普通高等教育“十一五”国家级规划教材

国家精品课程主讲教材

高等学校计算机科学与技术专业系列教材

数据结构与算法(第4版)

廖明宏　郭福顺　张　岩　李秀坤

高等教育出版社
Higher Education Press

普通高等教育“十一五”国家级规划教材

国家精品课程主讲教材

高等学校计算机科学与技术专业系列教材

数据结构与算法

（第4版）

廖明宏　郭福顺　张　岩　李秀坤

高等教育出版社

内容提要

本书是在教育部高等学校计算机科学与技术教学指导委员会制定的专业规范思想指导下，以哈尔滨工业大学国家精品课程“数据结构与算法”为基础，融入数据结构与算法的最新研究成果编写而成。本书为普通高等教育“十一五”国家级规划教材。全书按抽象数据型的观点组织，算法用类 C 语言描述，共分 8 章。第 1 章给出抽象数据型的定义、算法的基本概念及其复杂性的表示方法，扼要地介绍逐步求精的程序设计方法；第 2、3、4 章是对线性表、树、图等主要数据结构定义相应的抽象数据型，给出各种物理表示法和有关算法；第 5、6、7 章是关于数据处理技术的内容，介绍几种主要的查找和排序算法，同时还介绍文件的组织形式；第 8 章介绍几种典型的算法设计方法及其分析方法。

全书注意理论与实践相结合，内容深入浅出。本书可以作为计算机科学与技术专业的本科教材，同时也适用于计算机工程方向、软件工程方向和信息技术方向的本科教学；也可作为硕士研究生“算法设计与分析”课程的教学参考书，计算机学科相关专业的教材或参考书，同时对计算机科技工作者也有参考价值。

图书在版编目(CIP)数据

数据结构与算法/廖明宏等.—4 版.—北京:高等教育出版社,2007.11

ISBN 978-7-04-022473-3

Ⅰ.数… Ⅱ.廖… Ⅲ.①数据结构-高等学校-教材②算法分析-高等学校-教材 Ⅳ.TP311.12

中国版本图书馆 CIP 数据核字(2007)第 155055 号

策划编辑 刘 艳 **责任编辑** 彭立辉 **封面设计** 于文燕 **责任绘图** 黄建英
版式设计 张 岚 **责任校对** 王 雨 **责任印制** 毛斯璐

出版发行	高等教育出版社	**购书热线**	010-58581118
社 址	北京市西城区德外大街 4 号	**免费咨询**	800-810-0598
邮政编码	100011	**网 址**	http://www.hep.edu.cn
总 机	010-58581000		http://www.hep.com.cn
		网上订购	http://www.landraco.com
经 销	蓝色畅想图书发行有限公司		http://www.landraco.com.cn
印 刷	北京北苑印刷有限责任公司	**畅想教育**	http://www.widedu.com
		版 次	2000 年 6 月第 1 版
开 本	787×1092 1/16		2007 年 11 月第 4 版
印 张	19.5	**印 次**	2007 年 11 月第 1 次印刷
字 数	430 000	**定 价**	24.50 元

物料号 22473-00

高等学校计算机科学与技术专业系列教材编审委员会

序

计算机和通信技术的迅猛发展，不仅形成了融合度最高、潜力最大、增长最快的信息产业，而且成为推动全球经济快速增长和全面变革的关键因素。进入21世纪，我国的信息产业虽然已取得了长足的发展，但与发达国家相比，还有不小的差距。国家信息化的发展和信息产业国际竞争能力的提高，迫切需要高素质、创新型的计算机专业人才。

高素质计算机专业人才的培养离不开高质量的计算机教育。我们的专业虽然机会多，处于非常有利的条件，但是我们同样面临着一件事，就是从规模发展向质量提高的转变。怎么提高质量？专业素质的教育和应用素质的训练非常重要。尤其是我国高等教育进入大众化发展阶段，社会对计算机专业人才呈现出了多样化的需求。而与此同时，计算机学科的发展已极大地突破了原有的学科体系框架，形成了在“计算机科学与技术”之下向多个专业方向发展的新格局。在这种背景下，教育部高等学校计算机科学与技术教学指导委员会编制了《高等学校计算机科学与技术专业发展战略研究报告暨专业规范(试行)》(以下简称“专业规范”)。专业规范按照“培养规格分类”的指导思想，提出了三种类型、四个方向，即科学型(计算机科学方向)，工程型(计算机工程方向、软件工程方向)，应用型(信息技术方向)的计算机专业发展建议，体现了社会对不同人才类型的需求，对于指导我国计算机教学改革与建设，规范计算机教学工作，促进计算机教学质量的提高都具有重要的意义。

