

全国高等院校“十五”计算机规划教材
计算机科学与技术教材系列 (4)

数据结构教程

C 语言版

王庆瑞 编 著



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

全国高等院校“十五”计算机规划教材
计算机科学与技术教材系列（4）

数据结构教程

C语言版

王庆瑞 编 著



北京希望电子出版社
Beijing Hope Electronic Press
www.bhep.com.cn

内 容 简 介

本书详细介绍了基本数据结构、面向对象的程序设计和基本算法设计方法和算法理论。内容全面，讲解深入浅出，各章、节的重难点、主次内容都做了恰当合理的安排。

本书由8章构成，第1章概括性地介绍了算法和数据结构的概念，算法的描述方法，算法的评价标准和方法，以及算法设计的一般方法。第2、3、4章集中介绍了最基本的数据结构——表结构、树结构和图结构。第5章介绍了基本排序算法，包括内排序和外排序。第6章从常见的集合运算角度，介绍数据集合的组织形式、实现运算的算法以及算法效率。第7章介绍表、树、图等基本结构的类实现方法。第8章简单介绍了NP完全问题。

作者根据多年的教学经验，在整体结构安排、内容取舍以及整书的编写过程中，都充分考虑了教与学的特点，以及所面对的特定读者的具体需要。本书结构清晰，内容丰富，文字叙述简洁明了，可读性强，既便于教师课堂讲授，又便于自学者阅读。

本书可作为普通高校、职业学校、远程教学的计算机科学与技术专业本、专科学生的教材和教学参考用书，也是广大程序设计爱好者必备的理论学习指导书。

系 列 书 名：全国高等院校“十五”计算机规划教材
计算机科学与技术教材系列（4）

书 名：数据结构教程（C语言版）

文本著作者：王庆瑞 编著

责 任 编 辑：王玉玲

出版、发行者：北京希望电子出版社

地 址：北京市海淀区知春路63号卫星大厦三层 100080

网址：www.bhp.com.cn

E-mail：lxr@bhp.com.cn

电 话：010-62520290,62521724,62528991,62630301,62524940,62521921,82610344

（发行）010-82675588-202（门市） 010-82675588-501,82675588-201（编辑部）

经 销：各地新华书店、软件连锁店

排 版：希望图书输出中心 周玉

文本印刷者：北京广益印刷有限公司

开本 / 规格：787毫米×1092毫米 16开本 22.375印张 504千字

版次 / 印次：2002年8月第1版 2002年8月第1次印刷

印 数：0001-5000册

本 版 号：ISBN 7-900101-53 -5

定 价：26.00元

说明：凡我社产品如有残缺，可执相关凭证与本社调换。

计算机科学与技术教材系列

编委会成员名单

主任委员：徐洁磐

副主任委员：秦小麟 宋方敏

委员：（按姓氏笔画排序）

王元元 教授 解放军理工大学

宋方敏 教授 南京大学

张桂芸 副教授 天津师范大学

张新荣 教授 天津大学

柏家求 教授 天津大学

秦小麟 教授 南京航空航天大学

徐永森 教授 南京大学

徐洁磐 教授 南京大学

殷新春 副教授 扬州大学

蔡庆生 教授 中国科学技术大学

MAJ562/10

序

近年来计算机科学与技术的发展突飞猛进,其应用范围之广,对国民经济影响之大前所未有,特别是计算机网络、电子商务、多媒体技术等发展,正在彻底改变人类的工作方式与生活方式,同时也彻底改变了传统产业与传统的工作模式。目前,计算机科学与技术是高新技术的主要标志,是先进生产力的重要支柱,因此,发展计算机事业是摆在我们面前的重要任务。有鉴于此,我们特组织编辑了以大学本科学生为对象的《计算机科学与技术教材》丛书,为我国信息化培养人才作一份贡献。

当前,在我国计算机教学及教材建设中普遍存在着一些弊病与不足,主要有如下几种:

1. 在计算机教材特别是基础性教材中严重存在知识陈旧、落后,跟不上计算机科学与技术发展的步伐。由于计算机技术的飞速发展,内容更新要求极快,一般三五年就需作重大调整而一二年需作必要调整,而现有教材大都不能适应此种变化速度,这种现象在基础性教材中尤为突出。

