面向工程教育的本科计算机类专业系列教材

Data Structures and Algorithms

数据结构与算法

赵仲孟　张选平　耿 彧　编著

教育出版社

面向工程教育的本科计算机类专业系列教材

Data Structures and Algorithms

数据结构与算法

赵仲孟　张选平　耿 彧　编著

高等教育出版社·北京

内容提要

本书根据教育部计算机类专业教学指导委员会制定的“计算机科学与技术专业规范”中“数据结构与算法”课程大纲和专业培养方案要求编写。教材跟踪和反映国内外技术发展需求和教学改革现状，重点强调数据结构和算法设计两方面的基础知识，体系科学完整，内容简洁实用，实践性强。

本书共10章。第1~8章主要介绍数据结构及经典应用算法，内容包括基本概念、三大基本结构（线性结构、树形结构、图结构）和两大经典应用算法（排序算法、查找算法）。第9~10章主要介绍算法设计方法及应用，内容包括贪心算法、分治算法、动态规划、回溯算法和NP完全性理论。每章均附有知识要点、重点提示、常见问题解答、本章小结及大量的习题，针对难点问题还同时提供微视频讲解（读者可扫描相应二维码观看）。附录给出了课内实验和专题实验指导。

为便于读者使用，本书同时配有电子教案、习题解答、程序代码等资源。详见与本书配套的易课程网站。

本书既可作为高等学校计算机类专业“数据结构与算法”课程教材，也可供从事计算机应用开发和研究的工程技术人员参考。

图书在版编目（CIP）数据

数据结构与算法 / 赵仲孟，张选平，耿彧编著. --
北京：高等教育出版社，2016.11
ISBN 978-7-04-046322-4

Ⅰ. ①数… Ⅱ. ①赵… ②张… ③耿… Ⅲ. ①数据结构②算法分析 Ⅳ. ①TP311.12②TP312

中国版本图书馆CIP数据核字(2016)第196386号

策划编辑	倪文慧	责任编辑	倪文慧	封面设计	赵 阳	版式设计	童 丹
插图绘制	杜晓丹	责任校对	胡美萍	责任印制	毛斯璐		

出版发行	高等教育出版社	网 址	http://www.hep.edu.cn
社 址	北京市西城区德外大街4号		http://www.hep.com.cn
邮政编码	100120	网上订购	http://www.hepmall.com.cn
印 刷	北京中科印刷有限公司		http://www.hepmall.com
开 本	787mm×1092mm 1/16		http://www.hepmall.cn
印 张	23.5		
字 数	530千字	版 次	2016年11月第1版
购书热线	010-58581118	印 次	2016年11月第1次印刷
咨询电话	400-810-0598	定 价	40.00元

物料号 46322-00

数字课程资源使用说明

与本书配套的数字课程资源发布在高等教育出版社易课程网站，请登录网站后开始课程学习。

一、注册/登录

访问 http://abook.hep.com.cn/18718020，点击“注册”，在注册页面输入用户名、密码及常用的邮箱进行注册。已注册的用户直接输入用户名和密码登录即可进入“我的课程”页面。

