线性表的例子，

例1　（1，2，3，4，5）是一个线性表，其中的数据元素是数字，共有五个数据元素。

例2　（A，B，…，Z）是一个线性表，其中的数据元素是英文大写字母，共有26个数据元素。

例3　图2.1所示的学生操行等级表也是一个线性表，其中的数据元素是每一个学生所对应的一行信息，包括姓名、学号、性别、年龄和操行等级共五个数据项。通常把这种数据元素称为记录。对每一个记录，可以有多个数据项，而用来标识一个记录或数据元素的数据项的值称为关键字（key）。如图2.1中，关键字可选用学号，因为学号可以唯一地标识一个学生。

综合以上例子，可对线性表作以下描述：

一个线性表是 $n\geqslant 0$ 个数据元素 a_1，a_2，…，a_n 的有限序列。表中每个数据元素，除第一个和最后一个外，有且仅有一个直接前趋和一个直接后继。也就是说，线性表或者是一个空表，或者可以写成：

$$(a_1,\ a_2,\ \cdots,\ a_i,\ \cdots,\ a_n)$$

姓　名	学　号	性　别	年　龄	操行等级
陈　敏	881201	女	19	优
张永平	881202	男	19	良
潘　寅	881203	男	20	优
⋮	⋮	⋮	⋮	⋮

图2.1　学生操行等级表

其中，a_i 是属于某个数据对象的元素。

由此可知，线性表具有以下特性：