2. 现有教材很多以国外教材为蓝本,存在着脱离我国具体应用实际的弊病。如何根据我国国情并参考国外先进技术编写教材是当务之急。

3. 现有的教材大都适应面窄,多数仅适应计算机本科专业而尤其仅适应少数重点院校与重要院校。而目前,由于计算机教育发展迅速,各种与计算机相近与相关的专业蓬勃发展,在计算机本科专业中也出现了不同层次的要求,特别是以应用型人才为主的培养要求。因此,迫切需要有一套适应面较广的基础性教材以满足多种层次的要求。

根据以上分析,本丛书编写原则是

1. 适应计算机科学与技术飞跃发展的需要,本教材丛书具有先进性与时代特性,并且每隔一二年作一次小的调整,每隔四五年重新修订出版。

2. 本教材丛书具有适应面广,基础性强的特点,能满足多种层次、多种类型的计算机专业本科学生的需要,特别是满足计算机应用型人才培养的需要。

3. 本教材丛书具有密切结合我国实际应用、反映我国计算机事业发展需求的基本特点,并且能在实际应用中发挥作用。

4. 教材是有别于一般书籍的一种特殊读物,它要求基本概念清楚,基本理论扎实,知识量大与实际应用联系紧密等特点,它还要求教材的内容逻辑性强,可读性好,深入浅出,并附有习题与参考资料等内容,本教材丛书将突出体现这些特点使其适合于教材需要。

5. 本教材丛书选题是根据我国目前实际并参考国际最新动态而制订的。本教材丛书第一批8本都是具有普遍性与基础性的教材,在不久我们将分别推出第二批、第三批教材满足不同的需要。

6. 本教材丛书聘请有深厚理论基础与应用实践经验且长期在教学、科研第一线工作的高校教授编写,特别是有专长的中、青年教授是本丛书教材编写的主要力量。

我们感谢丛书编委会各委员为本丛书出版所作的贡献，我们也感谢北京希望电子出版为丛书的立题、编审所作的努力，最后我们感谢丛书的作者为丛书编写、发行与发展所作的努力。

我们还期待广大读者为丛书提出宝贵意见与建议，我们将通过修订，努力把本丛书办得更好。

计算机科学与技术教材编委会

2002年3月于南京大学

前 言

算法设计、数据结构和程序设计语言构成了计算机程序设计的三大知识板块。算法设计的职责是针对给定问题,提出(设计出)求解方法,数据结构的任务是根据算法的要求,从问题中抽象出数据、数据集合,以及数据之间的关系,并把算法分解成对数据集上的运算,而语言则提供实现算法和数据集上运算的程序设计工具。从程序设计角度来说,算法是程序的核心,是灵魂。从知识层次角度来说,数据结构则是介于程序语言和算法之间,起着承上启下的作用。因此,数据结构往往既含有初级算法设计知识,也含有较高层的程序设计知识。正因为如此,随着计算机应用领域的不断扩大,软件需求量也不断地增大,程序设计知识普及率也随之增高,在这种情况下,数据结构课程,已经由原来计算机专业的特有课程,逐步面向大多数非计算机专业,甚至中等职业学校开设。

本书内容的选取,定位于以数据结构知识为主,同时加强了算法设计与分析方法的内容,这对那些在本科阶段不单独开设算法设计课程,而又需要掌握一部分算法设计与分析的基本原理的读者群是非常合适的。

当今,面向对象的程序设计方法在软件开发中已经占据主流地位,但是,如果直接用类和对象描述数据结构和算法,对初学者来说是很难接受的,更何况即使采用类结构描述算法,其成员函数的实现仍然离不开传统的面向过程的程序设计技术。考虑到上述情况,本书采用循序渐近的方式,先用传统的方法介绍数据结构的基本理论和算法设计方法,再过渡到类的定义及使用。在定义类结构时,多数算法都由前文相关程序移植而来。采用这种编排方式,可以使数据结构的学习难点和面向对象的程序设计难点分散开来,这就大大减轻了读者学习的难度,也增强了知识的系统性和连贯性。

本书从基本算法设计方法入手,系统、全面地介绍表、树、图、集合等基本数据结构,各种结构的特点,所能承担的运算,实现运算的算法设计,及其对算法的性能加以评价。

书中的算法除个别的只给出非形式化描述形式外,都用 C/C++ 语言描述,而且多数都能直接作为程序使用,但也有一些算法,为追求易读性,只是一种描述形式,不一定完全符合 C/C++ 的语法规则,上机运算时需要稍作改动。