二、课程绑定

点击“我的课程”页面右上方“绑定课程”，正确输入教材封底防伪标签上的 20 位密码，点击“确定”完成课程绑定。

三、访问课程

在“正在学习”列表中选择已绑定的课程，点击“进入课程”即可浏览或下载与本书配套的课程资源。刚绑定的课程请在“申请学习”列表中选择相应课程并点击“进入课程”。

四、与本书配套的易课程数字课程资源包括案例素材，以便读者学习使用。

账号自登录之日起一年内有效，过期作废。

如有账号问题，请发邮件至：abook@hep.com.cn。

前　言

数据结构由美国 D. E. Knuth 开创其最初体系，于 1968 年开始在国外作为一门独立课程设立。随着计算机科学的发展，数据结构课程的重要性越发彰显。作者早期使用清华大学严蔚敏教授编写的《数据结构(C 语言版)》作为教材，近年顺应高校双语教学模式的改革，改为使用 Clifford A. Shaffer 编写的英文版教材《数据结构与算法分析(C++ 版)》。尽管这些书籍都堪称经典，但在实际教学中，仍有不少学生发出这样的感慨：我们都知道学好数据结构的重要性，但它太抽象，有些算法理解很困难，学得很吃力，不清楚数据结构在实际应用中的价值及关系。为此，作者编写了本教材，力图结合多年积累的授课经验，用精炼而通俗的语言把复杂问题简单化、条理化，在传授数据结构与算法知识的同时，让学生更好地感受计算机编程技术的魅力，在未来能设计出性优的数据结构和高效的算法解决实际问题。

在当今“互联网 +”的信息化时代，大数据的理念深入程序设计人员的思维，信息范围不断拓宽，信息结构更加复杂，已从结构简单的纯数值性问题发展到了复杂数据结构的非数值性问题。计算机技术的飞速发展，促使人们沉浸于各种各样的计算机应用之中，例如微信、QQ、淘宝、支付宝、办公自动化系统、生产自动化系统、智能家电系统等，这些都需要编制计算机软件来完成，而选择合理的数据结构和设计高效的算法是任何软件实现的基础。从计算机科学家、图灵奖获得者 Niklaus Wirth 提出的“算法 + 数据结构 = 程序”就可看出，程序设计的精髓是数据结构和算法，算法必须依附于具体的数据结构，数据结构则直接关系到算法的选择和效率。数据结构是对所求解问题中数据的逻辑结构和存储结构的定义；算法是求解问题的过程描述，也依赖于数据的存储；程序设计语言只是一种在计算机上求解问题时，用来描述算法和数据结构的工具。因此，程序设计人员不但需要掌握一定的程序设计技巧，还必须研究计算机程序加工的对象，即各种数据的特性以及数据之间的关系，才能编制更好的软件系统。

数据结构是高校本科计算机类专业的一门综合性专业基础课，是介于数学、计算机硬件和计算机软件三者之间的一门核心课程。该课程主要培养学生在系统开发中的数据结构设计、算法设计与分析以及数据组织能力，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的重要基础。本书根据教育部计算机类专业教学指导委员会制定的“计算机科学与技术专业规范”中“数据结构与算法”课程大纲和专业培养方案要求编写。教材密切跟踪和反映国内外技术发展和教学改革需求，重点强调数据结构和算法设计两方面的基础知识、基本结构、经典算法和基本算法策略，体系科学完整，内容简洁实用，实践性强。本书不仅适用于初学者，还可作为研究生入学考试的参考教材，对有一定经验的程序设计人员也具有一定的参考价值。

本书沿用线性结构、树形结构和图结构 3 种数据结构为主线，从逻辑层面上分析各种结构的关系特征及抽象操作，分析各种结构在计算机中的存储表示及基本操作与实现，讨论各种结

构的应用，并介绍算法设计方法及应用。全书共10章，分为数据结构与经典算法、算法设计策略两大部分。其中，第一部分为第1—8章，主要介绍数据结构及经典应用算法，内容包括一类基本概念(算法复杂度)、三大基本数据结构(线性结构、树形结构、图结构)和两大经典应用算法(排序算法、查找算法)。第二部分为第9—10章，主要介绍算法设计方法及应用，内容包括贪心算法、分治算法、动态规划、回溯算法和NP完全性理论。

本书有以下特色：

(1) 实用性强。充分考虑了计算机类专业的教学特点，案例设计具有经典性。

(2) 深入浅出，内容翔实。知识安排由浅入深，运用分析、图表等多种方式，突破难点，使教师易教、学生易学。

(3) 教材建设立体化。由纸质教材及整合各种教学资源的网络平台构成。针对当前MOOC教学模式的特点，将一些重难点讲解制作成微视频，在教材的相应位置标注二维码，学生通过手机等移动终端扫描二维码即可观看，满足个性化学习的需求。另外，在网络平台上还提供了各章教学PPT、习题参考答案、算法程序等教学资源。

