华东高校计算机基础教育研究会推荐教材

高职高专

数据结构（C语言版）

黄保和 主编

黄保和 王 卓 李 青 郭敏学 编著

中国水利水电出版社

上海交通大学出版社

东南大学出版社

华东高校计算机基础教育研究会推荐教材

数据结构（C 语言版）

黄保和　主编

黄保和 王卓 李青 郭敏学　编著

中国水利水电出版社
上海交通大学出版社
东南大学出版社

内 容 提 要

本书共分为八章。第一章从逻辑结构、存储结构和数据运算三个方面介绍了数据结构的基本概念；第二至第六章介绍了线性结构，详细介绍了顺序表、链表、数组、栈、队列、串等各种常用数据结构及其查找和排序等基本运算；第七章介绍了树结构；第八章简单介绍了图结构。

本书专为高等职业技术院校计算机专业学生的《数据结构》课程而编写。本着注重应用的原则，本书选材精练，对基本理论的叙述深入浅出，通俗易懂；书中实例丰富，主要算法均给出C语言描述。为了便于教学，各章都配置适当的习题。

本书也可作为大专院校学生《数据结构》课程的教科书，或从事计算机应用的工程技术人员的自学参考书。

图书在版编目(CIP)数据

数据结构（C 语言版）/ 黄保和主编；黄保和等编著. —北京：中国水利水电出版社, 2000.8

华东高校计算机基础教育研究会推荐教材

ISBN 7-5084-0422-X

I. 数… II.①黄… ②黄… III.数据结构—高等学校—教材 IV.TP311.12

中国版本图书馆 CIP 数据核字（2000）第 65676 号

书　　名	数据结构（C 语言版）
作　　者	黄保和　主编
出版、发行	中国水利水电出版社（北京市三里河路 6 号　100044） 网址：www. waterpub. com. cn E-mail：sale@waterpub. com. cn 电话：（010）63202266（总机）、68331835（发行部）
经　　售	全国各地新华书店
排　　版	北京万水电子信息有限公司
印　　刷	北京北医印刷厂 印刷
规　　格	787×1092 毫米　16 开本　10.75 印张　236 千字
版　　次	2000 年 8 月第一版　2000 年 8 月北京第一次印刷
印　　数	0001—5000 册
定　　价	13.00 元

凡购买我社图书，如有缺页、倒页、脱页的，本社发行部负责调换

前　言

《数据结构》是计算机专业的一门专业基础课。当前，在各高等院校计算机专业的教学计划中，均把《数据结构》列入核心课程。

《数据结构》的基本内容是讨论数据在计算机中的表示和处理。通过《数据结构》的学习，学生得到必要的程序设计技能训练；另一方面，为学习《操作系统》、《语言编译》、《数据库系统》、《软件工程》、《人工智能》等后续课程提供必要的知识准备。因此，通过《数据结构》的学习，学生应达到两个目的：

1．知识方面 通过学习常用的数据结构（如线性表、栈、队列、二叉树、图等），了解数据结构内在的逻辑关系，了解数据结构在计算机中的表示方法。

2．技能方面 通过典型应用，理解数据在不同的存储结构中，实现各种数据运算的不同方法。通过算法设计和算法分析，提高学生分析问题和解决问题的能力，包括提高程序设计的能力。

本书主要读者是在校的高等职业技术院校学生。针对高职生面向应用，注重实践的特点，本书力求做到低起点，取材少而精。书中对各种常用数据结构的介绍尽量做到从实例出发，通过对实例的分析，使学生了解数据结构的基本概念。尽量避免抽象理论，避免复杂的公式推导。在算法介绍中，主要选取常规、简单的算法，舍弃结构复杂的算法。书中重要的算法都给出了 C 语言描述，为学生上机提供方便。本书的各章都附有习题，学生可通过完成习题，加深对基本概念的理解，特别是通过完成算法类的习题，提高编程能力和上机程序调试能力。