全书共分成 8 章。第 1 章概括性地介绍了算法和数据结构的概念,算法的描述方法和评价标准、评价方法,并花费较大篇幅介绍算法设计的一般方法,包括递归、分治和平衡、动态规划、贪心法、搜索-回溯法等方法的基本原理,这些方法差不多贯穿全书。由于这部分内容稍有难度,使用者可以根据实际情况适当节选。

第 2、3、4 章集中介绍最基本的数据结构——表结构、树结构和图结构。在第 2 章表结构中,以基本的线性表为主,将栈、队、矩阵和字符串结构等也列在此章,因为这些“小”结构与线性表关系十分密切,有的是线性表的变种,有的可以转化成线性表处理。这 3 章内容是本书的基础部分,也是重点部分。

第5章介绍基本排序算法，包括内排序和外排序方法，这些精彩、经典的算法通常被认为是算法设计技术的入门知识。

第6章从常见的集合运算角度，介绍数据集合的组织形式、实现运算的算法以及算法效率。

第7章介绍表、树、图等基本结构的类实现方法。

第8章，简单介绍了NP完全问题。这部分内容，对于那些少有机会学习更深层算法设计理论的读者，了解算法设计的“禁区”是十分有用的。

本书各章都配有大量的练习题，其中一部分可选作上机练习题。大多数习题都紧扣正文中的教学内容，有些难度较大的习题（以标注“*”多少表示难易大小）往往是正文内容的引伸。

本书使用对象是计算机科学与技术专业的本科生，在略去部分带“*”的章节（难度较大或非基本内容）时，也可作为计算机科学与技术专业大专生和部分非计算机专业的本科生，或硕士生教材。为了拓宽读者面，便于自学，在编写过程中，力求语言简练、通俗易懂，内容广泛、由浅入深，并着力挑选实用性强的示例藉以阐明基本概念和基本算法。

本书在编写过程中，一直得到王元元教授的关注和支持，他对本书内容的编选提出了宝贵的意见，在此特向王元元教授表示深谢。还要感谢黄春华先生为本书打印部分稿件。

要特别感谢陈卫卫副教授，她认真审阅了此书，并根据她的实际教学经验，提出了很有价值的修改意见。

虽经作者再三努力，本书不当甚至错误之处在所难免，恳请读者指正，更请同行不吝赐教。

作者

2002年3月于南京

目 录

第1章 概述

1.1 数据结构的概念.....	1
1.1.1 问题的求解过程.....	1
1.1.2 数据和数据结构的意义.....	2
1.2 算法的描述和实现.....	3
1.3 算法的评价方法.....	6
1.3.1 评价标准.....	6
1.3.2 算法的时间复杂性.....	7
1.3.3 最坏情况和平均情况.....	9
1.3.4 计算时间复杂性的一般方法.....	10
1.3.5 算法的选用原则.....	12
1.4* 算法设计的一般方法.....	12
1.4.1 递归.....	12
1.4.2 分治和平衡.....	16
1.4.3 动态规划.....	19
1.4.4 贪心法.....	21
1.4.5 搜索-回溯法.....	23
习题一.....	25
第2章 表结构.....	27
2.1 表结构的概念.....	27
2.1.1 术语.....	27
2.1.2 存储方式.....	27
2.1.3 表结构的运算.....	31
2.2 顺序表的运算.....	32
2.2.1 插入和删除.....	32
2.2.2 查找.....	33
2.3 链表.....	38
2.3.1 链表的基本操作.....	38
2.3.2 链表的种类.....	43
2.3.3 链表的构造和遍历.....	44
2.3.4 链表的插入和删除.....	48
2.4 栈和队.....	51
2.4.1 概念.....	51
2.4.2 栈和队的插入删除.....	52
2.4.3 栈的应用.....	59
2.5 静态链表.....	62