高水平的教材是一流教育质量的重要保证。为了配合专业规范的试行，便于广大高校教师按照新的专业规范组织实施教学，高等教育出版社在大力支持专业规范研究与起草工作的同时，还邀请规范起草小组的有关专家成立“高等学校计算机科学与技术专业系列教材编审委员会”，组织规划了结合计算机专业规范、面向全国高等学校计算机专业本科生的“高等学校计算机科学与技术专业系列教材”。令人高兴的是，一批有创新、改革精神，且有丰富教学经验的高等学校教师投身到新体系计算机专业教材的编写中来，他们用自己创造性的思维、辛勤的汗水诠释专业规范的思想，把新的课程体系和教学内容生动地传达给师生，并进行着有意义的教学实践。

“高等学校计算机科学与技术专业系列教材”以专业规范和CC2001－CC2005有关教程为依据，以强化基础、突出实践、注重创新为原则，体现了学科课程体系和教学内容改革的新成果。此外，这一系列教材还配有丰富的教学辅助资源，并与现代教育技术手段相结合，充分发挥网络平台的作用，使教材更有利于广大教师和学生使用。目前，这一系列教材有不少选

题已列入普通高等教育“十一五”国家级规划教材，希望这些教材的出版能够对新形势下我国高等学校计算机专业课程改革与建设起到积极的推动作用，使我国高校的计算机专业教学质量再上一个台阶。

中国科学院院士

2006—2010 年教育部高等学校计算机科学与技术教学指导委员会主任

二 00 七年十一月

前言

本书是在教育部高等学校计算机科学与技术教学指导委员会制定的专业规范思想指导下，以哈尔滨工业大学国家精品课程“数据结构与算法”为基础，以郭福顺、廖明宏、李莲治编著的《数据结构与算法基础》为蓝本，经去粗取精，并融入数据结构与算法的最新研究成果编写而成。本书为普通高等教育“十一五”国家级规划教材。原书曾经是按电子工业部制定的工科电子类专业教材1986年—1990年编审出版规划，由《计算机与自动控制》教材编审委员会计算机编审小组组织征稿、评选、推荐出版，并于1992年获得第二届机械电子工业部电子类专业优秀教材二等奖。后来，随着教学的需要和教学经验的积累，又几次修改、再版。由于数据结构与算法之间的联系极其紧密，在编写本书时继承了原书中的主要内容，删除了不常用的部分，并将有关算法的基础性内容融入其中，使全书的各部分构成一个协调的整体。

全书按抽象数据型的观点组织，算法用类C语言描述，共分8章。第1章给出抽象数据型的定义，算法的基本概念及其复杂性的表示方法，扼要地介绍逐步求精的程序设计方法；第2、3、4章是对线性表、树、图等主要数据结构定义相应的抽象数据型，给出各种物理表示法和有关算法；第5、6、7章是关于数据处理技术的内容，介绍几种主要的查找和排序算法，同时还介绍文件的组织形式；第8章介绍几种典型的算法设计方法及其分析方法。

本书可以作为计算机科学与技术专业计算机科学方向的本科教材，同时也适用于计算机工程方向、软件工程方向和信息技术方向的本科教学；也可作为硕士研究生“算法设计与分析”课程的教学参考书，计算机相关专业的教材或参考书，同时对计算机科技工作者也有参考价值。当作为数据结构教材时，第8章及带星号的章节可作为选学内容。另外，根据学时的多少和讲授的深度，也可以选讲各章节中某些有关算法时间复杂性分析的内容。当本书作为数据结构与算法课程的教材时，第8章是课程的重要组成部分。全书既注重原理又注重实践，各章之后附有习题。

本书由廖明宏、郭福顺、张岩和李秀坤共同编写，并由廖明宏统编全稿。其中，李秀坤负责第1～3章，郭福顺负责第4章，张岩负责第5～7章，廖明宏负责第8章。全书各主要部分的内容，都经作者仔细审阅并取得一致意见。但由于作者水平有限，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。

