本书受国家自然基金“社会网络中藏语话题情感分析和主题词分析”项目（No.61672553）
与国家科技支撑计划“少数民族网络舆情综合分析与云服务关键技术研究
及应用示范”项目（No.2014BAK10B03）资助

算法设计与优化

SUANFA SHEJI YU YOUHUA

邱莉榕 胥桂仙 翁彧◎编著

中央民族大学出版社
China Minzu University Press

本书受国家自然基金“社会[redacted]分析”项目（No.61672553）
与国家科技支撑计划“少数[redacted]建技术研究
及应用示范”项目（No.2014[redacted]

算法设计与优化

SUANFA SHEJI
YU YOUHUA

邱莉榕 胥桂仙 翁 彧◎编著

中央民族大学出版社
China Minzu University Press

图书在版编目（CIP）数据

算法设计与优化/邱莉榕，胥桂仙，翁彧编著. —北京：中央民族大学出版社，2016.12（2017.6重印）

ISBN 978-7-5660-1296-8

Ⅰ. ①算… Ⅱ. ①邱… ②胥… ③翁… Ⅲ. ①算法设计 Ⅳ. ①TP301.6

中国版本图书馆 CIP 数据核字（2016）第 285003 号

算法设计与优化

编　　著　邱莉榕　胥桂仙　翁　彧
责任编辑　满福玺
封面设计　汤建军
出 版 者　中央民族大学出版社
　　北京市海淀区中关村南大街 27 号　　邮编：100081
　　电话：68472815（发行部）　传真：68932751（发行部）
　　68932218（总编室）　　68932447（办公室）
发 行 者　全国各地新华书店
印 刷 者　北京盛华达印刷有限公司
开　　本　787×1092（毫米）　1/16　印张：17.25
字　　数　365 千字
版　　次　2016 年 12 月第 1 版　2017 年 6 月第 2 次印刷
书　　号　ISBN 978-7-5660-1296-8
定　　价　60.00 元

前　言

计算机科学技术发展迅速，已经深入各行各业，应用到各个领域。它有效地解决了各种应用中的数值计算问题，同时也有效地解决了文本处理、信息检索、数据库管理、语音识别、图像识别、人工智能、机器学习、网络信息挖掘等非数值的数据处理问题。随着互联网技术和信息技术的飞速发展，电子数据与日俱增，如何有效地进行编程和算法分析已成为人们研究的热点问题。算法分析与设计是研究各种数据结构的逻辑存储、物理存储及各种算法的时间复杂度、空间复杂度分析的课程，有助于程序员更有效地组织数据、设计高效的算法、完成高质量的程序，以满足错综复杂的实际需要。

算法设计与优化是计算机学科的必修课程，具有理论性、抽象性、实践性。它涵盖了计算机学科中的数值分析、操作系统、编译原理等课程所涉及的相关算法。数据结构和算法分析与设计在计算机学科中的地位十分重要，是操作系统、软件工程、数据库概论、编译技术、计算机图形学、人机交互、数据挖掘等专业基础课和专业课程的先修课程。

本书包括算法设计与优化的基本概念、基本结构、基本算法三大部分，共 10 章。第一部分是线性表、栈、队列、树、图这些基本的数据结构，讲解这些数据结构的物理结构、逻辑结构及基本操作。第二部分是查找、排序、复杂算法介绍，分析各种算法的优劣。第三部分是分类算法、聚类算法介绍，讲解数据挖掘中的经典算法。本书各章节内容相对独立，以便针对不同专业和不同层次的需求组织授课内容，同时便于读者根据实际需要选择相关内容进行学习。本书还可供从事计算机应用工作的工程技术人员和编程爱好者参考。

本书采用类 C 语言描述各种算法，深入分析每种算法，内容全面，缜密严格，适合作为计算机相关专业本科生的数据结构课程、研究生算法分析课程和数据挖掘课程教材。数据结构课程可以使用本书第 1～7 章，算法分析与设计课程可使用第 1～7 章、第 10 章，数据挖掘课程可使用第 8～10 章。

