左　飞　编著
飞思科技产品研发中心　监制

C/C++ 开发专家

专业人士
权威经典

C++

数据结构原理与经典问题求解

5 STAR

内外兼修：精湛高超的 C++ 编程技巧与极富魅力的算法设计艺术相得益彰

神形并重：生动的经典问题求解与丰富的数据结构原理娓娓道来

润物无声：编程技能的培养与抽象思维的训练浑然一体

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

左　飞　编著
飞思科技产品研发中心　监制

C/C++ 开发专家

C++ 数据结构原理与经典问题求解

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内容简介

本书是一部关于计算机科学与工程领域基础性核心课程——数据结构与算法的专著。全书以典型数据结构、程序设计方法及问题求解方法为研究对象，用C++面向对象程序设计语言作为描述语言，时刻突出对经典问题求解这一要旨，并将丰富的C++语言程序设计实践融入其中。

全书采用“数据结构原理描述→面向对象实现→解决经典问题→STL介绍”的基本架构，既强调理论的完整性，又突出实例引导的驱动性，用经典问题和大量背景描述提高读者的阅读兴趣，从而使原本枯燥的理论变得妙趣横生。基于上述框架，本书简要回顾了基本C++程序设计方法后，又全面系统地介绍了链表、队列、栈、树、图等基本数据结构。此外，本书还提供了近百个算法、数十个经典问题和十余个综合问题的完整实现代码近万余行。

本书内容实用，体例新颖，结构清晰，既可以作为大、中专院校在校师生相关课程的参考书，也可以作为信息学竞赛中数据结构方面的辅导用书。此外，本书也可供计算机科学与工程领域从业人员参考和查阅。

图书在版编目（CIP）数据

C++数据结构原理与经典问题求解 / 左飞编著.—北京：电子工业出版社，2008.10

（C/C++开发专家）

ISBN 978-7-121-07321-2

Ⅰ.C… Ⅱ.左… Ⅲ.①C语言－程序设计②数据结构 Ⅳ.TP312 TP311.12

中国版本图书馆CIP数据核字（2008）第134625号

责任编辑：王树伟 李新承

印　　刷：北京智力达印刷有限公司

装　　订：北京中新伟业印刷有限公司

出版发行：电子工业出版社

北京市海淀区万寿路173信箱　邮编：100036

开　　本：787×1092　1/16　印张：34　字数：870.4千字

印　　次：2008年10月第1次印刷

印　　数：5 000册　　定价：55.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：（010）88258888。

出版说明

“聪明的人使用 Delphi，真正的程序员使用 C++。”
时至今日，这句曾经在业内广为流行的话语又增添了更丰富的内涵。
脚本语言、Java、.NET 等正在争夺更大的天地。
然而，C/C++仍不失为最好、最纯粹的编程语言。

—— “C/C++开发专家”引导你成为真正的程序员

C/C++的发展

作为一种结构化的中高级编程语言，C 语言具有功能齐全、适用范围广的优势，一直为很多程序员所钟爱，并被视为最佳的编程入门语言，拥有着庞大的使用和学习人群。C++是在 C 语言基础上开发的一种集面向对象编程、通用编程和传统的过程化编程于一体的编程语言，是目前业界广泛使用的一种编程语言。然而，软件产业的规模和环境发展到今天，已经发生了深刻的变化。如今企业级应用整合与开发的任务主要由 Java、基于.NET 平台的 C#及各种新型动态语言来承担。C++的应用场合有所收缩，不再像之前那样从上到下包打天下，呈现出鲜明的行业应用特色。未来 C++主要在系统级复杂应用程序，高性能、实时中间件和嵌入式领域发挥所长。随着多核 CPU 的普及和网络安全重要性的空前提升，在并发程序设计、系统安全及视频处理、嵌入式开发方面，C++将获得新的应用空间。在大规模、高性能计算，游戏开发、嵌入式实时应用开发方面，以及一些传统的客户端软件和构件开发中，C++也将继续保持其稳定的地位。

C/C++的图书现状

C++的教学和使用具有其复杂性，而传统图书和学习方法的各种弊端更加剧了这一现象，使 C++成为不少人望而生畏的难学、难用的“专家语言”。虽然国内的 C/C++图书并不缺乏，但大多只适合有一定经验的程序员提升功力之用，而内容全面准确、讲解循序渐进、学习简明易懂的原创图书并不多见。近期 C/C++图书市场存在如下特点：