作者

2007年1月于哈尔滨工业大学

目录

第1章 绪 论

学习目标

首先，通过对数据结构的基本概念、研究对象以及数据结构课程的发展历史的简单介绍，使学生对数据结构课程有一个宏观的认识。其次，引入一个贯穿全书的重要概念——抽象数据型，对其概念及实现方法做进一步介绍，使学生初步掌握抽象的技术和方法。最后，对算法、算法复杂性以及算法的逐步求精方法进行简单介绍，使学生掌握问题求解的基本方法和评价方法。

1.1 数据结构的研究对象

计算机科学是研究信息表示和处理的科学，信息在计算机内是用数据表示的。直观地说，数据是用于描述客观事物的数值、字符以及一切可以输入到计算机中并由计算机程序加以处理的符号的集合。数据的基本单位是数据元素，性质相同的数据元素的集合叫做数据对象。计算机程序通常作用在一组数据上，这里暂且称这组数据为信息表。程序在运行过程中一般要建立一些信息表，对表中的数据进行访问、加工，或在适当的时候将信息表注销。这些信息表中的数据元素并不是杂乱无章地堆积在一起，它们之间存在着一定的关系，数据元素之间的这种相互关系叫做结构。数据结构指的是数据元素之间抽象的相互关系，并不涉及数据元素的具体内容。这种结构关系将数据组织成某种形式的信息表，以便程序进行加工。因此，信息的表示和组织形式直接影响加工程序的效率，从而使得研究数据结构和研究算法很难完全分开。

下面通过具体的实例来介绍几个有关的概念，并说明数据结构的研究对象。这里只对这些概念做些说明，使读者有比较直观的印象并初步了解数据结构研究的对象是什么。

信息表的一种最简单的形式是线性表，其结构特性就是“线性”。在讨论线性表时将涉及一些与线性结构有关的问题，例如哪个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；对于表中一个给定的元素而言，哪些元素在它之前，哪些元素在它之后；在一个线性表中总共有多少个元素，等等。

更复杂一些的信息表，如多维数组、树结构等是非线性结构。一般情况下，树结构表示数据元素之间的层次关系或分支关系，是一种非线性的结构。

信息表是由结点组成的，每个结点由一个或多个相邻的存储单元组成，并分成若干个域，每

一个域都有一定的名称。在最简单的情形下，一个结点可能仅由一个存储单元组成，也可能只有一个域。

【例1-1】 在计算机内表示一个多项式

$$P(x) = \sum_{i=1}^{n} a_i x^i$$

由于多项式的项数可能随着对多项式的处理而增减，因此给存储分配带来一定的困难。例如，用数组来表示多项式就不一定很合适，但用链接式线性表却比较合适。链接式线性表就是线性表的一种具体表现形式，在用于表示多项式时，表中的每个结点表示多项式中的一个非零项 $a_i x^i$ $(a_i \neq 0)$。例如，多项式

$$P(x) = x^4 + 6x^3 + 8x + 3$$

可表示成如图1-1所示的形式。

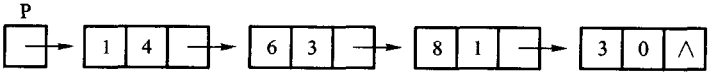


图1-1 用链接式线性表表示多项式

其中，P 是指针变量；符号 ∧ 表示它所在的结点是线性表的最后一个结点。每个结点有3个域，分别表示 x 对应项的系数、指数和指向下一个结点的“链”。结点的地址是组成该结点的存储单元的首地址，这个地址也叫做指针或者链。结点对应于C语言中的结构体类型。假设多项式所有项的系数都是整数，则用C语言书写的类型说明为：

```
struct Polynode{
    int coef;
    int exp;
    Polynode * link;
};
Polynode * polypointer;
```

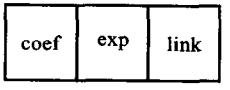


图1-2 结点 Polynode 的格式

结点 Polynode 的格式可画成如图1-2所示的形式。

【例1-2】 建立人事档案是应用数据结构的一个典型例子。假设要建立一个大学的教师和学生的档案，这种结构逻辑上可画成如图1-3所示的形式，这就是一种树形结构。结构中明显地反映出层次关系或从属关系。如果采用这种结构来建立人事档案需要解决它在计算机内的物理表示方法以及档案的查找问题。而且，许多数据结构不是静止不变的，这就是数据结构的动态特性，这种动态特性正是客观事物发展变化的反映。例如，当要增加一个系时，如何插入；一个专业被取消后，如何删除。因此，在这种结构上要定义查找、插入、删除等操作，并保证在插入和删除之后不破坏原来的结构类型。

