“十二五”普通高等教育本科国家级规划教材

普通高等教育“十一五”国家级规划教材
教育部普通高等教育精品教材
中国大学出版社图书奖优秀教材一等奖

普通高校本科计算机专业特色教材精选·算法与程序设计

数据结构
——从概念到C++实现（第3版）

王红梅 王慧 王新颖 编著

非外借

清华大学出版社

“十二五”普通高等教育本科国家级规划教材

普通高等教育“十一五”国家级规划教材
教育部普通高等教育精品教材
中国大学出版社图书奖优秀教材一等奖

普通高校本科计算机专业特色教材精选·算法与程序设计

数据结构
——从概念到C++实现（第3版）

王红梅 王慧 王新颖 编著

清华大学出版社
北京

内 容 简 介

数据结构是计算机及相关专业的核心课程，也是计算机及相关专业硕士研究生入学考试的必考科目，而且是理工专业的热门公选课程。本书介绍数据结构、算法以及抽象数据类型的概念；介绍线性表、栈和队列、字符串和多维数组、树和二叉树、图等常用数据结构；讨论查找和排序技术。本书合理规划教学内容，梳理知识单元及其拓扑结构，兼顾概念层和实现层，既强调数据结构的基本概念和原理方法，又注重数据结构的程序实现和实际运用，在提炼基础知识的同时，进行适当的扩展和提高。

本书内容丰富，层次清晰，深入浅出，结合实例，可作为高等学校计算机及相关专业数据结构课程的教材，也可供从事计算机软件开发和应用的工程技术人员参考和阅读。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。
版权所有，侵权必究。侵权举报电话：010-62782989　13701121933

图书在版编目(CIP)数据

数据结构：从概念到C++实现/王红梅，王慧，王新颖编著. —3版. —北京：清华大学出版社，2019
(普通高校本科计算机专业特色教材精选·算法与程序设计)
ISBN 978-7-302-50576-1

Ⅰ. ①数… Ⅱ. ①王… ②王… ③王… Ⅲ. ①数据结构 ②C语言－程序设计 Ⅳ. ①TP311.12 ②TP312.8

中国版本图书馆CIP数据核字(2018)第141998号

责任编辑：袁勤勇
封面设计：常雪影
责任校对：李建庄
责任印制：宋　林

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795954
印 装 者：清华大学印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：20.25　　**字　　数**：468千字
版　　次：2005年7月第1版　2019年5月第3版　　**印　　次**：2019年5月第1次印刷
定　　价：45.00元

产品编号：078470-01

前言

PREFACE

数据结构是计算机及相关专业的核心课程，也是计算机及相关专业硕士研究生入学考试的必考科目，而且是理工专业的热门公选课程。作为程序设计的重要补充和延伸，数据结构所讨论的知识内容、蕴含的技术方法、体现的思维方式，无论进一步学习计算机专业的其他课程，还是从事计算机领域的各项工作，都有着不可替代的作用。

数据结构课程的知识丰富，内容抽象，隐藏在各知识单元的概念和方法较多，贯穿于各知识单元的链表和递归更是加重了学习难度。笔者长期从事数据结构的研究和教学，深切理解学生在学习数据结构过程中遇到的问题和困惑，深入探究掌握数据结构的有效途径和方法，深刻思考数据结构对培养程序设计和计算思维的地位和作用，深度把握课程的教学目标和重点难点。本书第1版于2005年7月出版，第2版于2011年6月出版，国内有超过100所院校将本书作为主讲教材，第3版在体例和主要内容保持不变的基础上，在教学内容和教学设计等方面进行了如下处理。

① 合理规划教学内容。紧扣《高等学校计算机专业核心课程教学实施方案》和《计算机学科硕士研究生入学考试大纲》，涵盖教学方案及考研大纲要求的全部知识点。

② 遵循认知规律，厘清教学主线。根据学生的认知规律和课程的知识结构，按照从已知到未知的思维进程逐步推进教学内容，梳理和规划各知识单元及其拓扑结构，设计清晰的教学主线。知识单元及其拓扑结构如图1所示。

③ 提炼基础知识，适当扩展提高。考虑到不同学校教学要求的差异以及不同学生学习需求的差别，一方面本着“够用、实用”的原则，抓牢核心概念，提炼基础性知识，贯彻数据结构课程的基本教学要求；另一方面对某些知识点进行适当的扩充和提高（图1中打星号部分），这部分内容可用于选讲，也可用于学生自学或课外阅读（**教学建议：目录中打一个星号的可用于选讲，打两个星号的可用于课外阅读，其他是必讲的基础知识**）。