本书第一章、第四章、第六章和第七章由黄保和编写，第二章和第三章由李青编写，第五章由郭敏学编写，第八章由王卓编写，最后由黄保和统稿、修改和定稿。

上海工程技术大学施一萍老师、上海电气高等专科学校张小英老师参加大纲的讨论和制定，均为本书的参编者。

虽然本书作者都是从事本课程教学多年的教师，但由于水平有限，难免存在错误或不当之处，恳请各位老师批评指正。

作者

2000 年 6 月于厦门大学

目　　录

概论

本章介绍数据结构的有关概念，包括数据、数据元素、数据结构、逻辑结构、存储结构、数据运算、算法设计等基本概念。了解这些概念有助于对以后章节的深刻理解。

1.1 为什么学习数据结构

计算机是一种数据处理装置，把数据输入计算机，经过计算机处理之后得到人们需要的结果（处理结果还是数据）。数据是计算机处理的对象，对数据处理过程的描述称为算法。

计算机诞生初期，其运算速度较慢，存储容量有限，价格昂贵，主要应用于科学技术领域。在科学计算应用中，计算量大，但数据量少，数据结构简单。所以当时人们把程序设计的主要精力放在算法研究方面，并不关心（其实也没必要）对数据结构的研究，在当时，程序设计就等同于算法设计，认为程序就是用计算机语言表示的算法。

随着计算机技术的发展，60 年代以后，计算机大量应用于非数值计算领域，像银行业务处理，工商企业管理，政府部门管理等。计算机在这些领域应用的特点是需要处理大量的数据，但数据处理过程相对比较简单。如银行业务处理，需要输入、存储大量存款客户和货款客户的详细资料，数据量很大而且需要不断更新。而日常对这些数据的处理无非是分类、汇总、查找、计算利息等简单工作。面对大批量待处理的数据，人们为了对它进行有效处理，就必须研究数据的内在结构。

60 年代初期，各种高级程序设计语言相继出现。这些高级语言所能描述的数据类型逐渐丰富。例如 FORTRAN 语言允许使用复数类型和数组类型，COBOL 语言，PL/1 语言允许使用字符类型和记录类型。特别是数组类型和记录类型都已具备“结构”的雏型。数组把一批相关联的同类型的数据看作是一个整体，记录把一组相关联的不同类型的数据看作一个整体。有了结构的概念之后，各数据之间就不再是孤立的，而是相互联系的，这样便于表示，便于存储，也便于处理。

例如一个银行储户的基本情况包括姓名，地址、帐号、身份证号、开户日期、存款余额等项目的内容，把这些数据项组织起来就构成一个储户的基本资料，称为该储户的记录，把各储户的记录组合起来就是一个数组。把银行全体储户的数据按照它们之间的内在关系组织在一起，这就是一个数据结构。

60 年代中期，美国计算机界首先使用了数据结构这一名称。1968 年由美国计算机协会颁布的建议性计算机教学计划中，第一次规定了数据结构作为一门独立的计算机课程在大学里开设。同年，美国著名计算机专家唐·欧·克努特教授创立了数据结构的最初体系，他在《计算机程序设计技巧》第一卷《基本算法》一书中，全面系统地阐述了数据的逻辑结构和存储结构，并给出了各种典型的算法，为数据结构奠定了基础。

60 年代末 70 年代初，随着大型程序，大规模文件系统的出现，使得结构化程序设计方法和数据结构已不再是理论课题，它们在实际中受到极大的重视。人们认为，程序设计的本质是对要处理的问题选择一种好的数据结构，同时在此结构上施加一种好的算法。1976 年，瑞士著名计算机科学家，PASCAL 语言的发明者 N.沃思出版了《算法+数据结构=程序》一书，该书就是这种程序设计思想的著名代表作之一。

在我国，80 年代初就已经在大学中开设《数据结构》这门课程。在现行的计算机专业教学计划中，已把《数据结构》课程列为核心课之一。《数据结构》是一专业基础课，是程序设计方法学、数据库系统、操作系统、编译系统、软件工程学等课程的先修课程。在实际应用中，它是设计、实现大型应用软件的重要基础。