由例1-2可以看出，数据结构主要研究数据对象的结构形式、各种数据结构的性质及其在

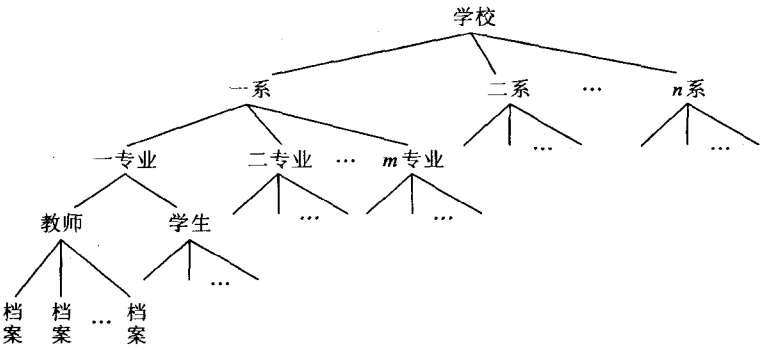


图 1-3 学校人事档案的树形结构

计算机内的表示方法、每种数据结构上定义的基本操作和算法，也研究算法的效率、数据结构的应用和数据排序以及查找等方面的技术。

1.2 数据结构发展概况

“数据结构”作为一门独立的课程是从 1968 年开始的。在此之前，其内容出现在不同的其他课程中，如“表处理语言”等。1959—1960 年由 J. McCarthy 设计的 LISP 系统（表处理语言）和 1962 年由 D. Farber 等人设计的 SONBOL（串处理语言）系统，将数据对象的结构形式表示成表结构或树结构。这些语言是以数据为中心，为处理非数值问题而设计的。而 FORTRAN、ALGOL 等算法语言则是为解决数值问题而设计的，它们都是以程序为中心的。以程序为中心的观点侧重于建立程序，而程序则在简单数据结构上进行复杂的运算，这种观点适合于数值计算问题。另一个观点把数据结构作为问题的中心（如数据库），而程序则围绕数据结构进行加工，如对数据进行查询和修改。这种观点适合于解决诸如航空订票系统等非数值问题，此类系统要求采用复杂的数据结构来描述系统的状态。可以说，程序设计以数据为中心的观点对数据结构的发展起到了推动的作用。

1968 年，在美国一些大学计算机系的教学计划中，曾把“数据结构”规定为一门课程，但对课程的内容并没有做明确的规定。当时的数据结构几乎与图论，特别是与树和表的理论是同义语；后来，扩充到包括网络、集合代数论、格、关系等方面，变成了现在称为“离散结构”的那些内容。但是，由于数据必须在计算机中进行处理，因此不仅要考虑数据的数学性质，而且还必须考虑数据的存储结构，这就要求进一步扩大数据结构的内容。随着数据库系统的不断发展，数据结构课程中又增加了文件管理的内容。近年来，算法与数据结构之间的关系日益明显，将算法设计与分析的基本思想引入数据结构课程中也成为一种趋势。

数据结构在计算机科学中是一门专业基础课和核心课程，它是设计和实现编译程序、操作系统、数据库系统和其他系统程序的重要基础。

1.3 抽象数据型

1.3.1 抽象数据型的定义

抽象数据型(Abstract Data Types,ADT)是程序设计语言中数据类型概念的推广。当在分析复杂的情况或处理复杂的事物时,经常使用抽象的思维方法,即舍去复杂系统中非本质的细节,只把其中某些本质的、能够反映系统重要宏观特性的东西提炼出来,构成系统的模型,并深入研究这些模型。这种抽象思维方法同样适用于软件系统的设计和研究。可以认为,一个复杂的软件系统是由一些数据结构、操作过程和控制机能所组成的。因此,在软件设计中,可以对3种不同的对象进行抽象,即过程抽象、控制抽象和数据抽象。抽象数据型就是对数据类型进一步抽象所形成的概念,对程序设计方法学和程序设计语言等都产生了深刻的影响,因此引起了人们的重视。下面通过一个例子来解释这个概念的含义,并引入概念的定义。

【例1-3】 设有一个线性表L。任意线性表的类型(或简称为"型")记为LIST。暂时不管L中的元素是什么,但可以把元素的"型"记为Elementtype。现在要求编写一个函数Purge( ),它以L为输入参数,执行结果是删除L中所有重复出现的元素。