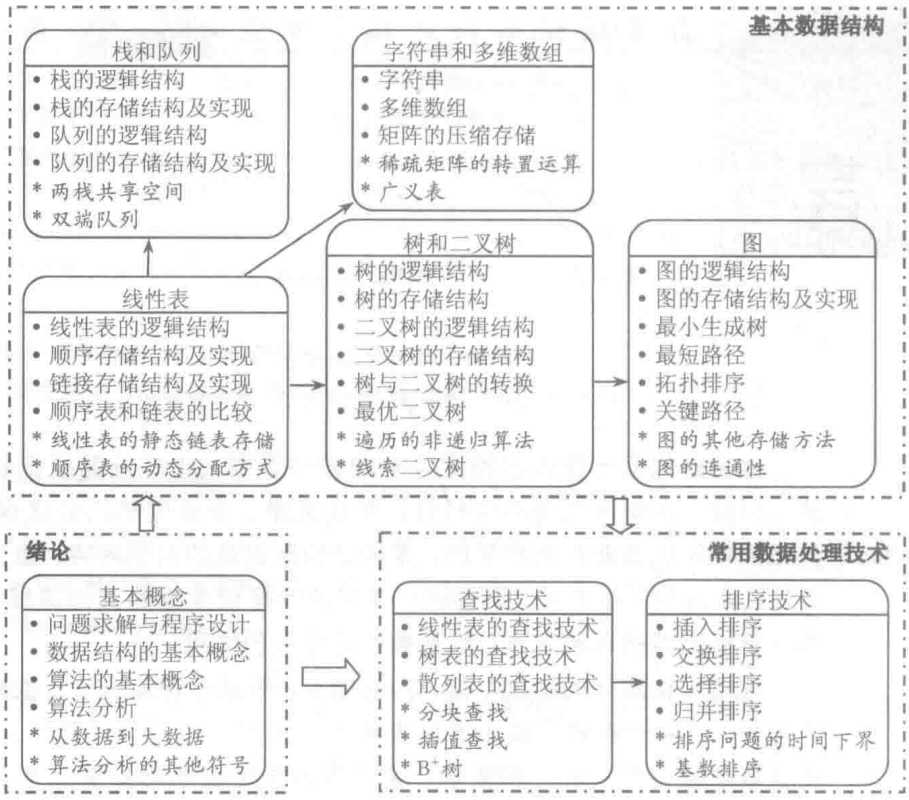


图 1　数据结构课程的知识单元及拓扑结构

④ 兼顾概念层和实现层。从抽象数据类型的角度将数据结构的实现过程分为抽象层、设计层和实现层，其中抽象层定义数据结构和基本操作集合，设计层是数据结构的存储表示和算法设计，实现层用C++语言实现数据结构，既强调了数据结构的基本概念和原理方法，又注重了数据结构的程序实现和实际运用。

⑤ 展现求解过程，培养计算思维。通过讲思路、讲过程、讲方法，按照“问题→想法→算法→程序”的模式进行问题求解，采用“阐述基本思想→伪代码描述算法→C++语言实现算法”的模式进行算法设计与实现，这个过程正是计算思维的运用过程。每章通过两个应用实例展示了问题求解过程以及算法设计过程。

⑥ 明确重点，化解难点。每一章开篇即给出该章的重点难点，以及各知识点的教学要求，在具体阐述时提供针对性的处理方法。针对数据结构内容抽象的特点，全书设计了大量插图，将抽象内容进行了具体化处理，降低了理解问题的复杂性。

总之，本书在概念的描述、实例的选择、知识的前后衔接、内容的组织结构，以及教学内容的理解、教学目标的实现、教学意图的融入、教学方法的运用等方面进行了系统思考和统筹设计，力图通过本书为读者构建多层次的知识体系。在问题求解层面，以数据表示和数据处理为主线，给出“问题→想法→算法→程序”的思维模式；在算法设计层面，通过伪代码描述算法，强调计算思维的培养；在算法分析层面，理解什么是“好”算法，给出算法分析的基本方法；在存储结构层面，通过存储示意图理解数据表示，再给出存储结构定义；在程序实现层面，给出所有数据结构的C++程序实现以及使