1.2 什么是数据结构

上一节提到了数据结构和算法是程序设计中两个同等重要的问题，相辅相成，缺一不可。那么什么是数据结构呢？

简而言之，数据结构是指数据之间的关系。

在计算机数据处理过程中，大批量的数据并不是彼此孤立，杂乱无章的。它们之间有着内在的联系。只有利用这些客观存在的关系，把各数据元素有机地组织起来，才有可能对这些数据进行处理，或者说才有可能对这些数据进行有效地处理。

例如我们考虑电话号码自动查找问题。电话号码查询最主要的工作是当给出单位或个人的名字时，能在电话号码表中迅速查找到其电话号码。当然，当有新用户要加入或旧用户要改号或撤消电话时，也要对电话号码表进行相应修改。那么应该怎样组织电话号码表，使得以上操作能够实现呢？最简单的组织方式如表 1-1 所示。把各用户的电话号码随机罗列出来，要查找某人或某单位的电话号码，只要从表中第一行开始，依次往后顺序查找，只要被查找者确实在册，就会找到。同样对新用户的添加，旧用户的修改，删除也可以操作。但采用这种方式查找，效率是很低的。

表 1-1 电话号码表

单位或个人名称	电话号码
市政府办公厅	5010123
永红电视机有限公司	4044001
新奇百货公司业务部	5678124
百利进出口公司运输部	8080888
百胜律师事务所	2524254
李百万	7654321
丁一	2185555
钱万金	6278782
……	……

为了提高查找效率，我们可以这样组织电话表：把单位电话和个人电话分开。单位电话按行业组织，把同行业的电话登录在一起；个人电话按姓氏笔划顺序登录，并各自建立一张

索引表，如表 1-2 所示。在表 1-2 中，假设要查找王五的电话，因是个人电话，先到表 1-2（b）中查找，找到王姓电话在号码在表的第 56001 行到 57012 行之间，然后再从表 1-2（c）的 56001 行开始往后查找，直至找到王五的名字或查找到 57012 行没有找到王五的名字，查找结束。

显然，在表 1-2 中查找电话号码，查找范围缩小了，查找速度加快了。

表 1-2 就是利用数据之间的关系来组织数据，从而使得数据处理效率提高。

表 1-2 带索引的电话号码表

（a） 单位电话索引表

部门、行业	行号范围
市政府	1-1292
工业	1293-3720
商业	3721-8970
运输	8971-12345
……	……

（b） 个人电话索引表

姓氏	行号范围
丁	55550-55600
马	55601-55762
叶	55763-56000
王	56001-57012
……	……

（c） 电话号码表

行号	单位或个人名称	电话号码
1	市政府办公厅	5010123
2	市政府信访处	5010110
……	……	
1292	市政府办公厅秘书	5011254
1293	永红电视机有限公司	4044001
……	……	
56001	王一民	7272001
……	……	
57012	王鑫	7767776
……	……	

由于表 1-1 和表 1-2 的数据组织方式不同，进行同样的查找工作，其查找算法也各不相同，查找效率当然也不同。

要在计算机上实现电话号码自动查询，还得把以上数据存储到计算机存储器中。到底该用什么方式存储，使之能体现出数据之间的关系，使得数据处理能正确、高效完成，这也是数据结构应讨论的问题。

可见，数据结构要讨论的问题应包括数据的逻辑结构、数据的存储结构及数据运算的表示三个方面。

把电话号码以表 1-1 的方式组织，并把该表以数组的方式存储到计算机中以便进行自动电话号码查询。这就是一个数据结构。类似地，表 1-2 也是数据结构。

表 1-1 称为线性表，是一个最简单，最常用的数据结构。

线性表的逻辑结构应能确定以下问题：哪个数据元素是表中的第一个元素；哪个元素是表中的最后一个元素；某指定元素的前一个元素是哪一个元素，后一个元素又是哪一个元素。

线性表的存储结构是指这样的内容：线性表中各元素在存储器中是顺序存储（用存储单元地址的连续性体现线性表的逻辑关系），还是链接存储（用指针链接体现数据元素间的逻辑关系）。