1．国外经典图书全面翻新。近年来国外一些书商根据 C++所发生的变化，不断地进行版本升级或全面改写书稿，推出新的力作。

2．国内原创图书缺乏力作。近年来国内虽然有一批令人耳目一新的 C++好书面世，但在技术层面上对实践的关注略显不足，难解读者之渴。

3．关键性图书存在空白。基于组件的软件开发、复杂网络应用，以及热度尚在的 COM 开发等方面的图书有待开发。

基于上述现状，我们组织 C/C++各应用领域的作者，推出本丛书“C/C++开发专家”，力求从新的、实用的、全面的角度介绍 C/C++，使其紧密地跟踪当前国内最实用、最热门的编程技术。我们期望通过这套丛书，能够提高各位读者的 C/C++开发水平及编程的实践能力，为我国计算机产业奉献一份微薄之力。

“C/C++开发专家”助你成为真正的程序员

“C/C++开发专家”的读者定位是：C/C++初学者，需要提升应用开发能力的程序员，具有实际开发经验的中高级程序员。对阅读本丛书的读者建议如下：

➢ *面向 C/C++初学者*

本丛书通俗易懂，并自成体系。丛书全面介绍 C/C++及 Visual C++的编程技术和实践操作。通过学习，初学者可快速地掌握涉及 OOP、STL、泛型编程等标准 C/C++的内容，对 C/C++技术应用有更深刻的理解。

➢ *面向需要提升应用开发能力的程序员*

对于那些急需提升应用开发能力的程序员来说，本丛书是再好不过的专家向导。丛书除全面介绍标准 C/C++的内容外，还涉及数字图像处理、流媒体、网络通信和嵌入式开发等多个领域，可以为从事相关领域开发的程序员提供有益的帮助和参考。

➢ 面向具有实际开发经验的中高级程序员

本丛书同样适合于具有实际开发经验的中高级程序员。书中列举的大部分实例具体翔实，非常值得广大高级程序员学习和借鉴。

“C/C++开发专家”为程序员量身打造

本套丛书通过不同种类的图书来满足读者的需求。

➢ 语言入门

C/C++是一门优秀的高级语言。它绝不像一些传统图书所述是一门晦涩难懂、高深莫测的“专家语言”。本丛书的语言入门分支面向初学者，以通俗易懂的语言，介绍标准的C/C++语言知识，以及Visual C++编程技术；在保证知识体系的完整性的同时，在语言、体例上更贴近程序员的学习心理需求。

➢ 应用实践

如果脱离了具体的应用背景，任何一门计算机语言的学习都是“纸上谈兵”。如果程序员没有真正掌握面向应用的实践开发技能，那么很有可能面临来自就业的压力。本丛书的应用实践分支面向数字图像处理、流媒体、网络通信、嵌入式开发等不同的行业应用方向，介绍C/C++应用技术。目标是努力将读者培养成具有实际开发能力的从业人员。

➢ 开发详解

只让人阅读一遍的书很难说是一本好书。任何一本书在读者的眼中总会经历“厚→薄→厚”的过程。同样，C/C++语言会耐人寻味，但真正理解C/C++一般性内容需要花时间，而要做到融会贯通则更要下工夫。本丛书的开发详解分支针对C/C++语言及Visual C++中的高级特性，进行深入的剖析和讲解。C/C++程序员一旦掌握更高级的编程技巧，且对C/C++的语言内涵及开发技术有更为深入的理解，就能得心应手地运用这门语言。

➢ 技巧集锦

从大规模的并行计算到嵌入式系统开发，C/C++的应用领域非常广泛。即便是世界上最厚的一本书，也无法介绍所有的C/C++技术。针对这一特点，本丛书的技巧集锦分支对程序员经常遇到的问题进行解答和分析，并注重举一反三，启发读者思考。通过对这一话题的讨论，给正在从事或即将从事C/C++开发的程序员以最大的启迪。

“C/C++开发专家”丛书特色

本丛书具有如下特色。

➢ 由浅入深，通俗易懂

实际上，根本就不存在只面向纯粹的初学者的C/C++书籍。原因很简单：C/C++就不是初级的语言。初学者选择C/C++的时候，除了有足够的兴趣之外，还要有足够的耐心和恒心。为此，本丛书在保持完整性的同时注重语言的通俗性和知识的趣味性，避免了较为复杂的理论概念，取而代之的是常见的编程技巧和实际例子，力求由浅入深，通俗易懂，充分调动读者的阅读兴趣。