(4) 针对学生在学习中遇到的常见问题进行了归纳整理，并在教材的相应部分进行注释，不仅解答学生在学习中可能遇到的疑问，还起到了提示作用，为其进一步深入学习排除理解障碍。

(5) 每章均提供了大量习题，对每个知识点都有相应的题目进行练习，加深学生对知识的消化理解。对于有考研需求的学生，也提供了指导。

(6) 由于算法设计的性能直接影响到软件的执行效率，所以本书特别编排两章对算法设计进行介绍。希望能提高学生的算法设计能力，以便更好地解决实际应用问题。

(7) 附录提供课内实验设计和专题实验设计指导。通过实验设计内容，培养学生规范编写程序的能力，为其今后从事相关工作打下良好的基础。

本书获得西安交通大学本科“十三五”规划教材资助。具体的编写分工为：赵仲孟负责编写第1、2章及统稿，张亚明负责编写第3、4章，耿彧负责编写第5、6章，王嘉寅负责编写第7章，朱晓燕负责编写第8章，张选平负责编写第9、10章，王学成负责编写各章习题。研究生许会彬、杨蓉蓉、赵信等参与了部分章节的资料整理。参加习题作答与校稿工作的研究生有刘建业、许静、崔代兵、冯旋、杜婴、李洋、崔荐健。

由于作者水平所限，书中难免出现疏漏，在此恳切地期盼读者特别是任课教师在使用本书的过程中提出宝贵意见和建议，以便进一步完善教材内容，更好地服务于课程教学。作者的Email：zmzhao@mail.xjtu.edu.cn。

作 者

2016年8月

目　录

第1章 基础知识

▶ 知识要点

(1) 理解数据结构的基本术语，掌握抽象数据类型与数据结构的关系。
(2) 能够描述数据的逻辑结构。
(3) 了解算法增长率的概念，掌握渐近分析的表示方法。
(4) 能够对程序、算法或问题的上下限作出估算。
(5) 掌握时间复杂度与空间复杂度的分析方法。
(6) 初步了解 C++ 编程语言基础知识。

计算机技术的迅猛发展已远远超过了人们预料，其应用范围也在互联网和大数据背景下得以迅速扩展，不再局限于科学计算，而是渗透到各个领域，如控制、管理、通信、物联网、云计算、信息处理等许多非数据处理领域。相关的数据处理对象也从数值发展到字符、表格和图形等带有结构的数据。由此，程序设计面临新的问题和新的挑战。在计算机中如何组织数据、处理数据也随之成为程序设计的关键。掌握数据在计算机中的各种组织和处理方法是继续深入学习计算机理论的基础。

信息的表示是计算机科学的基础。大多数计算机的主要目标不仅是完成计算，更重要的是存储、检索和处理信息。从存储空间和运行时间的实现角度来看，程序必须组织信息，以支持高效的信息处理过程。数据结构、算法描述、性能分析和度量是设计和实现应用程序的四大要素。

学习数据结构可以提高学生分析和设计算法的能力及对问题进行抽象的能力。采用不同的数据结构和算法，会使程序执行效率相差甚远，同时影响程序代码的可重用性。要实现数据的高效利用，就必须研究“数据结构与算法”。

1.1 数据结构的基本概念

自然界中许多问题的解决方案是由数据结构进行描述的。例如，读者到图书馆借书关心的是图书馆是否有此书以及书在哪里存放，这一问题的解决必须根据图书目录信息进行查询。又如，计算机对弈问题就是计算机根据一定的策略进行选择判断的过程，可从一种格局派生出多种格局，这一过程需要由数据结构进行描述和解决。使用计算机解决实际问题，大致需要以下3个步骤：