函数Purge( )如下:

```
void Purge(LIST &L)
/* 类型 LIST 有待具体说明 */
{
  position p,q;    /* p 和 q 的类型是"位置" */
                   /* p 指向 L 中的"当前位置";q 向前搜索以寻找与 p 相同的元素 */
(1) p = First(L);
(2) while(p! = End(L)) {
(3)   q = Next(p,L);
(4)     while(q! = End(L))
(5)       if(Same(Retrieve(p,L),Retrieve(q,L)))
(6)         Delete(q,L);
          else
(7)         q = Next(q,L);
(8)   p = Next(p,L);
  }
}
```

函数Purge( )中各语句所起的作用分别是:First(L)求L的第一位置;Next(p,L)求L中位置p的下一个位置;End(L)用做线性表的结尾标志。每当执行语句(3)、(7)或(8)之前,都要测

试 p 或 q 是否等于 End(L),不等才能执行相应的循环;否则就会有 p = Next(End(L),L)或 q = Next(End(L),L),它们都超出了线性表表尾,因而没有定义。函数 Retrieve(p,L)返回线性表 L 中处于位置 p 的元素。Same(x,y)定义为:若 x 与 y"相同",则 Same 的值为 TRUE,否则为 FALSE;其中,x 和 y 都是类型为 Elementtype 的元素。Delete(q,L)表示删除 L 中位于 q 所指位置的元素。

这样编写的程序很容易读,整个程序的工作过程是这样的:变量 p 由线性表 L 的第一个位置开始向后扫描。随着程序的执行,在位置 p 前边的任何位置上,凡是与处于位置 p 的元素相同的元素均已从 L 中删除。在循环(2)~(8)的每一次执行中,q 用于扫描跟在位置 p 后面的各元素,以删除与处于位置 p 的元素相同的元素。然后,p 移向下一个位置,并重复执行上述过程,直至 p 等于 End(L)为止。

在例 1-3 的程序中,线性表 L 就是一种抽象的数据,这里没有涉及 L 的具体表示方法以及 L 有多长等具体性质;变量 p 和 q 也是抽象的,这里没有涉及位置是怎样表示的。可见,函数 Purge( )不依赖于线性表的具体实现方式,无论采用什么样的数据结构来表示线性表,都不影响 Purge( )的程序编码;Purge( )与线性表产生联系的唯一途径是调用线性表上的 First、Next、Delete 等操作。

函数 Purge( )的一般性与数据、函数的抽象性是密切相关的。例如,函数 Same( )的抽象性就值得注意。Same(x,y)用于判断 x 与 y 是否相同,若 x 与 y"相同",则 Same 的值为 TRUE,否则为 FALSE;其中 x 和 y 都是类型为 Elementtype 的元素。这里"相同"的概念是抽象的,否则就会损坏 Purge 的一般性而使其适用范围变窄。这是因为,若 Elementtype 是实型,则当且仅当 x = y 时,Same(x,y)的值为 TRUE;但是,若 Elementtype 是复合数据类型,例如是一个具有 3 个域的结构:

```
struct Elementtype {
  int acctno;
  char name[20];
  char address[50];
};
```

则当且仅当 x. acctno = y. acctno 时,Same(x,y)的值为 TRUE。如果将函数 Purge( )中的条件语句写成

```
(5) if(Retrieve(p,L) = = Retrieve(q,L))
(6)    Delete(q,L);
       else
(7)    q = Next(q,L);
```

则 Purge 就不适用于 Elementtype 结构体的情形,因为这时检查出 Retrieve(p,L)和 Retrieve(q,L)是两个结构体类型。

在例 1-3 中,如果把线性表和线性表上定义的 First、Next、Delete 等操作一起看成是一个整

体,就是下面要定义的抽象数据型概念的一个例子。

抽象数据型是一个数学模型和在该模型上定义的操作集合的总称。抽象数据型的例子很多,高级程序设计语言中的整型、实型和数组等都可以看成是抽象数据型。例如,整型一般都定义四则运算、乘方、赋值、比较和输入、输出等操作,这些操作成为程序中对整型数进行加工处理的手段。使用语言的用户只要知道这些操作的用途就可以编程序;至于这些操作是怎样实现的以及整型数在内存中是如何表示的,并不影响用户所编程序的编码形式。