本书受国家自然基金“社会网络中藏语话题情感分析和主题词分析”项目（No.61672553）与国家科技支撑计划“少数民族网络舆情综合分析

与云服务关键技术研究及应用示范”项目（No.2014BAK10B03）资助。本书在编写过程中，得到了杨国胜、翁彧等多位教师的大力支持，本书也得到了余佳、姚海申、李杰、王长志、张惠丽、齐琦、陈碧荣、徐亚等同学的帮助，特此感谢。

由于作者水平有限，书中的不妥之处在所难免，敬请读者批评指正。

编者

2016 年 11 月 20 日

目　录

第一章　绪论

目前，计算机已深入到社会生活的各个领域，其应用已不再仅仅局限于科学计算，而更多的是用于控制、管理及数据处理等非数值计算领域。计算机科学是一门研究用计算机进行信息表示和处理的科学。这里面涉及两个问题，即信息的表示和信息的处理。

信息的表示和组织又直接关系到处理信息程序的效率。应用问题的不断复杂化，导致信息量剧增与信息范围的拓宽，使许多系统程序和应用程序的规模很大，结构又相当复杂。因此，必须分析待处理问题中对象的特征及各对象之间存在的关系，这就是高级算法设计这门课所要研究的问题。

编写解决实际问题程序的一般过程：如何用数据形式描述问题，即由问题抽象出一个适当的数学模型；问题所涉及的数据量大小及数据之间的关系；如何在计算机中存储数据及体现数据之间的关系？处理问题时需要对数据作何种运算？所编写的程序的性能是否良好?上面所列举的问题基本上也是由高级算法设计这门课程来回答的。

著名的计算机科学家 N.沃思提出了一个有名的公式：算法+数据结构=程序。由此可见，数据结构和算法是程序的两大要素，二者相辅相成，缺一不可。一种数据结构的优劣是在实现其各种运算的算法中体现的。对数据结构的分析实质上也就是对实现其多种运算的算法的分析。下面我们就分别来介绍数据结构与算法。

1.1 什么是数据结构

1.1.1 什么是数据结构

算法与数据结构是计算机科学中的一门综合性专业基础课，是介于数学、计算机硬件、计算机软件三者之间的一门核心课程，它不仅是一般程序设计的基础，而且是设计和实现编译程序、操作系统、数据库系统及其他系统程序和大型应用程序的重要基础，下面我们通过具体的例子来认识数据结构。

例：首先分析学籍档案类问题。设一个班级有 50 个学生，这个班级的学籍表如表 1-1 所示。

表 1-1　学籍表

序号	学号	姓名	性别	英语	数学	物理
01	20160301	李明	男	86	91	80
02	20160302	马琳	男	76	83	85
……	……	……	……	……	……	……
50	20160350	刘薇薇	女	88	93	90

我们可以把表 1-1 中每个学生的信息看成一个记录，表中的每个记录又由 7 个数据项组成。该学籍表由 50 个记录组成，记录之间是一种顺序关系。这种表通常称为线性表，数据之间的逻辑结构称为线性结构，其主要操作有检索、查找、插入或删除等。

又如，对于学院的行政机构，可以把该学院的名称看成树根，把下设的若干个系看成它的树枝，把每个系分出的若干专业方向看成树叶，这样就形成一个树形结构，如图 1-1 所示。