2.5.1 静态链表的定义	62
2.5.2 自由队列	64
2.5.3 静态链表的插入删除操作	65
2.5.4 多表共享空间	66
2.5.5* 不带数据域的静态链表	67
2.6 矩阵运算	68
2.6.1 矩阵的存储	68
2.6.2 稀疏矩阵的转置	71
2.6.3 稀疏矩阵的乘法	74
2.6.4 稀疏矩阵的链式存储	77
2.7 字符串	78
2.7.1 字符串的运算和存储方法	78
2.7.2 简单的模式匹配算法	80
2.7.3* KMP 算法	82
2.7.4* 其他模式匹配算法	88
2.8* 表结构的其他存储形式	91
2.8.1 目录存储和索引目录存储	91
2.8.2 单链域的双向链表	94
习题二	96
第 3 章 树结构	103
3.1 树结构的概念	103
3.1.1 术语	103
3.1.2* 树的存储方式	107
3.2 二叉树	108
3.2.1 二叉树的概念和基本性质	108
3.2.2 特殊二叉树	110
3.2.3 二叉树的存储方式	113
3.2.4 树和森林与二叉树的相互转换	114
3.3 二叉树的遍历	116
3.3.1 遍历方法	116
3.3.2 递归的遍历算法	118
3.3.3 遍历算法的应用	121
3.3.4 遍历序列的性质	123
3.3.5* 非递归的遍历算法	125
3.4 二叉树的构造	128
3.4.1 树的唯一性	128
3.4.2 用先序序列和中序序列构造二叉树	130
3.4.3 用扩充先序序列构造二叉树	131
3.5 检索树	132
3.5.1 检索树的查找	132
3.5.2 检索树的插入和构造	134

3.5.3 检索树的删除	136
3.6 平衡树	140
3.6.1 平衡树的概念	140
3.6.2 平衡树的插入	141
3.6.3* 平衡树的删除	146
3.7* 红黑树	151
3.7.1 红黑树的概念	151
3.7.2 红黑树的旋转	154
3.7.3 红黑树的插入	155
3.7.4 红黑树的删除	159
3.8 哈夫曼树	163
3.8.1 编码和编码树	163
3.8.2 哈夫曼算法	166
3.9* 判定树	169
习题三	173
第4章 图结构	180
4.1 图的概念和存储结构	180
4.1.1 图的概念	180
4.1.2 图的存储结构	184
4.2 先深搜索和先广搜索	191
4.2.1 先深搜索	191
4.2.2 先深搜索的简单应用	195
4.2.3* 先广搜索	198
4.3* 无向连通图的双连通分量	199
4.3.1 关节点	199
4.3.2 找关节点的算法	201
4.4 最小生成树	203
4.4.1 Kruskal 算法	204
4.4.2 Prim 算法	208
4.5 最短路径	211
4.5.1 单源最短路径	211
4.5.2* 每对顶点之间的最短路径	214
4.6 有向无回路图	216
4.6.1 基本概念	216
4.6.2 拓扑排序	218
4.6.3* 关键路径	221
习题四	224
第5章 排序	227
5.1 基本概念	227
5.2 插入排序	228
5.2.1 直接插入排序	229

5.2.2 二分插入排序	231
5.2.3 希尔排序	232
5.3 交换排序	235
5.3.1 汽泡排序	235
5.3.2 快速排序	237
5.4 选择排序	242
5.4.1 树选排序	243
5.4.2 堆排序	244
5.5 合并排序	250
5.5.1 递归的合并排序	250
5.5.2 非递归的合并排序	252
5.6 基数排序	254
5.7 外部排序	259
5.7.1 多路合并	260
5.7.2 胜者树和败者树	262
5.7.3 替代选择算法	264
5.7.4* 初始顺串的生成	267
5.7.5* 最佳合并树	270
5.7.6* 磁带排序	271
习题五	274
第 6 章 集合运算	279
6.1 集合的基本运算	279
6.2 散列表	281
6.2.1 散列函数	281
6.2.2 散列表的构造	284
6.2.3* 散列表的性能分析	286
6.3* 最优检索树	288
6.4* 平衡树模式	292
6.4.1 2-3 树	292
6.4.2 B 树和 B+树	294
6.4.3 键树	297
6.5 不相交集合并	298
6.5.1 问题的背景	298
6.5.2 树结构的 union-find 算法	299
6.5.3* 表结构的 union-find 算法	302
习题六	304
第 7 章 类结构	306
7.1 表结构的类	306
7.1.1 栈类和队类	306
7.1.2 顺序表类	308
7.1.3 链表类	310