(1) 分析实际问题，抽象出一个适当的数学模型。

(2) 为此数学模型设计一个合适的数据结构。

(3) 设计和实现具体操作的算法。

实质上，好的程序设计就是合适的数据结构加上好的算法。

数据结构涉及如下的一些基本概念和术语。

(1) 数据(data)：指计算机能接受和处理的一切对象。如整数、实数、复数是数据，字符、文字、表格、图形、图像、声音、程序等也是计算机能够接受和处理的数据。总之，存储在计算机中可用二进制表示的内容都是数据。

(2) 数据元素(data element)：指在计算机中作为一个整体考虑和处理的对象，是组成数据的基本单位。一般来说，数据元素是由若干数据项(data item)组成的，数据项是具有独立含义的数据的最小单位。如图书销售系统中，“图书”是数据元素，由商品ID、品名、类别、单价、出版社等数据项组成。

(3) 数据对象(data object)：指性质相同的数据元素的集合。如自然数的数据对象是集合 $N=\{1,2,3\cdots\}$，字符数据对象是集合 $Letter=\{A,B,C,\cdots,Z,a,b,\cdots,z\}$，图书数据对象是集合 $B=\{Book1,Book2,Book3\cdots\}$。故数据对象也就是一组特定实例或值的总称。

(4) 数据结构(Data Structure，DS)：研究数据及数据之间的关系，包括以下3个方面。

① 数据的逻辑结构(logic structure)，即依据实际情况描述数据元素之间的逻辑关系。

② 数据的存储结构(storage structure)，或称物理结构(physical structure)，即数据在计算机中的表示和存储方式。

③ 数据的操作(operation)实现，即在相应数据结构存储下所提供的基本操作的算法实现。

例如，手机中的通讯录为数据对象，每一个联系人为一个数据元素。在通讯录中有若干数据项，常见的有姓名、移动电话、办公电话、E-mail、QQ等，如表1-1所示。通讯录如何进行表示、按照什么方式组织联系人属于数据之间的逻辑关系。对于通讯录中的相关信息在手机中如何保存、如何表示它们之间的关系属于数据的存储结构。设计算法实现在通讯录中插入、删除、排序、修改及查找某联系人的信息均属于数据的运算范畴。

表1-1 手机通讯录的数据结构

姓名	移动电话	办公电话	E-mail	QQ
张一	13002900101	029-11111111	zhyi@163.com	129159164
王二	13802900202	029-22222222	wer@sina.com	204554897
李三	13902900303	029-33333333	lisan@qq.com	705897623
赵四	13002911101	029-44444444	zhsi@126.com	422215108
…	…	…	…	…

数据的逻辑结构按关系分为线性结构(数据元素之间的关系是线性组织的)和非线性结构(数据元素之间的关系是非线性组织的)。线性结构包括线性表、栈和队列等。非线性结构包括树、二叉树、图以及集合等。数据的逻辑结构可用一个层次图描述，如图 1-1 所示。

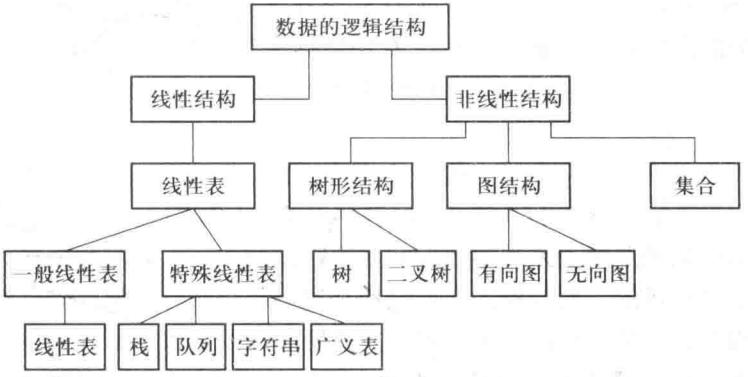
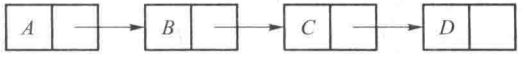
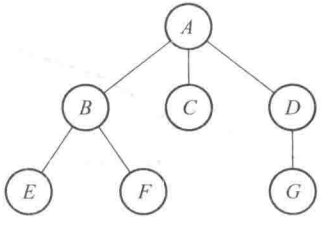