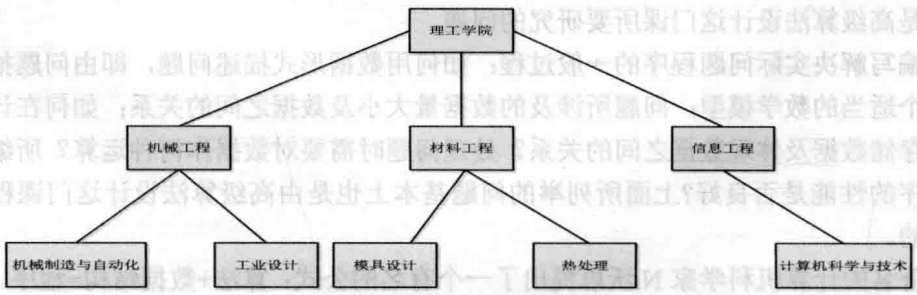


图 1-1　专业设置

树中的每个结点可以包含较多的信息，结点之间的关系不再是顺序的，而是分层、分叉的结构。树形结构的主要操作有遍历、查找、插入或删除等。最后分析交通问题。如果把若干个城镇看成若干个顶点，再把城镇之间的道路看成边，它们可以构成一个网状的图（如图 1-2 所示），这种关系称为图形结构或网状结构。在实际应用中，假设某地区有 5 个城镇，有一调查小组要对该地区每个城镇进行调查研究，并且每个城镇仅能调查一次，试问调查路线怎样设计才能以最高的效率完成此项工作？这是一个图论方面的问题。交通图的存储和管理确实不属于单纯的数值计算问题，而是一种非数值的信息处理问题。

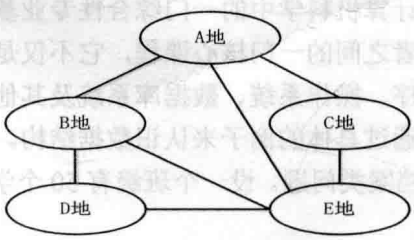


图 1-2　交通示意图

1.1.2 基本概念和术语

数据（Data）：是客观事物的符号表示。在计算机科学中指的是所有能输入到计算机中并被计算机程序处理的符号的总称。

数据元素（Data Element）：是数据的基本单位，在程序中通常作为一个整体来考虑和处理。

一个数据元素可由若干个数据项（Data Item）组成。数据项是数据的不可分割的最小单位。数据项是对客观事物某一方面特性的数据描述。

数据对象（Data Object）：是性质相同的数据元素的集合，是数据的一个子集。如字符集合 C={ ‘A’，‘B’，‘C’，……} 。

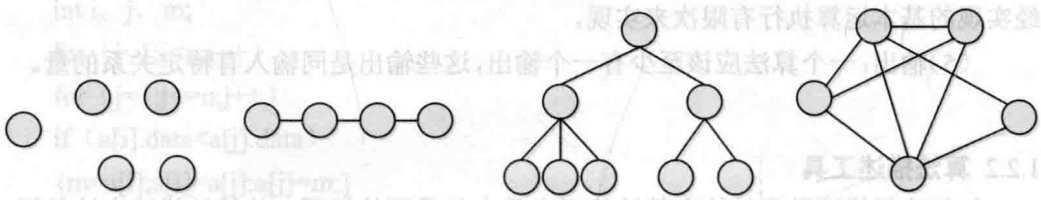
数据结构（Data Structure）：是指相互之间具有（存在）一定联系（关系）的数据元素的集合。元素之间的相互联系（关系）称为逻辑结构。数据元素之间的逻辑结构有四种基本类型，如图 1-3 所示。

（1）集合：结构中的数据元素除了“同属于一个集合”外，没有其他关系。

（2）线性结构：结构中的数据元素之间存在一对一的关系。

（3）树形结构：结构中的数据元素之间存在一对多的关系。

（4）图状结构或网状结构：结构中的数据元素之间存在多对多的关系。



（a）集合　（b）线性结构　（c）树形结构　（d）网状结构

图 1-3　四类基本结构图

1.1.3 数据结构的存储方式

数据结构在计算机内存中的存储包括数据元素的存储和元素之间的关系的表示。元素之间的关系在计算机中有两种不同的表示方法：顺序表示和非顺序表示。由此得出两种不同的存储结构：顺序存储结构和链式存储结构。

顺序存储结构：用数据元素在存储器中的相对位置来表示数据元素之间的逻辑结构（关系）。

链式存储结构：在每一个数据元素中增加一个存放另一个元素地址的指针（pointer ），用该指针来表示数据元素之间的逻辑结构（关系）。

例如有数据集合 A={3.0，2.3，5.0 ，-8.5 ，11.0} ，对应两种不同的存储结构。顺序结构：数据元素存放的地址是连续的；链式结构：数据元素存放的地址是否连续没有要求。