7.2 树结构的类.....	317
7.2.1 二叉树类.....	317
7.2.2 检索树类.....	319
7.2.3 平衡树类.....	322
7.3 图结构的类.....	325
7.3.1 图的存储.....	325
7.3.2 先深搜索.....	326
7.3.3 拓扑排序.....	328
习题七.....	328
第8章 NP 完全问题简介.....	330
8.1 问题的时间复杂性.....	330
8.1.1 算法的重要性.....	330
8.1.2 问题的固有难度.....	331
8.2 不确定性算法和 NP 问题.....	333
8.2.1 不确定性算法.....	333
8.2.2 P 问题类和 NP 问题类.....	334
8.3 NP 完全问题类.....	336
8.3.1 NP 完全问题的定义.....	336
8.3.2 问题的 NP 完全性证明.....	337
习题八.....	339
参考文献.....	340

第1章

概 述

1.1 数据结构的概念

“数据结构”属于程序设计方法类的一门重要课程，主要研究怎样合理地安排组织待处理问题所涉及的数据，设计处理数据的算法，提高程序运行效率。

随着计算机技术（尤其是软件技术）的发展，数据结构技术也不断地发展。在程序设计技术中，数据结构知识起着十分重要的作用。因此，所有计算机及相关专业都将“数据结构”课程列为重要的技术基础课。

1.1.1 问题的求解过程

当人们着手为给定问题设计处理该问题的计算机程序（或软件）时，往往先要对问题进行认真分析，设计出求解策略，即算法（algorithm）。这时的算法是程序的雏形，我们把这样的算法称为“问题级的算法”（或初级算法）。然后根据问题级算法的需要，从问题中抽象出所要处理的数据，设计出数据的组织方式（即建立数据结构），将问题级算法分解成对数据结构的运算，进而设计出对数据结构进行处理的算法（数据结构级算法）。接下来，要对算法的性能（可行性和运行效率等）进行评估（称为对算法进行分析）。若分析结果比较满意，则可以着手程序设计，并适当地选择一些数据，对程序进行调试，最后将程序交付使用。

算法的运行效率指执行算法（实际上是执行算法所对应的程序）时需要花费多少时间，占用多少内存空间。

若分析结果不能满意，则要逐级返工。若支持算法的数据结构不能满足要求，则要重新设计数据结构和处理数据结构的算法；若对初级算法不满意（因运行时间过长，或需要空间太多，不能满足客观要求），则要重新设计解题算法。反复如此，直到满意为止。

图 1.1 表示这种基本解题过程。

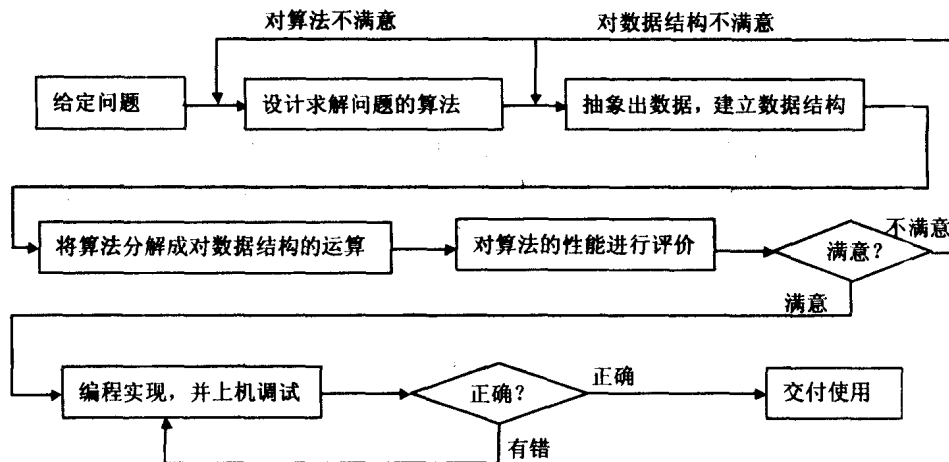


图 1.1 求解问题的大体步骤

因为程序是按算法设计思想进行设计的,所以程序中含有算法。就这个意义上说,算法就是程序(但算法不等于程序)。为叙述方便,本书中有时称程序为算法,有时也称算法为程序。