数据运算包括在表中查找一个元素，在表中插入一个元素或在表中删除一个元素应如何操作等。

综上所述，按某种关系组织起来的一批数据，以一定的存储方式把它们存储到计算机的存储器中，并在这些数据上定义一个运算集合，这就是数据结构。

1.3 数据的逻辑结构

1.3.1 基本概念

数据 能输入计算机且能被计算机处理的一切对象。如整数、实数、字符、文字、逻辑值、图形、图像、声音等都是数据。数据是信息的载体，是对客观事物的描述。

数据元素 数据元素是对现实世界中某独立个体的数据描述，是数据的基本单位。如前面列举的一个银行储户资料就是一个数据元素，电话号码表中的一行也是数据元素。数据元素一般由一个或多个数据项组成。数据元素有时也称为结点，记录或表目。

数据项 具有独立意义的最小数据单位，是对数据元素属性的描述。数据项也称域或字段。在表 1-1 中，每个数据元素由两个数据项组成，其中“单位或个人名称”数据项描述了电话所有者的名字，“电话号码”数据项描述了该电话的号码。

数据类型 每个数据项属于某确定的基本数据类型。如银行储户资料中，姓名是字符串型，开户日期是日期型，存款余额是实数型等。一个数据元素如果只包含一个数据项，则数据项的类型就是该结点的类型，否则数据元素的类型是由各数据项类型构造而成的构造类型，在 C 语言中称为结构体类型。各种高级语言提供的基本数据类型有所不同，如 C 语言提供了整型，实型，字符型和指针型等基本数据类型。

数据对象 具有相同特征的数据元素的集合。例如电话号码查找系统中，数据对象就是全体的电话用户；银行业务处理中，数据对象就是全体储户资料及全体贷款客户的资料。

关系 在数据对象中各数据元素之间存在着某种关系（或称联系）。这种关系反映了该数据对象中数据元素所固有的一种结构。在数据处理领域，通常把数据之间这种固有的关系简单地用前驱，后继关系来描述。例如在编写族谱时，数据对象是家庭中的所有成员，对家族中某成员的描述就是一个数据元素，各数据元素之间存在着血缘关系。父亲是儿子的前驱，儿子是父亲的后继，小孩子没有后继。又如考虑一张按名次排列的成绩表，数据对象是全班同学，对某个同学属性（姓名，成绩等）的描述就是一个数据元素，各数据元素之间存在着名次的关系，第一名没有前驱，其后继是第二名，第二名的前驱是第一名，其后继是第三名。可见，前驱后继关系所表示的实际意义随着数据对象的不同而不同。一般地，数据元素之间的任何关系都可用前驱，后继关系描述。

某结点（数据元素）如果没有前驱，则称该结点为根结点或起始结点；某结点如果没有后继，则称此结点为终端结点。

1.3.2 数据的逻辑结构

上节已经介绍过数据的逻辑结构，下面用数学方法给出逻辑结构的定义：

设 D 表示数据元素的集合，R 表示 D 上关系的集合（即 R 反映了 D 中各元素的前驱，后继关系），则一个数据结构 B 可以表示为：

B=（D，R）

可见数据结构由两部分构成：

（1） 表示各数据元素的信息 D；

（2） 表示各数据元素之间关系的信息 R。

一般用二元组表示 D 中各数据元素之间的前驱，后继关系。例如假设 a,b 是 D 中的两个数据元素，则二元组<a,b>表示 a 是 b 的前驱，b 是 a 的后继。

【例 1】 一周七天的数据结构可表示为：

```
B= (D,R)
D={Sun, mon, tue, wed, thu, fri,sat}
R={<Sun,mon>, <mon,tue>, <tue, wed>,<wed,thu>,<thu,fri>,<fri,sat>}
```