数据的逻辑结构和物理结构是密不可分的，一个算法的设计取决于所选定的逻辑结构，而算法的实现依赖于所采用的存储结构。

1.2 算法描述与分析

1.2.1 什么是算法

在解决实际问题时，当确定了数据的逻辑结构和存储结构之后，需进一步研究与之相关的一组操作（也称运算），主要有插入、删除、排序、查找等。为了实现某种操作（如查找），常常需要设计一种算法。算法（Algorithm）是对特定问题求解步骤的一种描述，是指令的有限序列。描述算法需要一种语言，可以是自然语言、数学语言或者是某种计算机语言。算法一般具有下列 5 个重要特性：

（1）输入：一个算法应该有零个、一个或多个输入。

（2）有穷性：一个算法必须在执行有穷步骤之后正常结束，而不能形成无穷循环。

（3）确定性：算法中的每一条指令必须有确切的含义，不能产生多义性。

（4）可行性：算法中的每一个指令必须是切实可执行的，即原则上可以通过已经实现的基本运算执行有限次来实现。

（5）输出：一个算法应该至少有一个输出，这些输出是同输入有特定关系的量。

1.2.2 算法描述工具

如何选择描述数据结构和算法的语言是十分重要的问题。传统的描述方法是用 PASCAL 语言。在 Windows 环境下涌现出一系列功能强大、面向对象的描述工具，如 Visual C++，Borland C++，Visual Basic，Visual FoxPro 等。近年来在计算机科学研究、系统开发、教学以及应用开发中，C 语言的使用越来越广泛。因此，本教材采用 C 语言进行算法描述。为了能够简明扼要地描述算法，突出算法的思路，而不拘泥于语言语法的细节，本书有以下约定：

（1）问题的规模尺寸用 MAXSIZE 表示，约定在宏定义中已经预先定义过，例如：#define MAXSIZE 100。

（2）数据元素的类型一般写成 ELEMTP，可以认为在宏定义中预先定义过，例如：#define ELEMTP int 在上机实验时根据需要，可临时用其他某个具体的类型标识符来代替。

（3）一个算法要以函数形式给出：类型标识符函数名（带类型说明的形参表）{语句组}例如：

```
int add （int a，int b）{int c;
c=a+b;
```

```
return（c）;
}
```

除了形参类型说明放在圆括号中之外，在描述算法的函数中其他变量的类型说明一般省略不写，这样使算法的处理过程更加突出明了。

（4）关于数据存储结构的类型定义以及全局变量的说明等均应在写算法之前进行说明。

下面的例子给出了书写算法的一般步骤。

例 1-1：有 n 个整数，将它们按由大到小的顺序排列，并且输出。

分析：n 个数据的逻辑结构是线性表（a1，a2，a3，……，an）；选用一维数组作存储结构。每个数组元素有两个域：一个是数据的序号域，一个是数据的值域。

```
struct node
{int num; /*序号域*/
int data; /*值域*/
}
/*算法描述*/
void simsort（struct node a [MAXSIZE]， int n）/*数组 a 的数据由主函数提供*/
{
int i，j，m;
for（i=1;i<n;i++）
for（j=1;j<=n;j++）
if（a[i].data<a[j].data）
{m=a[i];a[i]=a[j];a[j]=m;}
for（i=i;i<=n;i++）
printf（"%8d %8d %10d\n"，i，a[i].num，a[j].data）;
}
```

1.2.3 算法分析技术初步

评价一个算法应从四个方面进行：正确性、简单性、运行时间、占用空间。但主要看这个算法所要占用机器资源的多少。而在这些资源中时间和空间是两个最主要的方面，因此算法分析中最关心的也就是算法所需的时间代价和空间代价。

1.空间

算法的空间代价（或称空间复杂性），是指当问题的规模以某种单位由 1 增至 n 时，解决该问题的算法实现所占用的空间也以某种单位由 1 增至 f（n），并称该算法的空间代价是 f（n）。存储空间一般包括三个方面：指令常数变量所占用的存储空间；输入数据所占用的存储空间；辅助（存储）空间。一般地，算法的空间复杂度指的是辅助空间。一维数组 a[n]的空间复杂度为 O（n）；二维数组 a[n][m]的空间