➢ 案例为主，内容生动

如果没有“案例”，C/C++的学习可能非常枯燥无趣；如果没有合适、有趣的“案例”， C/C++的学习仍会枯燥无趣。与以往的风格不同，本丛书强调编程实践，提供了大量的实例及源代码。这些案例均由作者从实际开发工作中设计的原型案例精简加工而成，形式丰富多样，具有很好的实用价值。

➢ 倡导正确的编程思想

“授之以鱼，不如授之以渔。”本丛书并非按部就班地完成知识传授，而是在介绍知识的同时倡导正确的学习思想和方法。如：倡导OOP思想、泛型编程、流行的设计模式、不断的重构理念和开源精神等。读者在阅读本书的同时，会接触到这些新的理念和方法。在某些开放性话题上，本丛书一反以往一些图书的“专家”面孔，更加贴近读者，从各个角度与读者展开交流和探讨。

飞思科技产品研发中心

前　言

■ 本书缘起

一家世界一流的 IT 公司给其面试者出了如下两道测试题。

1．一辆有 7 节车厢的列车在星期五下午 18 点 17 分离开车站，并以 50 km/h 的速度行驶。现在是周末，请问你要去哪里？

2．股票 A 目前的报价是 100 元。3 个月后，这个价钱可能涨到 120 元，也可能跌到 90 元。如果现在给你一次机会允许你用 110 元钱在接下来的 3 个月内买这个股票，你将如何使用这 110 元钱。请将你的决策过程告诉我们。

无独有偶，许多世界顶级的软件公司都喜欢在面试时问一些考查应试者思维能力的问题，为什么呢？道理很简单，单纯掌握一门编程语言并不足以编写出好的程序。重要的是，掌握思考问题的方法和解决问题的策略。

古语云：“授人以鱼，不如授人以渔。”如果说吃鱼是目的，那么钓鱼就是手段。尽管一条鱼能解一时之饥，但却不能解长久之饥；如果想永远有鱼吃，那就要学会钓鱼的方法。这就是作者写作本书的目的和动机。

本书不仅是一本向程序员传授专业技术的书，更是一本教导人们如何思考问题的书。

■ 本书对象

数据结构和算法是计算机程序设计领域的重要理论和技术基础，是计算机学科的核心课程，也是计算机科学研究的基础方向。数据结构不但为数据存储和问题解决提供逻辑结构基础，还可以提供一种抽象现实世界的思维方式，即使非计算机专业的人员学习和了解它也有助于发散思维，激发创意。而算法设计则在提供实际问题解决方案的同时，帮助训练人脑对问题的抽象能力和逻辑思维能力，因此学习这方面的知识也大有裨益。

- 对于计算机应用及程序设计相关方向的工程技术人员而言，数据结构和算法是处理实际问题的必备利器。
- 对于计算机相关方向的在校学生而言，数据结构和算法是深入学习本学科其他知识的必要保障。
- 对于热衷计算机技术或程序设计的业余爱好者而言，数据结构和算法是开拓思维、培养能力的必由之路。

■ 本书特点

尽管数据结构相关课程教育引入我国不过二三十年，但在这短短的时间里其相关领域的教学和科研成果层出不穷。近年来，有关数据结构的书籍可谓是“百花齐放，百家争鸣”。不言而喻，读者可选择的范围更广了，可研习的材料更多了，但在林林总总之间选择却也不由得令

人眩晕起来。观察现有的众多数据结构书籍，大部分教材注重相关理论的阐述，而在一定程度上忽略了这些理论的应用和实现，与程序设计本身相脱离，内容难免晦涩又缺乏实用性。本书则力求避其短而扬其长，以典型数据结构、程序设计方法及问题求解方法为研究对象，将三者融会贯通的同时，巧妙地将丰富的C++面向对象语言程序设计实践融入其中。最大限度地提高本书的含金量，以飨读者。

笔者始终本着强调重点、突出实践、精益求精、存同求异的思想努力将这本书写好。总的来说，本书主要有如下3点区别于其他同类书籍。

- **道法自然、内外兼修。**将程序设计方法、面向对象思想同数据结构与算法设计有机结合，进而突出工程性、实用性。
- **问题典型、实例经典。**时刻不离对经典问题求解这一要旨，通过生动丰富的问题描述突出经典问题求解的趣味性和实效性。
- **深入浅出、通俗易懂。**注意从日常生活中可能遇到的实际问题出发以引导读者思考，并杜绝生硬灌输和填鸭式教学。