图 1-1 数据逻辑关系图

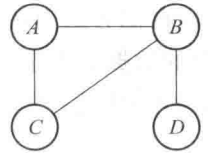
数据的逻辑结构强调数据元素之间的逻辑关系，反映数据的特性。任何数据结构都是由数据集和关系集组成，$DS=(D,R)$。其中，D 表示数据元素集合，R 表示数据元素之间的关系集合。图 1-2 描述了线性结构与非线性结构的典型例子。



(a) 线性表L



(b) 树T



(c) 图G

图 1-2 数据的逻辑结构示例

图 1-2(a)描述了线性结构——线性表 L。L 用数据集与关系集表示如下：

$$L=(D,R)$$

其中：

$$D=\{A,B,C,D\}$$

$$R=\{<A,B>,<B,C>,<C,D>\}$$

图 1-2(b)描述了非线性结构——树 T。T 用数据集与关系集表示如下：

$$T=(D,R)$$

其中：

$$D=\{A,B,C,D,E,F,G\}$$

$$R=\{<A,B>,<A,C>,<A,D>,<B,E>,<B,F>,<D,G>\}$$

图 1-2(c)描述了非线性结构——图 G。G 用数据集与关系集表示如下：

$$G=(D,R)$$

其中：

$$D=\{A,B,C,D\}$$

$$R=\{(A,B),(A,C),(B,C),(B,D)\}$$

数据的逻辑结构是从逻辑上来观察数据和数据之间的关系，它与数据在计算机中的存储无关，因此是独立于计算机的。而数据的存储结构是逻辑结构在计算机存储器中的具体实现，因而是依赖于计算机的，由数据在内存中的存放方式所决定。所以，研究数据的存储结构就是研究数据集 D 和数据之间关系集 R 的计算机表示。同一种逻辑结构可以有多种不同的存储结构。

计算机的存储器是由多个存储单元组成的，每个存储单元有唯一的地址。数据在计算机中的存放有以下 4 种基本的存储方法。

(1) 顺序(sequential)存储方法：就是把每个数据元素按某种顺序存放在一段连续的存储单元中。其局限性一是需要足够大的连续存储空间，不能有效利用零碎小块；二是事先无法得知所需存储空间的大小，预留过大或过小都不合理。

(2) 链式(linked)存储方法：就是把每个数据元素按结点结构分别零散地存放在存储单元中。这种方法就是将结点所占的存储单元分为两部分：一部分存放结点本身的元素信息，另一部分存放此结点的逻辑前驱或后继结点所对应的存储地址，称为指针项。指针项也可以有多个。

