高等学校计算机专业教材

GAODENG XUEXIAO JISUANJI ZHUANYE JIAOCAI

数据结构与算法

SHUJU JIEGOU YU SUANFA

王若梅 罗笑南 编著

中山大学出版社

数据结构与算法

王若梅　罗笑南　编著

中山大学出版社

·广州·

图书在版编目(CIP)数据

数据结构与算法/王若梅,罗笑南编著.—广州:中山大学出版社,2000.11
ISBN 7-306-01708-X

Ⅰ.数…
Ⅱ.①王…　②罗…
Ⅲ.①数据结构　②电子计算机-算法理论
Ⅳ.TP311.12

中国版本图书馆 CIP 数据核字(2000)第 71089 号

中山大学出版社出版发行
(地址:广州市新港西路 135 号　邮编:510275
电话:020-84111998、84037215)
广东新华发行集团股份有限公司经销
广东省番禺市市桥印刷厂印刷
(地址:广东番禺市市桥环城西路 201 号　邮编:511400　电话:020-84881937)
787 毫米×1092 毫米　16 开本　18.5 印张　462 千字
2000 年 11 月第 1 版　　2000 年 11 月第 1 次印刷
定价:26.00 元

内容简介

本书根据计算机发展的主流，依据计算机软件系列课程的要求，详细地介绍了在求解问题中，如何进行问题的分析、设计和实现，如何进行抽象数据类型的设计；讨论了线性数据结构（如线性表、栈、队和串），非线性数据结构（如树和图），集合类型等基本的数据结构，以及在软件设计中常用到的排序技术、外部数据的组织和处理技术；并简单介绍了有关算法设计的基本技术、并行处理技术和NP－完全性问题。

本书是计算机专业本科教材，也可作为计算机应用工程技术人员的参考用书。

前　言

现代社会是高科技的社会，是计算机技术广泛应用的社会。如今，几乎每时每刻都会产生计算机的最新技术和应用，人们已经不可以忽视计算机的重大作用了。

作为计算机的核心技术——计算机软件，从方法上，也从传统的面向过程的控制流式的方法，发展成今天的面向对象的软件技术。为了适应飞速发展的计算机技术，人们越来越多地研究软件开发过程中，如何为求解问题而进行的分析、设计和实现这几个重要环节的工作，研究软件设计中的工程化过程和算法设计的技术，以及影响软件性能的主要因素，以此来保证软件的正确、软件的重用，提高软件的质量和缩短软件的开发周期。

数据结构与算法这门课主要研究：如何在用计算机求解问题的软件编制过程中进行数据抽象，以及如何权衡利弊，选择适当的方法在计算机中表示和处理这些对象，保证正确、高效、易读和易维护。

本书根据计算机发展的主流，依据计算机软件的系列课程的要求，系统详细地介绍了在求解问题中，如何进行问题的分析、设计和实现，如何进行抽象数据类型的设计；讨论了线性数据结构（如线性表、栈、队和串），非线性数据结构（如树和图），集合类型等基本的数据结构，以及在软件设计中常用到的排序技术、外部数据的组织和处理技术；并简单介绍了有关算法设计的基本技术、并行处理技术和 NP－完全性问题。

全书共分十章。第一章介绍求解问题的基本方法和关于抽象数据类型、数据结构等基本概念；第二至第五章介绍基本的数据结构和有关的抽象数据类型；第六和第七章介绍内外排序的技术和外部数据的组织；第八章介绍算法设计的基本技术；第九和第十章介绍并行处理技术和 NP－完全性问题。全书采用面向对象的思想进行问题的分析和设计，强调理论抽象，强调设计和分析，并通过丰富的例题进行问题的求解说明，在文字上力求语言简练，通俗易懂。

本书是作为计算机专业本科学生的教材编写的，讲授学时为 60～90 学时，除进行课堂讲授外，还布置有课后作业和上机练习。本书也可作为计算机应用工程技术人员的参考用书。

对本书的编写，中山大学计算机科学系朱秉寰教授给予了很大的帮助，在许多方面得到了他的指教。朱老师是一位学识渊博、治学态度严谨的专家、学者，长期从事数据结构和算法方面的教学科研，他的一些独到的见解，使我们受益匪浅，我们深深表示感谢。中山大学计算机科学系的领导和各位老师也都给予了很多关心和支持，在此一并表示感谢。