■ 本书结构

本书以C++语言作为描述语言，时刻突出对经典问题求解这一要旨。全书采用“数据结构原理描述→面向对象实现→解决经典问题→STL介绍”的基本架构，既强调理论的完整性又突出实例引导的驱动性，用经典问题和大量背景描述提高读者的阅读兴趣，从而使原本枯燥的理论变得妙趣横生。基于上述框架，全书共分为11章。

第1~3章简要地回顾了基本C++程序设计方法，以为读者后续学习奠定基础。其中，第3章为过渡性章节，既承接了前一章中C++面向对象程序设计语言方面的内容，又为后面数据结构方面知识的介绍做了导引。

第4~10章（除第6章外）则是本书内容的主体部分，该部分全面系统地介绍了链表、队列、栈、树、图等基本数据结构，包括它们的设计原理和面向对象实现，同时配有生动丰富的实例问题分析，以避免枯燥和乏味。

第6、11章是关于算法设计思想方面的一些内容。其中第11章介绍的10种排序算法是针对具体问题展开的数种经典解决方案，可以认为是算法设计的一种实践，读者从中可以更深切地体会到数据结构和算法设计方面的核心内涵和精要所在。

此外，本书还提供了近百个算法、数十个经典问题和十余个综合问题的完整实现代码近万余行，参考价值很高，可供有兴趣的读者学习研究之用。

全书程序都在Microsoft Visual C++ 6.0环境下调试过，完整的代码可以从本书网站http://book.vcer.net/下载得到。

■ 关于本书

本书由左飞编著。在此期间，薛佟佟、姚远、张黎等参加了本书部分内容的编写及素材整理工作。李飘、初甲林、李冠廷、刘航、万晋森等参与了本书部分程序的编写和代码测试工作。这里，向他们表示最诚挚的感谢。与刘航、万晋森合作编写《Visual C++数字图像处理开发入

门与编程实践》一书（作者的另一本著作）的经历非常愉快，笔者也期待着再次与他们合作，并祝他们事业有成，前途似锦。最后，还要感谢白乔先生，感谢他长久以来给与作者的大力支持与帮助。

本书凝聚了作者的心血和汗水，甚至曾数易其稿，力求精益求精。然而由于编写时间仓促，纰漏和欠缺之处在所难免，还望读者不吝赐教和批评。联系信箱：fzuo@yahoo.cn，详情请垂询 http://books.vcer.net/data。

左　飞

2008年夏于花城广州

联系方式

咨询电话：（010）68134545　　88254161－67

电子邮件：support@fecit.com.cn

服务网址：http://www.fecit.com.cn　　http://www.fecit.net

通用网址：计算机图书、飞思、飞思教育、飞思科技、FECIT

目　录

第1章

绪论

数据结构是计算机科学中的基础理论之一，该领域任何问题的解决方案都不可能脱离数据结构而单独存在。本章作为全书的导引章节，将从数据与数据结构的概念谈起，帮助读者对将要学习的内容有一个初步的认识。此后，本章将讨论关于算法设计和算法分析的一些内容。全书选择面向对象的 C++作为描述语言，因此本章末尾也将对 C++做一些简单的介绍。有关 C++的更详细的内容将在本书第 2 章进行讨论。

本章学习地图

- 数据与数据结构
- 算法的基本概念
- C++语言简介

1.1 数据与数据结构

数据是一切有意义信息的基本存在形式。如何有效地组织和利用数据一直是计算机科学家们研究的重点。无论是学术研究还是工程应用，数据和数据结构都扮演着极其重要的角色。本节主要向读者介绍有关数据和数据结构的一些概念。

1.1.1 数据及其类型

数据（Data）是信息的载体，有序的数据组织就形成了信息。信息是人类可以直接利用或感知的意识形式。而数据则是用来被计算机识别、存储和处理的，它是计算机利用或感知的基本单位。