(3) 索引(index)存储方法：按关键字段建立索引表，用结点的索引号来确定结点的存储地址，而把每个结点的数据元素按一定规律顺序存放在存储单元中。

(4) 散列(hash)存储方法：设计散列函数，每个结点的存储单元位置通过设计的散列函数计算得到。

数据运算是定义在数据的逻辑结构上的一组操作，每种数据结构都有自己的运算集。本书主要研究数据运算在给定存储下如何在计算机中编程实现。

在解决实际问题中，一个数据结构的选择需以下 3 步。

(1) 分析问题，确定解决问题需满足的资源限制。一种算法如能在所需求的资源限制内将问题解决好，则称该算法是有效率的。

(2) 确定满足资源限制的基本操作。例如，对数据结构中的数据元素进行插入、删除、查找和定位等。

(3) 选择满足需求的数据结构。每种数据结构都与代价和效益紧密相连，每一个问题都有时间和空间资源的限制。

因此，选择数据结构是以数据为中心的设计理念，即先定义数据以及对该数据的操作，然后确定数据的表示方法，最后在计算机中实现数据表示。

1.2 抽象数据类型

抽象是指抽取出事物具有普遍性的本质，忽略非本质的细节，从而使所设计的数据结构更具有一般性，可以解决一类问题。使用者仅需了解抽象操作或界面服务，通过界面中的服务来访问这些数据。抽象数据类型(Abstract Data Type，ADT)是指数据结构作为一个软件组件的实现，也是一种研究数据结构的方式，它独立于具体的语言，也与某种具体的实现方式无关。研究数据结构的重点集中于抽象数据类型。ADT 本身含有封装和泛化的思想，将声明和实现相分离。故利用 ADT 研究数据结构的优点在于信息隐蔽，模块化，说明的精确性、简单性、完整性以及实现的独立性。

ADT 是数据结构的外部特性，主要描述该数据结构对外所提供的功能，只提供成员函数的接口定义，没有具体的成员函数实现过程。而本书中所研究的数据结构就是 ADT 类型在计算机内部的具体实现。

数据项也有逻辑形式和物理形式两个方面。用 ADT 给出的数据项定义是其逻辑形式，数据结构中对数据项的实现是其物理形式。图 1-3 说明了数据项的逻辑形式和物理形式之间的关系。实现一个 ADT，也就是处理相关数据项的物理形式。在程序中其他地方使用 ADT 时，则涉及相关数据项的逻辑形式。

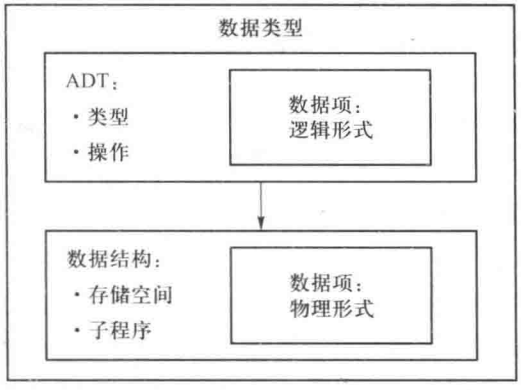


图 1-3 数据项、抽象数据类型和数据结构之间的关系

抽象数据类型的定义是由一组值域和定义在这组值域上的一组操作构成的。而抽象数据类型的表示和实现则要依赖于所使用的语言功能。到目前为止，ADT 的表示和实现概括起来应该有以下 3 种情况：

(1) 面向过程的表示和实现，如 C 语言。

(2) 面向模块结构的表示和实现，如 Turbo Pascal 4.0 版中的 Unit。

(3) 面向对象技术的表示和实现，如 Visual C++ 、Visual Basic、Java 等。

抽象数据类型可用三元组(D,R,P)表示。其中，D 表示数据对象，R 是 D 上的关系，P 是对 D 的基本操作集。

```
ADT 抽象数据类型名 {
    数据对象：〈数据对象的定义〉
    数据关系：〈数据关系的定义〉
    基本操作：〈基本操作的定义〉
} ADT 抽象数据类型名
```

其中数据对象和数据关系的定义用伪码描述，基本操作的定义格式如下：

```
基本操作名(参数表)
初始条件：〈初始条件描述〉
操作结构：〈操作结构的功能描述〉
```

表 1-1 中所示的手机通讯录的抽象数据类型描述如下：

```
ADT Address_list{
    数据对象：D = {a_i | a_i ∈ ElemSet, i = 1,2,…,n; n≥0}
    数据关系：R = { <a_i, a_{i+1}> | a_i, a_{i+1} ∈ D, i = 1,2,…,n}
    基本操作：添加新联系人(指定人)；
              删除联系人(指定人)；
              排序联系人(指定数据项)；
              修改联系人(指定人)；
              查询联系人(指定数据项)；
              等等；
} ADT Address_list
```