本书虽经多次教学实践，难免还会存在一些错误和不足，敬希各位读者给予批评指正。

编著者

2000 年 8 月 25 日

目　录

第一章 引 论

计算机解题的程序，实际上是针对问题中的数据在计算机上的特定表示方式和结构的基础上，对解题的抽象算法的一个具体表述。不了解施加于数据上的算法，就无法确定如何构造数据；反过来，算法的结构和选择又依赖于算法的基础：数据结构。因此，任何人想编制计算机解题程序，都应当对数据结构这一学科有所了解。

这一章共分五节。第一节介绍算法、类型、抽象数据类型等一些基本概念以及本书所用的背景程序设计语言 C++的一些特征。第二节讨论处理大量数据的问题中，数据之间的基本的逻辑结构和计算机上表达这些结构的基本方法。第三节讲述开发大型数据处理软件的基本方法，以及算法和数据结构设计在这个过程中的作用。第四节介绍衡量算法或数据结构好坏的一些标准。第五节讨论递归。

第一节 抽象数据类型、算法和 C++

一、抽象数据类型

所有高级程序设计语言都会提供一些基本数据类型，供程序员描述在解题时遇到的运算对象和结果。例如，PASCAL 语言中的 integer，real，char，boolean，在 C++中的 int，long，unsigned，float，double，char，等等。属于某一个数据类型的数据，其值都必定落在某一个指定的范围之内。例如，C++中的 unsigned int 和 unsigned long 的数据，其值就分别落在 0 至 $2^{16}-1$ 和 0 至 $2^{32}-1$ 之间。通常给出定义：类型是数据的取值范围。

程序设计语言中的基本数据类型，除了提供数据的类型（即数据的取值范围）之外，还提供了属于该类型的数据所容许的一组操作。例如，设 i，j 是 C++中类型为 int 的变量，则诸如 i+j，i-j，i*j，i/j，i%j，i++，i--，i=j，i<j 等等都是合法的操作。因此有定义如下：

数据类型是一个类型及属于该类型的数据上的一组操作

只要按照程序设计语言的规定编写程序，通过编译程序翻译、链接程序连接，所得到的目标程序就可以通过系统的支持而运行起来，执行像 int，float，char 等类型中的数据上的各种合法操作。

但并不总是简单地用程序设计语言提供的基本数据类型，就能描述待解问题中的数据，以及完成在这些数据上的运算。要得到解题程序，往往需要建立新的数据类型。

例如，密码学中常常涉及大整数的运算。人们常用上百位数字的质数来加密和解密。对于这种上百位数字的整数，也涉及加、减、乘、除、求模、赋值、比较等操作。只要把

这种数据的类型和属于该类型的数据上的操作作出明确说明，我们也可以得到一个数据类型。不过这个数据类型并非程序设计语言直接支持的，在所编制的程序代码中不能直接引用它。只有在使用程序设计语言时把这种数据类型表达出来，并经过编译之后，才能像程序设计语言中的基本数据类型一样引用这种新类型的数据以及对这些数据进行操作。

我们把这种不是程序设计语言直接支持，但为求解问题需要而定义的数据类型叫做抽象数据类型，简记为 ADT（Abstract Data Type）。把使用程序设计语言来表达 ADT，使它能在编写程序时加以引用，叫做 ADT 的实现。

二、ADT 的描述

我们约定用三个部分来定义一个 ADT，即 ADT 的头部、ADT 的数据说明和 ADT 的操作说明。

ADT 的头部的表示形式为：

ADT〈ADT 的名字〉

ADT 的数据说明由一个数据说明表组成，每一表目的说明如下：

DATA

〈类型名〉〈数据表〉

数据表由一些数据名组成，每一个数据名都是这个 ADT 的一个数据分量，数据名之间用逗号分开。以后，若说明某个对象是属于该 ADT 后，则以对象名后继句点再后继数据名来引用对应的分量。

数据表最后还需要说明数据之间的逻辑关系。

ADT 的操作说明由操作说明表组成。每个表目用如下的方法说明：

OPERATION

〈操作名〉(〈参数表〉)〈情况说明〉

参数表列出该操作涉及的参数，参数表中的参数之间用逗号分隔。情况说明可由四个部分组成：

（1）输入：说明输入数据的类型和名称。

（2）输出：说明哪些变量是操作结果输出的载体。

（3）前件：说明执行该操作前所应满足的条件。如果这些条件不被满足，则不保证输出结果正确。

（4）后件：说明操作后的情况。就像 C++一样，我们用双斜杠表示注解，以双斜杠开始的文字至行尾是注解。

说明属于某 ADT 的对象，通过对象名后继句点再后继操作名和实际参数表来引用该 ADT 对应的操作。

ADT 的描述规范为：

```
ADT ADT 的名字
  DATA
    ADT 的数据说明
  OPERATION
    ADT 的操作说明
END ADT
```