试读结束：需要全本请在线购买：www.ertongbook.com

用范例；在数据结构和算法的运用层面，通过应用实例理解如何为求解问题设计适当的数据结构，如何基于数据结构设计算法，从而将数据结构、算法和程序设计有机地融合在一起。

参加本书编写的还有王涛、党源源、肖巍、刘冰等，2015级本科生陈圣、陈甲旺、房子龙同学参与了本书的代码调试工作，所有代码均在 Dev-C++ 5.8.1、Code::Blocks 16.01、C-free 5.0编程环境下调试通过。本书采用 C++ 11 标准，具体变化是：①用 nullptr 取代 NULL；②定义模板时，用 typename 取代 class；③定义对象变量时，用花括号{}取代圆括号()。本书程序源代码均以二维码形式提供，使用前请阅读 readme（二维码）。

由于作者的知识和写作水平有限，书稿虽再三斟酌几经修改，仍难免有缺点，欢迎专家和读者批评指正。作者的电子邮箱是：wanghm@ ccut.edu.cn。

作　者

2019年2月

目录

CONTENTS

试读结束：需要全本请在线购买： www.ertongbook.com

第1章 绪 论

CHAPTER 1

本章概述	用计算机求解任何问题都离不开程序设计，程序设计的关键是数据表示和数据处理。数据要能被计算机处理，首先必须能够存储在计算机的内存中，这项任务称为数据表示，其核心是数据结构；一个实际问题的求解必须满足各项处理要求，这项任务称为数据处理，其核心是算法，数据结构课程讨论数据表示和数据处理的基本思想和方法。 本章通过一个实例的求解过程给出程序设计的一般过程，说明数据结构和算法在程序设计中的作用，介绍数据结构和算法的基本概念，说明用大 O 记号进行算法分析的基本方法。				
教学重点	数据结构的基本概念；数据的逻辑结构、存储结构以及二者之间的关系；算法及算法的特性；大 O 记号				
教学难点	抽象数据类型；算法的时间复杂度分析				
教学内容和教学目标	知 识 点	教 学 要 求			
		了解	理解	掌握	熟练掌握
	程序设计的一般过程	√			
	数据结构在程序设计中的作用		√		
	算法在程序设计中的作用		√		
	数据结构的基本概念				√
	抽象数据类型		√		
	算法及算法的特性				√
	算法的描述方法			√	
	算法的时间复杂度			√	
	算法的空间复杂度		√		

1.1 问题求解与程序设计

计算机科学致力于研究用计算机求解人类生产生活中的各种实际问题，只有最终在计算机上能够运行良好的程序才能解决特定的实际问题，因此，程序设计的过程就是利用计算机求解问题的过程。

1.1.1 程序设计的一般过程

用计算机求解任何问题都离不开程序设计，但是计算机不能分析问题并产生问题的解决方案，必须由人（即程序设计者）分析问题，确定问题的解决方案，采用计算机能够理解的指令描述这个问题的求解步骤（即编写程序），然后让计算机执行程序最终获得问题的解。用计算机求解问题（即程序设计）的一般过程[①]如图1-1所示。

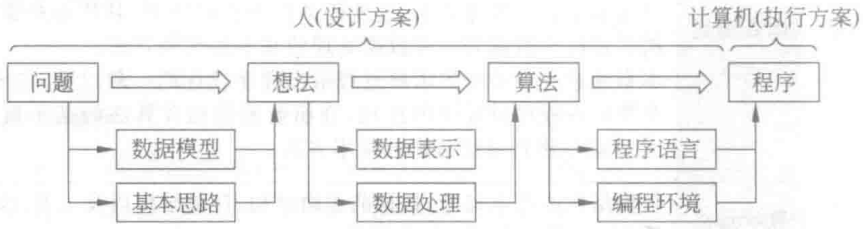


图1-1 程序设计的一般过程

由问题到想法需要分析待处理的数据以及数据之间的关系，抽象出具体的数据模型并形成问题求解的基本思路。对于数值问题抽象出的数据模型通常是数学方程，对于非数值问题抽象出的数据模型通常是表、树、图等数据结构。

算法用来描述问题的解决方案，是具体的、机械的操作步骤。利用计算机解决问题的最重要一步是将人的想法描述成算法，也就是从计算机的角度设想计算机是如何一步一步完成这个任务的。由想法到算法需要完成数据表示和数据处理，即描述问题的数据模型，将数据模型从机外表示转换为机内表示；描述问题求解的基本思路，将问题的解决方案形成算法。

由算法到程序需要将算法的操作步骤转换为某种程序设计语言对应的语句，转换所依据的规则就是某种程序设计语言的语法。换言之，就是在某种编程环境下用程序设计语言描述要处理的数据以及数据处理的过程。