1.3 问题、算法和程序

问题(problem)是一种计算机需要完成的任务。一般说来，当计算机解决一个具体问题时，通常要从实际问题中抽象出一个适当的数学模型，寻求数学模型的实质就是分析问题，从中提取出各类操作对象，找出对象间所含的关系，再用数学语言加以描述。例如，预测肿瘤增长速度及平均发展情况，以及药物在体内分布变化规律的数学模型是微分方程，预测商品销售量的数学模型是抛物线方程。但是，计算机要在准确定义并完全理解问题的基础上才能研究问题的解决方法，问题的定义中应该包含对任何可行方案所需资源的限制。对于任意一个需要计算机解决的实际问题，总会存在某些资源限制。例如，任何计算机程序只能使用可用的主存储器和磁盘空间，而且必须在合理的时间内完成运行。

算法(algorithm)是计算机对特定问题求解步骤的一种描述，它是指令的有限序列。如果把

问题抽象出一个数学模型，那么算法就是求解该模型的一个方法或者一系列步骤。算法的设计和分析是数据结构中研究的重要内容，一个问题通常可以有多种算法来解决，设计高效算法对于开发大规模计算机系统起着关键的作用。此外，算法还应具有以下几个特性。

（1）有穷性：对任何合法的输入，一个算法的执行步骤是有限的，且每一步都可在有穷时间内完成。

（2）确定性：算法中的每个步骤都应明确，算法的每一条指令必须有确切的含义，使读者理解时不会产生歧义。而且，算法在给定输入条件下都有且只有一条实际的执行路径，即相同的输入下得到相同的输出。

（3）可行性：指令可以在现在的计算技术基础上实现，即算法是有效的。

（4）输入/输出：任何算法都是对输入数据加工的过程，最后给出输出结果。

（5）通用性：算法要具有一般性，对一般的数据集合都要成立。

（6）可读性：具备良好可读性的算法有利于查错和理解。

（7）健壮性：当输入数据非法时，算法能适当地处理并作出反应，而不应出现死机状况或输出异常结果。

时间复杂度与空间复杂度是算法分析中的两个重要因素，即如何使算法执行速度尽可能快、所需时间尽可能短，如何使执行算法运行所需的存储量尽可能小。任何算法可以用自然语言、伪代码或者某种计算机语言来描述。

程序(program)是指一组指示计算机每一步动作的指令序列，通常用某种程序设计语言编写，运行于某种目标体系结构上。因此，程序可以被认为是算法的具体实现。尽管“算法”和“程序”是两个相互独立的概念，常因简化表达而将两者混用。

只有能终止的程序才是真正意义上的算法，很长的程序(需要运行相当长的时间)或者死循环的程序都不是算法。

以上内容可以总结为：首先要从“问题”中抽象出数学模型，然后设计一个求解此数学模型的“算法”，选择合适的数据结构，最后编出“程序”，进行调试直至得出解答。

1.4 算法分析概述

算法分析是对一个算法所需的资源进行预测的过程。这里的资源是指程序或算法运行时所需的计算时间和存储空间。对于一个给定问题通常可以采用多种候选方案来解决，而各个解决方案之间存在着性能差异，如何选择最佳方案呢？主要通过时间及空间复杂度对其进行评价。

在数据结构中，假定算法正确的情况下，两种算法解决问题的效率可以采用分析法或实验法来确定。实验法需要先运行依据算法编写的程序，通过计算机内部的计时功能可精确得到运行时间，而所得的时间统计量依赖于计算机的软硬件等因素，因此算法本身的优劣性可能被掩盖。所以在实际中通常采用分析法，即不需要运行程序就可估算出运行效率的优劣。

一个用高级程序语言编写的程序在计算机上运行时所消耗的时间取决于下列因素：