现在，用这些约定对大整数抽象数据类型定义如下：

```
类型名：biginttype          取值范围：0 至 2^512 - 1 的整数
ADT bigint                  //抽象数据类型名叫 bigint
  DATA
    biginttype n;           //该对象只有一个分量名叫 n 是 biginttype 类型的
  OPERATION
    Addone;                 //递增 1
      前件：n+1<2^512
      后件：n=n+1

    Subone;                 //递减 1
      前件：n>0
      后件：n=n-1

    Mult (x, y);            //乘法
      输入：x 是 bigint 的对象
      输出：y 是 bigint 的对象
      前件：n * x.n<2512
      后件：y.n=n * x.n

    Mod (x, y);             //求余数
      输入：x 是 bigint 的对象
      输出：y 是 bigint 的对象
      前件：x.n≠00
      后件：y.n 是 n/x.n 的余数

    Div (x, y);             //整除
      输入：x 是 bigint 的对象
      输出：y 是 bigint 的对象
      前件：x.n≠0
      后件：y.n=n/x.n 的整数部分

    Compare (x);            //比较
      输入：x 是 bigint 的对象
      后件：若 x.n<n 则返回 2，若 x.n=n 则返回 1，否则返回 0

    Setup (x);              //赋初值
      输入：x 是 bigintttype 类型的整数
      后件：n 被赋成 x
```

END ADT //ADT bigint 定义完毕

三、算法

除了要考虑程序在运行中的数据特征之外，编写解题程序时还必须考虑解题算法。即必须考虑在这些数据上进行什么样的操作才能得到答案。

算法，是指有输入和输出的一个指令序列。该指令序列对输入数据进行加工，一步一步地操作，最终得到输出数据作为答案。指令序列中每条指令要满足三个性质：可行性，即原则上可用纸和笔来完成；确定性，即明白无误，只有一种解释；有穷性，即可以在有限时间内完成。对算法的任何输入，整个指令序列也必须能在有限时间内完成。

一个计算机解题程序通常就是一个算法。其中的输入是对该解题程序的输入数据，输出是解题程序终结时所得到的计算结果，指令就是计算机指令系统中的指令。但不是任何计算机程序都是算法。例如，操作系统也是一个计算机指令序列，也是一个程序，但它不是算法，因为只要把它运行起来，它就总是永不停机地运行下去。也不是所有的算法都必须用计算机的指令系统中的指令或程序设计语言的语句来书写，只要算法中的每条指令满足上述的三个性质就可以了。例如，尽管我们还未弄清楚 biginttype 类型上的数据的加减乘除等操作的实现细节，但我们都明白这些操作是可行、确定和有穷的。我们可以用这些操作来书写一个算法，判定给定类型为 biginttype 的整数 p 是否为质数。描述如下：

算法1.1：判定已知类型为 biginttype 的整数 p 是否为质数。

输入：类型为 biginttype 的整数 p，p>2。

输出：若 p 为质数则返回 yes，否则返回 no。

方法：设 q 是 biginttype 型整数，t 是 int 类型的，执行下述步骤：

1）令 q=2 和令 t=1；

2）当（p>q）且（t=1）时重复执行 2.1 行操作

2.1）若（p/q 的余数）=0 则令 t=0 否则令 q=q+1；

3）如果 t=0 则返回 no 否则返回 yes

4）停机

在上面的指令序列中，第 1 行的两条指令 q 被赋值 2 和 t 被赋值 1，显然满足算法的三个性质。第 2 行的指令，相当于在执行第 2.1 行语句之前先测试（p<q）和（t=1）是否同时成立（这种测试显然也是满足三个性质的），如果成立则执行 2.1 行的指令，否则执行第 3 行指令。第 2.1 行语句是按照 p mod q 是否为 0 分别执行给 t 赋值 0 和令 q 增值 1，无论测试（p mod q=0）是否成立还是给 t 和 q 赋值，也都满足三个性质。因此上述每条指令都满足三个性质。