以上数据结构可用图 1-1 形象地表示。

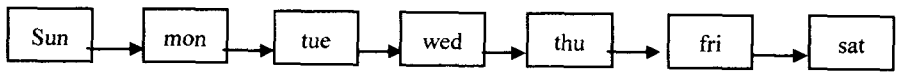


图 1-1 一周七天数据结构图示

图 1-1 称为数据结构的图形表示，图中方框（有时也用圆框）表示一个结点，框内文字是该结点的值，带箭头的线段表示前驱，后继关系。

【例 2】 如图 1-2 所示，是一个具有 10 个结点和两个关系的数据结构，每个结点包括两个数据项：名称和年龄。两个关系分别为直系血缘关系 R1 和第三代各兄弟年龄大小排列顺序关系 R2。在图 1-2 中，R1 用双实线表示，R2 用单实线表示。方框内的内容是结点的值，方框外的文字是结点的名称。

```
B= (D, R)
D={ai|1≤i≤10}={a1,a2,a3,a4,a5,a6,a7,a8,a9,a10}
R={R1,R2}
R1={<a1,a2>,<a1,a3>,<a1,a4>,<a2,a5>,<a2,a6>,<a2,a7>,<a3,a8><a4,a9>,<a4,a10>}
R2={<a5,a6>,<a6,a8>,<a8,a9>,<a9,a7>,<a7,a10>}
```

本书的几个约定：

（1）数据元素也称结点、记录、表目、顶点等，以后一般称为结点。

（2）在实际应用中，结点都是由若干个数据项组成的，即结点类型是一个结构体类型。因为在数据结构讨论中，我们研究的重点是结点之间的关系，并不涉及结点的内部结构。为方便起见，以后如果没有特别说明，都假设结点的类型为整型（int）。