在例 1 -3 中编制的程序有待于相关的抽象数据型的实现才能成为可运行的程序。在此例中,如果进一步提供 L 和 position 的适当说明,并用函数实现线性表上的各种操作,则 Purge( )就可以成为一个可运行的程序。抽象数据型的实现,就是将 ADT 转化成现有程序设计语言的语句,加上对应于该 ADT 中每个操作的函数。换言之,抽象数据型的实现是用现有程序设计语言能够支持的、适当的数据结构来表示 ADT 中的数学模型,并用一组函数来实现该模型上定义的各种操作。而数据结构则利用该语言的基本数据类型和复合数据类型的构造机制来构成。例如,数组和结构体就是 C 语言中的两种主要的、复合数据类型的构造机制。

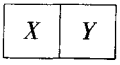
在 ADT 的定义中还有一点值得注意。根据这个定义,如果在相同的数学模型上定义两个不同的操作集,则认为其代表两个不同的抽象数据型。因此,相同的数学模型以及在此数学模型上定义的相同操作,有可能在不同的抽象数据型中出现。

1.3.2 数据类型、数据结构和抽象数据型

尽管“数据类型”、“数据结构”和“抽象数据型”等术语看起来很类似,但事实上它们具有不同的意义。在程序语言中,一个变量的数据类型是该变量值的类型。例如,布尔型变量的取值只能是“真”或“假”。基本的数据类型随各程序语言而异,在 C 语言中,有整型、浮点型和字符型等;利用基本数据类型构造复合数据类型的规则同样随各程序语言而异。下面将介绍在 C 语言中如何构造复合数据类型。

抽象数据型是一个数学模型及在该模型上定义的操作集的总称,而数据结构则是抽象数据型中数学模型的表示。按照抽象数据型的观点组织和设计算法,当用给定语言实现算法时,必须首先根据语言自身提供的抽象数据型和运算符找到表示抽象数据型的具体方式。当使用数据结构来表示作为抽象数据型基础的数学模型时,认为结点是数据结构的基本构件;而一个结点是由若干相同或不同类型的常数或变量,按照一定的方式进行组合所形成的数据集合。给相关结点集合起一个名字,并且把结点中的某个值解释为结点之间的链接信息(例如看做指向下一个结点的指针),这样就建立了一个数据结构。

例如,下面的结点



由两个变量 X 和 Y 组成:X 是一个整型变量;Y 是指向下一个结点的指针,若没有下一个结点,则令其为 NULL。把 n 个这样的结点链接起来(见图 1 -4),就建立了一个数据结构,其中 F 即可看

成是该数据结构的名字，也是指向首结点的指针。

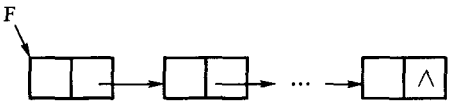


图 1-4 建立数据结构

在 C 语言和其他程序语言中，最简单的信息聚集机制是数组。一个数组是结点的一个有序序列，其中每个结点的数据类型都相同。数组中的每个结点是相继存放的，因此只需知道数组名（意味着知道数组的首地址）和结点在数组中的相对位置（即下标值），即可访问数组中的每个结点。结点中的值可以是任何一种基本型或复合型的数据。

在 C 语言中，把信息聚集在一起的另外一种常见的机制是结构体类型。一个结构体类型由若干个域组成，每个结构体类型的变量对应一个结点。与数组不同，结构体类型中每个域的类型可以互不相同，并且它们是通过名字而不是通过下标访问的。若干个同一类型的结构体变量可以组成一个数组，由结构体类型中各个域所定义的数据类型（复合型）就成为该数组中的“结点型”。例如，C 语言说明语句表明 reclist 是一个具有 4 个元素的数组，每个元素（结点）是一个具有两个域 data 和 next 的结构

```
struct record{
    float data;
    int next;
}reclist[4];
```

第三种聚集信息的方法是文件。文件同（一维）数组一样，由一组按一定顺序排列的具有相同类型的数据元素所组成。但是，文件中没有下标，各元素按名访问或按它们在文件中出现的顺序访问。利用文件聚集信息的一个优点是文件中的元素个数可以随时变化并且不受限制，这是数组所不具备的。

此外，在程序设计语言中，类型说明和变量说明是有区别的。例如，在类型说明中

```
struct date{
    int dayofmonth;
    int month;
    int year;
};
```

只规定了类型为 date 的变量的格式和取值范围。这时，如果没有进一步做变量说明，则实际上还不存在类型为 date 的变量；如果写成了

```
date date1,date2;
```

则说明 date1 和 date2 是类型为 date 的变量的两个实例。这两个实例的格式和取值范围由 date