著名的计算机学者尼古拉斯·沃思[②]给出了一个公式：算法＋数据结构＝程序。从

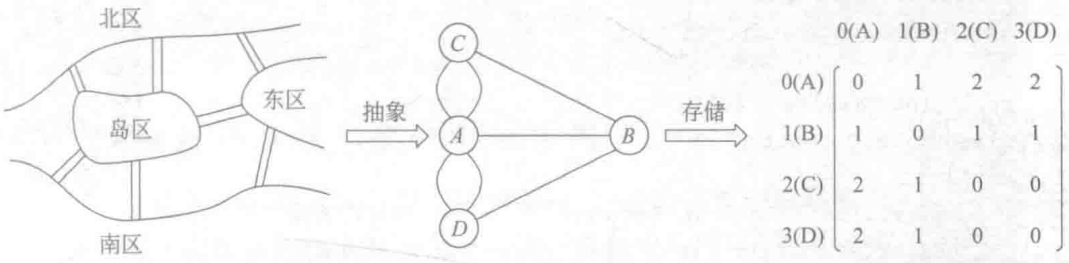
---

① 在作者编著的算法与程序设计系列教材《程序设计基础——从问题到程序》《数据结构——从概念到C实现》《数据结构——从概念到C++实现》《数据结构——从概念到Java实现》《算法设计与分析》中，都是通过这个问题求解过程培养学生的计算思维能力和程序设计能力。

② 尼古拉斯·沃思（Niklaus Wirth）1934年生于瑞士。1968年设计并实现了Pascal语言，1971年提出了结构化程序设计，1976年设计并实现了Modula 2语言。除了程序设计语言之外，沃思在其他方面也有许多创造，如扩充了著名的巴科斯范式，发明了语法图等。1984年获图灵奖。

这个公式可以看到，数据结构和算法是构成程序的两个重要的组成部分，一个“好”程序首先是将问题抽象出一个适当的数据模型并转换为机内表示，然后基于该数据结构设计一个“好”算法，或者说，学习数据结构的意义在于编写高质量、高效率的程序。下面以著名的哥尼斯堡七桥问题为例，说明程序设计的一般过程。

【问题】 哥尼斯堡七桥问题(以下简称“七桥问题”)。17 世纪的东普鲁士有一座哥尼斯堡城(现在叫加里宁格勒，在波罗的海南岸)。城中有一座岛，普雷格尔河的两条支流环绕其旁，并将整个城市分成北区、东区、南区和岛区 4 个区域，全城共有七座桥将 4 个城区连接起来，如图 1-2(a)所示。于是，产生了一个有趣的问题：一个人是否能在一次步行中经过全部的七座桥后回到出发点，且每座桥只经过一次。



(a) 七桥问题　　(b) 七桥问题的数据模型　　(c) 对应的二维数组mat[4][4]

图 1-2　七桥问题的数据抽象过程

【想法】 将城区抽象为顶点，用 A、B、C、D 表示 4 个城区。将桥抽象为边，用 7 条边表示七座桥。抽象出七桥问题的数据模型如图 1-2(b)所示，从而将七桥问题抽象为一个数学问题：求经过图中每条边一次且仅一次的回路，后来人们称为欧拉回路。欧拉回路的判定规则是：

① 如果没有一个城区通奇数桥，则无论从哪里出发都能找到欧拉回路；

② 如果通奇数桥的城区多于两个，则不存在欧拉回路；

③ 如果只有两个城区通奇数桥，则不存在欧拉回路，但可以从这两个城区之一出发找到欧拉路径(不要求回到出发点)。

由上述判定规则得到求解七桥问题的基本思路：依次计算与每个顶点相关联的边数，根据边数为奇数的顶点个数判定是否存在欧拉回路。

【算法】 将顶点 A、B、C、D 编号为 0、1、2、3，用二维数组 mat[4][4]存储七桥问题的数据模型，如果顶点 $i(0 \leqslant i \leqslant 3)$和顶点 $j(0 \leqslant j \leqslant 3)$之间有 k 条边，则元素 mat[i][j]的值为 k，如图 1-2(c)所示。求解七桥问题的关键是求与每个顶点相关联的边数，即在二维数组 mat[4][4]中求每一行元素之和，算法描述如下：

```
算法：oddVertexNum
输入：二维数组 mat[][],顶点个数 n
输出：通奇数桥的顶点个数 count
    1. count 初始化为 0;
    2. 下标 i 从 0~n -1 重复执行下述操作:
       2.1  计算第 i 行元素之和 degree;
```

```
        2.2  如果 degree 为奇数,则 count++;
    3. 返回 count;
```