在计算机科学中，更加具体的数据定义可以表述为计算机加工和处理的对象，这里数据可以是数值数据，也可以是非数值数据。数值数据包括一些整数、实数或复数等，它主要用于工程和科学计算等；非数值数据包括文字、图像、声音等。在计算机中，这些数据都以二进制数的形式（由 0 和 1 组成）存放在物理介质上。每个二进制位称做一个位（bit），每 8 个二进制位组合起来称做一个字节（Byte）。

那么这些 0 和 1 是如何组成复杂而有意义的数据的呢？这就需要用到数据类型的概念了。所谓数据类型就是一个值的集合和定义在这个值集上的一组操作的总称。一般而言，数据类型可以分为原子型和结构型。在程序设计语言中，每一个数据都属于某种数据类型。类型明显或隐含地规定了数据的取值范围、存储方式，以及允许进行的运算。这些已经事先定义好的数据类型就是所谓的原子型。经过数据类型规定后的若干有序 0 和 1 的组合就在计算机中呈现出一定的意义，最初的表现形式就是原子型数值数据。而在某些时候，一旦需要引入某种新的数据类型时，总是借助编程语言所提供的原子型数据类型来描述或定义新数据的存储结构，这也就是所谓的结构型。换句话说，原子型数据经过一定规则重组后即可形成结构型数据。

非数值型数据也是用 0 和 1 来存储和表示的，只不过组成规则要复杂得多，其内容已经超出了本书的研究范围，因此这里不再赘述，有兴趣的读者可以参阅相关书籍。

下面首先来看看 C++语言中的几种常用原子数据类型，如表 1-1 所示。

表 1-1 C++中的基本数据类型

类 型 名	描 述	取 值 范 围
bool	布尔型	true，false
char （signed char）	字符型	-128~127
unsigned char	无符号字符型	0~255
short（signed short）	短整型	-32 768~32 767
unsigned short	无符号短整型	0~65 535

（续表）

类 型 名	描 述	取 值 范 围
int（signed int）	整型	-2 147 483 648~2 147 483 647
unsigned int	无符号整型	0~4 294 967 295
long（signed long）	长整型	-2 147 483 648~2 147 483 647
unsigned long	无符号长整型	0~4 294 967 295
float	浮点型	3.4×10^{-38}~3.4×10^{38}（绝对值精度）
double	双精度浮点型	1.7×10^{-308} ~1.7×10^{308}（绝对值精度）
long double	长双精度浮点型	1.7×10^{-308} ~1.7×10^{308}（绝对值精度）

这些数据在内存中存储时占据了多少内存空间呢？通过下面这段程序可以测得一些结果。

```
#include <cstdlib>
#include <iostream>
using namespace std;
int main() {
    cout << "bool: " << sizeof(bool) << endl;
    cout << "char: " << sizeof(char) << endl;
    cout << " short: " << sizeof(short) << endl;
    cout << "int: " << sizeof(int) << endl;
    cout << "long: " << sizeof(long) << endl;
    cout << "float: " << sizeof(float) << endl;
    cout << "double: " << sizeof(double) << endl;
    return 0;
}
```

编译并运行程序后，结果如图 1-1 所示。由此可见，布尔型数据在内存中仅占 1 字节，一个字符型数据也占 1 字节；短整型占两字节，整型、长整型和浮点型占 4 字节；双精度值占 8 字节。同样一组 0 和 1 的序列如果数据类型不同，那么编译出来的结果也是不一样的。

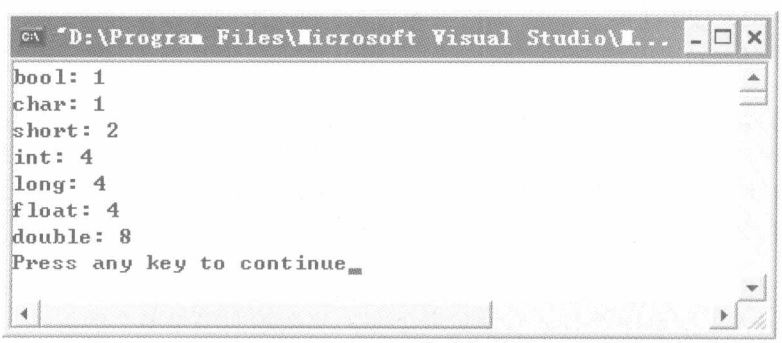

图 1-1 程序运行结果

结构化的数据可以由简单的数据组成。我们可以将某人的联系地址想象成一个结构化的

数据，那么在这个结构中可能有若干个项目，它们都是一些简单的基本数据。例如，居住地的门牌号，然后是所在街道的名字、城市的名称、省份的名称和邮政编码，所有这些都是简单的数字和汉字的组合。