（3）通常一个数据结构都存在着若干个关系，为方便讨论我们都假设一个数据结构只存在一个关系。如果一个数据结构包含有多个关系，可以用类似的方法分别讨论。

（4）在不引起混乱的情况下，我们把数据的逻辑结构简称为数据结构，把数据的存储

结构简称为存储结构。

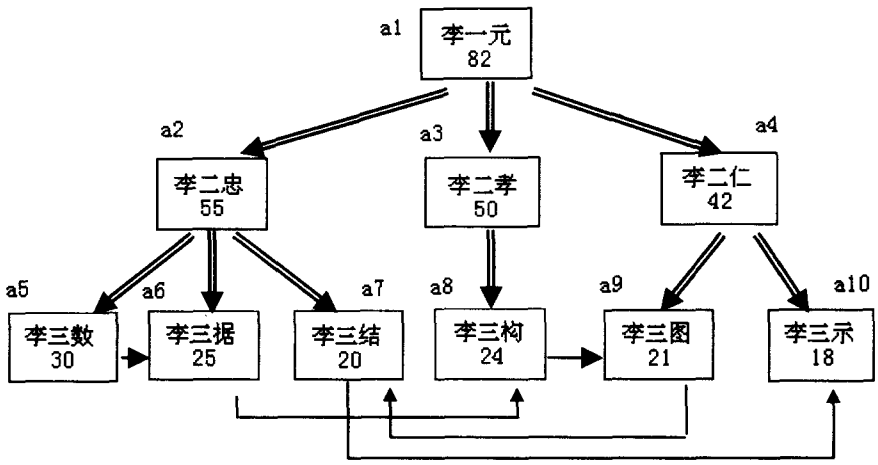


图 1-2 家庭成员数据结构图

1.3.3 数据结构的分类

根据数据结构中各结点之间前驱，后继结点的个数，我们把数据结构分为四种基本类型，它们分别为：

集合 数据结构中只有结点，结点之间不存在关系，即 R={ }。如图 1-3（a）所示。这是数据结构的一种特例，本书不予讨论。

线性结构 在该数据结构中，除了一个根结点外，其他各结点有唯一前驱；除一个终端结点外，其他各结点有唯一后继。是一种一对一关系。如图 1-3（b）所示。

树型结构 在该数据结构中，除了一个根结点外，各结点有唯一前驱；所有结点都可以有多个后继，是一种一对多关系。如图 1-3（c）所示。

图状或网状结构 在该数据结构中，各结点可以有多个前驱或多个后继。是一种多对多的关系。如图 1-3（d）所示。

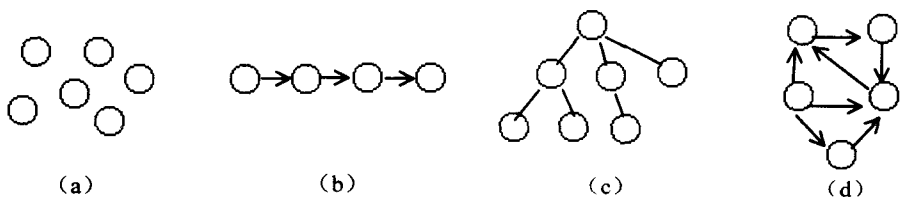


图 1-3 四种基本数据结构图

一般地，把树型结构和图状结构称为非线性结构。

在数据结构 B=（D，R）中，也可以不包含任何结点，即 D={ }，称为空数据结构。空数据结构到底属于哪种类型，应视具体情况而定。

1.4 数据的存储结构

前面都是从逻辑上观察分析数据元素之间的关系。但数据结构需要用计算机处理，必须把数据结构存入计算机的存储单元中。数据结构在计算机中的表示称为数据的存储结构，简称存储结构。

怎样既能合理利用有限的存储空间，又能使数据结构有效地处理，这就是存储结构要讨论的问题。

计算机的存储器（内存）是由很多存储单元（word 或 byte）组成的，每个存储单元都有唯一的地址。各存储单元的地址编码是线性连续的，即每一个存储单元有唯一的前驱单元和唯一的后继单元。我们把两个互为前驱后继的存储单元称为相邻存储单元，把一片相邻的存储单元称为存储区域。

一般情况下，一个结点所占用的空间并不止一个存储单元，不同类型的结点所占用的存储单元数量也各不相同。因为我们是在高级语言（C 语言）的层面上讨论数据结构，所以可以通过高级语言的类型说明，使一个结点对应一个存储单元，从而把问题简化。在以后的叙述中，如果没有特别地说明，存储单元都是指高级语言层面上的虚拟存储单元。例如我们可以用一个一维数组对应一个存储区域，一个数组元素对应一个存储单元。

数据结构包括结点的值及结点之间的关系，其存储结构除了必须存储结点的值外，还得能以直接或隐含的方式体现出结点之间的关系。

下面介绍四种基本的存储表示方法：

1. 顺序方式

顺序存储最适合于线性结构，它把逻辑上相邻的结点存放到物理上相邻的存储单元中。顺序存储结构只存储结点的值，不存储结点之间的关系，结点之间的关系通过存储单元的相邻关系隐含地表示出来。

【例 3】 B=（D，R）

D={A，B，C，D，E}

R={<A,B>,<B,C>,<C,D>,<D,E>}

存储结构如图 1-4 所示。

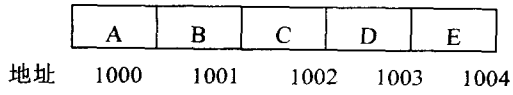


图 1-4 线性结构的顺序存储结构

顺序存储结构在 C 语言中用数组表示。

一些特殊的非线性结构也可以用顺序方式存储。

2. 链接方式

链接存储方式不仅存储结点的值，而且还存储结点之间的关系。它利用结点附加的指针字段，存储其后继结点的地址。

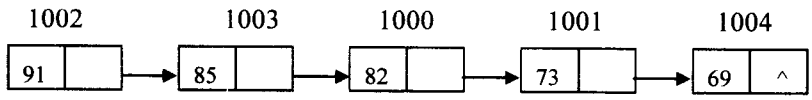
链接存储方式中结点由两部分组成，一部分存储结点本身的值，称为数据域，另一部分存储该结点的各后继结点的存储单元地址，称为指针域。指针域可以包含一个或多个指针，

这由结点之间关系的复杂程度决定。有时，为了运算方便，指针域也用于指向前驱结点的存储单元地址。

【例 4】 假设线性结构的结点集合 D={82，73，91，85，69}，以结点值降序为关系 R={<91，85>，<85，82>，<82，73>，<73，69>}，其链接存储结构如图 1-5 所示。