【程序】 将函数 oddVertexNum 定义为类 EulerCircuit 的成员函数，类 EulerCircuit 的成员变量表示七桥问题对应的数据模型。主函数首先定义对象变量 G，然后调用函数 oddVertexNum 计算图模型中通奇数桥的顶点个数，再根据欧拉规则判定是否存在欧拉回路。程序如下：


用前必读


源代码

```
#include <iostream>
using namespace std;

const int MaxSize = 4;
class EulerCircuit {
public:
    EulerCircuit(int **a, int n);                    //构造函数
    ~EulerCircuit();                                 //析构函数
    int oddVertexNum();                              //求图中度为奇数的顶点个数
private:
    int mat[MaxSize][MaxSize];                       //二维数组存储图
    int vertexNum;                                   //顶点个数
};

EulerCircuit :: EulerCircuit(int **a, int n)
{
    for(int i = 0; i < n; i++)
      for (int j = 0; j < n; j++)
        mat[i][j]= * ((int *)a + n * i + j); //取元素 a[i][j]
    vertexNum = n;
}
EulerCircuit :: ~EulerCircuit()
{
}
int EulerCircuit :: oddVertexNum()
{
    int count = 0, i, j, degree;
    for (i = 0; i < vertexNum; i++)                  //依次累加每一行元素
    {
        degree = 0;                                  //记录通过顶点 i 的边数
        for (j = 0; j < vertexNum; j++)
            degree = degree + mat[i][j];
        if (degree % 2 != 0) count++;
    }
    return count;
```

```
}

int main()
{
    int a[4][4] = {{0, 1, 2, 2},{1, 0, 1, 1},{2, 1, 0, 0},{2, 1, 0, 0}};
    EulerCircuit G(a, 4);
    int num = G.oddVertexNum();                    //得到通奇数桥的顶点个数
    if (num >= 2)                                  //两个以上的顶点通奇数桥
        cout <<num <<"个地方通奇数桥,不存在欧拉回路" <<endl;
    else                                           //没有顶点通奇数桥
        cout <<"存在欧拉回路" <<endl;
    return 0;
}
```

1.1.2　数据结构在程序设计中的作用

在冯·诺依曼[①]体系结构下，程序的原始输入、中间结果和最终输出都以数据的形式存储在计算机的内存中，因此，数据的组织（求解问题的第一步就是将实际问题抽象为合适的数据模型）和存储（将数据模型从机外表示转换为机内表示）将直接影响和决定程序的效率。请看下面两个例子。

【例 1-1】 握手问题。Smith 先生和太太邀请 4 对夫妻来参加晚宴。每个人来的时候，房间里的一些人都要和其他人握手。当然，每个人都不会和自己的配偶握手，也不会和同一个人握手两次。之后，Smith 先生问每个人和别人握了几次手，他们的答案都不一样。问题是，Smith 太太和别人握了几次手？

【想法——数据模型】 这个问题具有挑战性的原因是没有一个明显的思考点，如果将此问题抽象出一个合适的数据模型，则问题会变得豁然开朗。

首先将每个人抽象为一个结点，如图 1-3(a)所示。其次考虑每个结点的权值，Smith 先生问了房间的 9 个人，每个人的答案都不相同，因此，每个答案都在 0 和 8 之间，相应地修改模型如图 1-3(b)所示。最后根据握手信息建立结点之间的边，8 号除了他（她）自己和配偶，与房间里的其他人总共握了 8 次手，基于这个观察，可以画出 8 号的握手信息并且知道 8 号的配偶是 0 号，如图 1-3(c)所示；7 号除了 0 号和 1 号，与房间里的其他人总共握了 7 次手，因此 7 号的配偶是 1 号……问题是不是变得豁然开朗了？

【例 1-2】 电话号码查询问题。假设某手机中存储了若干电话号码，如何查找某个人的电话号码？

【想法 1——数据模型】 将电话号码集合线性排列（即组织成线性结构），如表 1-1 所示，则查找某个人的电话号码只能进行顺序查找。

---

① 冯·诺依曼（Von Neumann）1903 年出生于匈牙利布达佩斯。19 岁就发表了有影响的数学论文，曾游学柏林大学，成为德国大数学家希尔伯特的得意门生。1933 年受聘于美国普林斯顿大学高等研究院，成为爱因斯坦最年轻的同事。冯·诺依曼在数学、应用数学、物理学、博弈论和数值分析等领域都有不凡建树，为计算机的逻辑设计奠定了坚实的基础。

试读结束：需要全本请在线购买：　www.ertongbook.com