C 语言允许用户自己定义数据类型，基本方式是通过结构体的形式将基本数据类型进行组合，生成结构型数据类型。例如，下面的一个结构体定义了一个 student 类型的数据。这个 student 类型是包括学号、姓名、性别和班级在内的若干个基本数据类型的组合。

```
struct student
{
    int num;
    char name[10];
    char sex;
    int class_number;
};
```

为使这种形式得以进化，在 C++中还使用类和对象来产生新的数据类型。关于类和对象将在下一章中进行更加详细的介绍。

前面的例子在于说明一个结构化的数据可以由一些简单的基本数据类型组合而成。而事实上，如果一个结构化的数据包含了其他结构时，它也可能变得非常复杂。例如，一个图书馆有很多书柜，书柜中存放了很多书籍。书柜中的每层格架上可以放一些书，也可以放一些装书的盒子。在这个例子中，书柜是一个由书籍与盒子组成的数据结构。盒子也是一个数据结构，因为盒子里面还放着书。这样一个结构化的数据就又包含了另外一种结构。结构化的数据既提供了一种存储数据的方式，还提供了一种问题解决途径。在图书馆的例子中，结构化的数据不但帮助我们将书籍有序地组织在一起，而且在数据查询中起到了至关重要的作用。再例如，一个装满文件和文件夹的文件柜，它是一个展现多个结构化的数据如何重组得到一个新结构化数据的典型例子。这里文件柜是一个完整的结构化数据。文件柜由抽屉组成，而抽屉中装有文件夹，文件夹中装有文件。

一旦数据之间通过结构化的方式被组织到一起，那么也就引出了数据结构的概念。

1.1.2 数据结构简介

当今世界纷繁复杂，人们每天都面对着林林总总的事物，同时信息也从各个方面以多种多样的形态呈现在我们面前。人们所面对的数据量有时是非常巨大的，如果没有一个有效的方法来储存、组织和表示信息，那么信息将变得没有任何价值。想想某个高校的图书馆里的藏书，如果没有图书管理员维护图书管理系统，那么想找到某本书是多么费劲的一件事啊。

即使面对一组相对较小的信息时，一个结构化的数据表示方法也是必不可少的。假想公共汽车站等车的人不按顺序争先恐后地抢着上车，那么场面将是多么的混乱！相反，如果人们排着队井然有序地逐个上车情况就好得多了。很明显，这种线性的数据组织方法在信息存储中是具有举足轻重的地位的。

简单说来，数据结构就是一个信息的结构化表示。无论是图书馆中用以存储书籍的系统，

还是电影院里排队的人都是数据结构在现实世界中的实例。可以说，数据结构为存储表示数据及问题解决带来了极大的便利。例如，在图书馆中，数据结构可以帮助人们解决每一本书该被放在哪个特定位置等这样的问题；而在电影院买票时，人们以排队等待的形式可以很容易决定出下一个该轮到谁。

数据结构也可以被认为是计算机存储、组织数据的方式。关于数据结构的研究距今已有30多年的历史了，关于现代计算机所采用的大量数据结构的最早的系统论述出现在图灵奖得主D.E.Knuth于1973年发表的著作《计算机程序设计技巧》中。至今数据结构已经成为计算机程序设计和问题求解的重要工具。通常情况下，正确地使用数据结构可以为高效的问题解决创造可能。数据结构是数据对象，以及存在于该对象的实例和组成实例的数据元素之间的各种联系。这些联系可以通过定义相关的函数来给出。现实世界中的万事万物都是相互联系的信息体，通过对现实世界进行高度抽象而得到的就是数据对象或数据元素。数据对象之间的关系和组织方式称为结构。基本的结构分为两大类，即线性结构和非线性结构。线性结构又包括队列、链表和栈等。非线性结构则包括树和图等。集合结构是独立于线性和非线性以外的一种结构，集合结构中的数据元素除了同属于一种类型外，其相互之间别无其他关系。