地址	data	link
1000	82	1001
1001	73	1004
1002	91	1003
1003	85	1000
1004	69	0

（a）



（b）

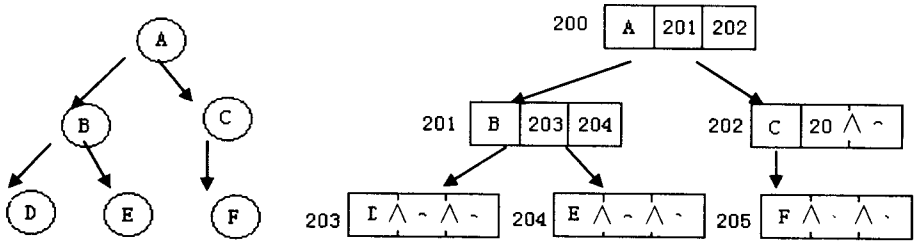
图 1-5 线性结构的链接存储

【例 5】 逻辑结构为 B=（D，R）

D={A，B，C，D，E，F}

R={<A,B>,<A,C>,<B,D>,<B,E><C,F>}∧

如图 1-6 所示分别是该树型结构的图形表示和链接存储结构。图中各结点有三个域，其中后两个域是指针域。



（a）树的图型表示（逻辑结构） （b）树的链接存储结构

图 1-6 树的逻辑结构及其存储结构

我们可以看到，同样的数据结构，采用链接存储比采用顺序存储占用更多的存储空间。这是因为链接存储方式增加了存储其后继结点地址的指针域。

我们把存储空间都完全被结点值占用的存储方式称为紧凑存储，否则称为非紧凑存储。显然，顺序存储是紧凑存储，链接存储是非紧凑存储。

用存储密度 d 来衡量结点值在整个数据结构存储中所占存储空间的比重：

$$d = \frac{\text{结点值的存储量}}{\text{整个结构的存储总量}}$$

显然，d 值越大，表示数据结构所占用的存储空间越少。紧凑存储的存储密度 d 为 1。

3. 索引方式

在线性结构中，各结点可以依前驱，后继关系排成一个序列（a1,a2,a3……an）。每个结点 ai 在序列中都对应一个序号 i，序号 i 也称为结点 ai 的索引号。索引存储就是通过建立一个附加的索引表，然后利用索引表中索引项的值来确定结点的实际存储单元地址。

索引存储有密集索引和分块索引两种。

密集索引：线性结构中每个结点对应索引表中的一个索引项，第 i 个索引项的值就是第 i 个结点的存储单元地址。

分块索引：把线性结构中若干个相邻的结点组合在一起称为一块。每块对应索引表的一个索引项，第 i 个索引项的值就是第 i 个结点块中第一个结点的存储单元地址。分块索引也称为非密集索引。1.2 节中电话号码查询就是采用分块索引存储方式。

只要按一定规则把非线性结构线性化，非线性结构也可以采用索引存储方式。

索引存储是解决各结点长度不同的数据结构存储的一种有效的方式。

4. 散列方式

散列存储方式利用结点的值确定该结点的存储单元地址。

把结点 a 中的某数据项（也称为关键字）a.key 作为自变量，通过一个称为散列函数 Hash（a.key）的计算规则，确定出该结点的实际存储单元地址。即：

loc（a）=Hash（a.key）

其中 loc（a）为结点 a 的地址。

散列存储将在第六章详细讨论。

一般的数据结构都可以采用上述基本存储方式之一存储，或者通过基本存储方式组合存储。一个数据结构可以采用不同的存储方式，到底选择什么存储方式，取决于数据运算的方便性和可利用的存储资源。

对给定的数据结构，一旦建立了存储结构，则结构中某结点 a 及存放该结点的存储单元就溶为一体，为叙述方便，有时就没有很严格的区分。例如当我们说到结点 a 的地址时，应该是指存储结点 a 的存储单元的地址，说到结点 a 的指针域时，应该是指存储结点 a 的存储单元的指针域等等。

1.5 数据运算和算法

1.5.1 数据运算

按一定的逻辑结构把数据组织起来，采用适当的存储方式把数据结构存储到计算机中，最终目的是为了有效地处理数据，提高数据的运算效率。

在实际应用中，不同的逻辑结构有不同的运算集合。不过常见的数据结构，都有一些相似的基本运算，下面介绍几种常见的数据运算。