有了程序之后,问题的求解完全交由计算机去做,所以有时称程序(或算法)就是问题的解,即算法就是给定问题的解答(答案),称这种解是问题的算法解,或程序解,软件解等。

1.1.2 数据和数据结构的意义

数据是对客观事物的名称、数量、特征、性质的描述形式,是计算机所能处理的一切符号的总称。数据既是计算机加工的对象,又是计算机生产出来的产品(计算结果)。

在客观世界中,数据的表示形式是千差万别的,有数值型数据(整数、实数等),有文字型数据(字符,或字符串等),有声音、图像形式的数据,甚至程序代码也被认为是数据。不过在计算机中,数据总是以某种编码形式出现的。

“数据结构”中研究的数据并不考虑数据的实际含义,不研究那些单个的孤立的数据,也不大关心只涉及少量数据问题,着重研究由众多数据元素组成的数据集合,研究数据元素之间存在的内在联系(即关系),以及通常需要对数据和数据集合进行哪些运算,怎样才能提高运算效率等方面。

在数据集合中,一个数据元素(data element)又叫做一个数据结点,简称结点(node)。同一个集合中,各结点应当具有相同的数据类型,并且具有某种关系。

用来描述一个独立事物的名称、数量、特征、性质的一组相关信息组成一个数据结点。通常,一个结点含有多个数据项(即记录型结点,或结构型结点),每个数据项是结点的一个域(field),能够唯一标识结点的域叫关键字(key)。某些情况下,一个数据结点只含一个数据项(单值结点),为简化描述,可以把这样的结点看成一个整数。

例如,在设计处理学生成绩问题的程序时,每个学生有关的数据项(域)构成一个数据结点,包括学生的姓名、学号、各科考试成绩等等,学号可以作为结点的关键字。

又如,在处理商品问题时,一个数据结点对应一种商品的相关数据项,包括商品编号和名称、规格、数量、生产厂家、单价、入库日期等,商品编号可以作为关键字。

一个有穷的结点集合,以及该集合中各结点之间有穷的关系集合,便组成一个数据结构(data structure)。数学上可以表示成

$$B = (D, R)$$

其中D表示有穷结点集,R表示D上的关系集,B则表示一个数据结构。

至于一个结点内部各域之间的关系,不是“数据结构”研究的重点,因为它们之间的关系是一种相对固定的简单关系,仅在研究数据的存储表示(即或物理结构)时会涉及到这种关系。

有时,为描述方便,常用结点的关键字代表结点。

常见的数据结构有表结构、树结构和图结构,以及用于描述数据在外部存储介质上存储形式的文件结构等。

表结构用于描述结点之间某种简单的先后次序关系,比如学生成绩单。

树结构用于描述结点之间的层次关系,或嵌套关系,比如某部门的组织机构。

如果结点之间存在复杂的“多对多”关系，那么就用图结构进行描述，比如城市交通网。某些问题涉及到的结点之间存在一种特殊的“无关关系”，即离散关系。

图 1.2 形象地表示出上述几种基本数据结构。

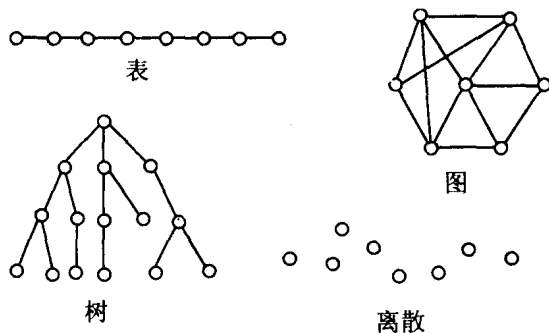


图 1.2 基本数据结构

数据的存储形式（存储表示）称为存储结构，或物理结构。相应的，上述的结构是逻辑结构。在存储数据的同时，还必须体现出结点之间的关系。

用于存储一个数据结点的存储单元叫做一个存储结点。因为一个数据结点对应一个存储结点，所以通常存储结点也简称为结点。预留的存储结点（即尚未存储数据的存储结点）叫空白结点，或称空结点、自由结点。

对数据和数据结构的处理操作称为对这个数据结构进行的运算（operation）。对不同的数据结构可能有不同的运算。比如，查找、插入、删除、排序等都是表结构的常见运算。