由于语句 2.1 要么令 t=0，要么令 q 增值 1，而令 t=0 之后，再在第 2 行作测试时就不执行第 2.1 行的指令而执行第 3 行的指令，以及由于第 1 行给 q 的初值为 2，所以，语句 2.1 至多可给 q 作 p-2 次增值 1 就可令 q=p，这时，第 2 行语句中（p<q）且（t=1）的条件就不再成立，从而不执行 2.1 行指令改执行第 3 行指令了。因此，整个指令序列都可在有限时间内完成（实际费时和 p 的大小有关，p 越大，费时越多）。

又因为 t=0 当且仅当有 q≥2，且 q 整除 p，即 p 是合数，因此算法 1.1 正确地判定 p

(p>2) 是否为质数。

四、算法的描述

不同的程序设计语言有不同的描述 ADT 和算法的约定。本书采用类 C++语言来描述算法和 ADT 的实现。

(1) 可以预定义常量和类型:

定义常量:

```
#define MAX 100
#define TRUE 1
#define FALSE 0
```

定义类型:

```
typedef int state
```

表示类型 state 是一个整型。

(2) 把所有的算法都写成函数,用函数头、参数说明和函数体三部分来说明一个函数。函数头一般由四个部分组成:

```
〈存储类型〉〈类型名〉〈函数名〉(〈参数表〉)
{函数体
}
```

不管函数是否有参数,圆括号是语法规定必需的。如果函数返回值,则其值的类型由类型名指定。如果不返回值,则类型名为 void。名叫 main 的函数是整个算法的入口点,不设类型名。存储类型有 static, auto, register, extern 多种,主要作用是确定变量的(或函数的)作用范围和存储方式。缺省约定为 auto。

参数说明部分要说明函数头的参数表中的每个参数的类型,它由一或多个如下的说明组成:

```
〈类型名〉〈参数表〉;
```

参数表列出各参数名,参数名之间以逗号分隔。当参数是引用调用时,参数名前加“&”。

函数体由一对花括号括住,花括号内说明函数用到的工作变量名字和所属类型,以及一组语句。

(3) C++中的语句有五大类:

```
函数调用语句:        函数名(参数表);
赋值语句:            变量名=表达式;
选择语句:            if(表达式)语句序列;
                     if(表达式)语句序列1;
                        else语句序列2;
开关语句:            switch(表达式)
                      {
                       case值1:语句序列1; break;
                         ......
                       case值n:语句序列n; break;
                       default:语句序列n+1;
```

```
                                    }
循环语句：                          for（赋初值表达式序列；条件；修改表达式序列）
                                        语句序列；
                                    while（表达式）
                                        语句序列；
                                    do {
                                        语句序列；
                                    } while（表达式）
```

（4）结束语句：

```
函数结束语句：                      return 表达式；
case 结束：                         break；
异常结束：                          exit（提示异常代码）
```

（5）输入输出语句：

```
输入语句：                          scanf（变量表）；
输出语句：                          printf（变量表）；
```

（6）注释语句：

```
//注释内容；
```

C++提供了 int，long，unsigned，float，double，char，short 等基本数据类型，还提供了指针、结构（struct）、联合（union）、枚举（enum）、数组等构造新类型的方法，以及逻辑运算符。特别地提供了描述 ADT 的方法：（class）类。

类由多个存放数据值的成员和加工数据的运算组成，这些运算也称为方法。类型为类的变量称为对象。类分为公共部分和私有部分。公共部分描述用户使用类的界面，它使用户不必了解对象的内部细节而使用对象。而私有部分由帮助实现数据抽象的数据和内部操作组成。类通过把数据和方法包装在一起并将它们视为整体来封装信息。类在结构上隐藏了应用细节并严格限制对其数据和操作的外部访问，我们将类的这种特性称为信息隐藏，它保护了数据的完整性。

类的说明方法如下：

```
class〈类名〉{
    〈数据说明表〉
〈操作声明表〉}
```

数据说明表中说明该类的运算对象所拥有的各种分量的类型和权限，分量的权限和类型说明如下：

```
〈权限〉:〈类型名〉〈分量名表〉
```

其中的权限有三个可能的选择：private，public 和 protected，缺省约定为 private。权限为 private，意味着这个分量（或操作）仅对这个类可用，在这个类之外的程序段不能引用；public 意味着可为该类之外的程序段引用；protected 意味着可由该类派生的子类引用，除此外不能引用。一个类可以有多个分量的权限和类型说明。

类中的操作说明表中的每个操作说明如下：

```
〈权限〉:〈类型名〉〈函数名〉(〈类型表〉);
```

其中的类型表列出了参加这一操作的参数所属的类型名。而权限说明则和数据说明中的定