线性结构中元素之间存在一对一关系。其中，最通常的数据组织方式有两种，即顺序式和链式。顺序存储结构是一种最基本的存储表示方法，通常借助于程序设计语言中的数组来实现。顺序存储结构采用把逻辑上相邻的结点存储在物理位置相邻的存储单元里，以这种方式进行数据组织。此外，结点间的逻辑关系由存储单元的邻接关系来体现。与顺序式不同，链式结构追求更加灵活的数据组织结构。它不要求逻辑上相邻的结点在物理位置上也相邻，但结点间的逻辑关系是由附加的指针字段表示的。链式存储结构通常借助于程序设计语言中的指针类型来实现。

作为非线性结构的代表，树结构中元素之间存在一对多关系，而图结构中元素之间存在多对多关系，因此图结构又被称做网状结构。

算法的设计取决于数据结构，而算法的实现依赖于所采用的数据结构。数据的运算是在数据结构之上定义的操作算法。不同的数据结构具有各自不同的操作，对于同一类型的数据结构如果拥有操作的种类和数目不同，即使逻辑结构相同，数据结构能起的作用也不同。不同的数据结构其操作集是不同的，但某些特定的操作则是具有共性的，这些共同的数据操作包括如下内容：

- 结构的生成和销毁。
- 在结构中对满足条件的数据元素进行检索。
- 在结构中插入新的数据元素或删除已经存在的数据元素。
- 对数据结构中的数据进行遍历。

在描述数据结构时常采用抽象数据类型作为工具。抽象数据类型（ADT，Abstract Data Type）是对一个数学模型及定义在该模型上的一组操作的称谓。抽象数据类型实际上就是对该数据结构的定义。因为它定义了一个数据的逻辑结构，以及在此结构上的一组算法。抽象数据类型描述程序处理的实体时，强调的是其本质的特征、其所能完成的功能，以及它和外部用户之间交互的接口。该点特征首先体现了ADT的数据抽象特性。

1.2 算法

计算机科学家 Nikiklaus Wirth 曾提出一个著名的公式：数据结构+算法=程序，可见算法的重要性。因此脱离了算法的数据结构是不足以解决和处理问题的。本节将着重讨论有关算法的一些内容。

1.2.1 算法的概念

关于算法的定义，其表述方式有很多种。一种从强调算法跟数据结构关系角度出发的定义：除了有一个数据结构来表示和组织信息以外，要解决一个问题还需要一系列步骤来完成问题的解决过程。操作并使用数据结构来解决问题的一系列步骤称为算法。关于这个定义，这里可以举个简单的例子加以说明。例如，考虑先前谈到过的在电影院排队买票的例子，其有待处理的问题是如何管理一组等待买票的人，那么正在等待的一队人就是一个数据结构。解决这一个问题的算法是，当售票员将票卖给队列中排在最前的那个人后，就将这个人从队列中删除，然后将其后面紧挨着的人排到第一位并卖票给他，直到票卖光，或者没有人排队了为止。

当然，也可以从问题解决角度去定义算法，即算法是有穷规则构成的为解决某一类问题的运算序列。不论从何种角度去描述算法的定义，一些要素对于算法都是必不可少的。首先是输入，任何算法必须先接受一定的数据输入；然后是输出，经过一系列运算规则的处理，算法应将结果输入。

大多情况下，解决同一问题的方法可能有很多种。例如，求圆周率 π 的近似值这个问题，大家能够想出多少种算法呢？事实上，求圆周率 π 近似值的方法的确有很多。例如，首先可以使用下面这个公式来求出 π 的近似值。

$$1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\cdots+\frac{(-1)^{n-1}}{2n-1}+\cdots=\frac{\pi}{4}$$

其次，还可以使用所谓的蒙特卡洛法来求 π。这是一种基于随机数而设计的方法，它最早起源于 1777 年法国科学家蒲丰提出的随机投针法，1805 年瑞士天文学家沃尔夫使用该方法求得的 π 值为 3.1596，1901 年意大利数学家拉兹瑞尼使用该方法求得的 π 值 3.1415929。关于原始的蒲丰投针方法这里不再赘述。但需要说明的是原始的蒲丰投针方法是极其繁杂而低效的。随着计算机技术的发展，在计算机上模拟实现蒙特卡洛法求解 π 值的过程已经十分方便了。众所周知，圆的面积公式为：$S=\pi R^2$，当 R=1 时，S=π。如果在一个边长为 2 的正方形中有一个内切圆，那么该圆的面积就为 π。利用这一原理，可以假设有大量随机点等概率地均匀地落入正方形中。这时可以得知：