1.2 算法的描述和实现

算法记载了人们求解问题所采用的方法和步骤，是“程序的流程”。

计算机科学大师克努特（D.E.Knuth）在他的《计算机程序设计技巧，第一卷：基本算法》（见参考文献[2]）一书中给算法下了科学的定义：

一个算法就是一个有穷规则的集合，其中，规则规定了解决某一特定问题的运算序列，并且，算法还应具有以下 5 个重要特性。

（1）有穷性。一个算法必须在执行有穷步骤后终止。只具有其他特性，而不具有有穷性的只能称为“计算方法”。

（2）确定性。算法的每一步必须具有确切的定义。在任何情况下，对有待执行的动作必须严格规定，而不能含混不清。即使用自然语言描述的算法，也应当尽量准确，不带二义性或多义性。

（3）输入。一个算法可以有输入数据（执行前的初始量），也可以没有输入数据（要处理的对象是固定的，例如求解八皇后问题的算法就不要求输入数据）。

（4）输出。算法必须有输出数据（包括输出某种动作，或控制信号）。没有输出数据的算法相当于“什么也没做”。

（5）可行性。算法中所有的运算都可以精确地实现。

例如，两个有限位数整数相加，总能实现；而两个含有无限小数位的实数相加却不能

实现。

算法的存在形式有两种，一种是程序形式，另一种是描述形式。

算法的程序形式称为算法的实现（例如，C 语言程序等），是算法的最终形式。

算法的描述形式是算法的原始形式。

为了着重体现算法的主体，忽略支持算法执行的某些具体细节，这些细节在程序中是必不可少的，但常有碍于阅读，在介绍算法时，人们喜欢选用较为简单明晰的语言叙述算法，目的是便于理解、记忆和互相交流，即算法的描述形式。

算法的描述形式多种多样，通常可分为形式化描述，半形式化描述和非形式化描述三类。

算法的形式化描述通常指用类语言描述的形式，又称为“伪程序”，或伪代码。

所谓的类语言是指，以某种程序设计语言（C 语言，C++语言或 PASCAL 语言）作为“宿主语言”，对宿主语言略加改造而形成的语言，比如类 C 语言，类 C++语言，类 PASCAL 语言等。

为了突出算法设计的主体部分，伪程序中使用的语句可以不那么严格，甚至可以使用“杂牌语句”屏蔽某些处理细节（比如进栈退栈操作细节等）。尤其不要求带有大片大片的类型定义、变量定义、输入/输出等对算法只起“簿记性”工作的语句，只保留程序的基本框架结构，强调循环、递归、函数调用等主要控制性语句，使算法更加紧凑。

由于伪程序与“真程序”非常接近，只要对伪程序略加修改就可将其改造成可以上机运行的真程序，所以算法的形式化描述形式是一种“半程序”形式。

当然，算法的形式化描述和“半形式化”描述没有严格的界限。如果对算法的实现细节屏蔽得过强，或夹杂大量的文字叙述语句，与真程序相差太远，那么，这样的伪程序只是算法的半形式化描述形式。

算法的流程图（简称流图）也是算法的一种半形式化描述形式。

算法的非形式化描述，主要是指用自然语言（如中文、英文等）的叙述形式，是算法的最原始形式。这种形式直接记录了人们求解问题的思维过程，简单直观，且易于理解。当然，为便于思考和阅读，也使算法的处理步骤更精确些，最好采用分步骤的叙述方式。

例 1.1 自然选择排序算法。

把一组（共 n 个）杂乱数据排列成由小到大的递增序列，即排序问题。排序的算法很多，其中最简单的一种方法是简单选择法（simple selection sort），或自然选择法，文字叙述如下：

先从 n 个数据中选出一个最小元素，使它作为序列的第一项；再从剩下的 $n-1$ 个数据中选出一个最小元素作为第二项……，重复地作上述选择工作，直至选择最后一项。最终将得到有序序列。

这种描述形式并不依赖数据结构，不需要考虑数据是如何存储的。如果假定数据存储于数组 $a[n]$ 中，可进一步描述如下：

先从 a 的 n 个元素中选择一个最小元素（比如某个 $a[j]$ ），将它与 $a[0]$ 交换位置；

再从剩下的 $n-1$ 个元素中选择一个最小元素，将它与 $a[1]$ 交换位置；

再从剩下的 $n-2$ 个元素中选择一个最小元素，将它与 $a[2]$ 交换位置；

……

反复进行上述选择和交换，直到选择出最后一个元素，从而完成排序工作。