

21世纪高等学校计算机**基础**实用规划教材

# 数据结构 (C语言版)

肖宏启 编著



清华大学出版社

21世纪高等学校计算机**基础**实用规划教材

# 数据结构 (C语言版)

肖宏启 编著

清华大学出版社  
北京

## 内 容 简 介

本书对常用的数据结构做了系统介绍,既注重原理又强调实践,配有大量的图表和习题,概念讲解清晰、逻辑性强、可读性好。主要内容包括:数据结构的基本概念;算法描述和算法分析初步;线性表、堆栈、队列、串、数组、树、图等结构;排序和查找的各种方法;另外还用一章的篇幅详细介绍了链式存储结构以加深读者的理解。每一章均列举了典型应用实例,并配有算法和程序以供教学和实践使用。

本书可作为高等院校应用型本科、专科及高等职业院校计算机类专业数据结构课程的教材,也可以作为大学非计算机专业学生的选修课教材和计算机应用技术人员自学参考书。

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: C 语言版/肖宏启编著.--北京:清华大学出版社,2016

21世纪高等学校计算机基础实用规划教材

ISBN 978-7-302-43352-1

I. ①数… II. ①肖… III. ①数据结构 ②C语言—程序设计 IV. ①TP311.12 ②TP312

中国版本图书馆 CIP 数据核字(2016)第 062795 号

责任编辑:黄 芝 李 晔

封面设计:何凤霞

责任校对:李建庄

责任印制:沈 露

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](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:20

字 数:498千字

版 次:2016年9月第1版

印 次:2016年9月第1次印刷

印 数:1~2000

定 价:39.50元

# 出版说明

---

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和教学方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生的知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程可以有针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制定教材规划时依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机基础实用规划教材  
联系人:魏江江 weijj@tup.tsinghua.edu.cn

# 前 言

---

随着社会经济的高速发展,我国的高等教育已步入从精英教育走向大众化教育的发展阶段。国际高等教育的历史说明高等职业教育必将成为教育发展的一种趋势。对于我国这种从专科转型的高等职业教育,现今还处于探索阶段,如何做好教材建设更是需要研究的重要方面。

高等职业教育是高等教育的一个新的类型,它与传统的普通高等教育既有紧密的联系,又有本质的区别,高等职业教育强调面向社会、生产、管理、服务第一线,培养技术应用型人才,大学生毕业后,即可发挥其所学专长。因此,根据我国教育部规定,高等职业教育需根据其自身的特点,建立自己的教材体系。

本书列入“21世纪高等学校计算机基础实用规划教材”,主要面向应用型本科及高等院校计算机类专业的学生,教材内容的构造力求体现“以应用为主体”,强调理论知识的理解和运用,实现高校应用型本科及高等职业教育教学以实践体系为主及以技术应用能力培养为主的目标,符合现代高等职业教育对教材的需求。

“数据结构”是计算机程序设计的重要理论基础,是计算机及其应用专业的一门重要基础课程和核心课程。它不仅是学习后继软件专业课程的先导,而且已成为其他工科类专业的热门选修课程。

本书共分9章。第1章阐述数据、数据结构和算法等基本概念。第2~7章分别讨论了线性表、链表、栈和队列、串和广义表、树和二叉树以及图等基本数据结构及其应用,其中,第3章专门总结了链式存储结构的基本概念和应用,为学好后面各类数据结构打好扎实的基础。第8~9章讨论查找和排序的各种实现方法和实用分析。

本书对大量抽象、难懂的概念进行了深入浅出的分析和讲解。

长期以来,由于数据结构课程自身的抽象性和严密性,教师大都感觉数据结构课程难教,学生普遍反映数据结构课程难学,学生很难独立完成算法的实现。基于上述问题,我们在编写本教材时充分考虑了学生的知识结构和教师的教学方法,既注重原理又注重实践,既注重抽象思维又注重形象思维,既方便自学又方便教学。

本教材的特点有:

(1) 对基础理论知识的阐述由浅入深、通俗易懂。内容组织和编排以应用为主线,略去了一些理论推导和数学证明的过程,淡化算法的设计分析和复杂的时空分析。

(2) 各章都列举并分析了很多实用的例子,这有助于学生加深对基础理论知识的理解和实际应用的能力培养。

(3) 考虑到此课程的先导课程是“C语言程序设计”,书中算法均采用可在计算机上运行的C语言程序来描述。这样,降低了算法设计的难度,使学生能更直观形象地理解这些

算法。鉴于微软对于 VC++ 6.0 早已停止维护升级,对于出现的不兼容问题已不再解决,本书中提供的实现代码均在简单易用的 Dev-C++ 平台上编译通过,并给出了所有程序的运行结果。若教师教学采用 TC 环境,只需在相应实现代码中加上相关头文件即可(Dev-C++ 平台安装包及使用说明书在教学资源包中下载)。

(4) 为配合本教材的使用,还编制了多媒体课件,对加深理解基本概念具有更直观的效果。多媒体课件、书中所有算法及上机实训源代码和习题答案可在清华大学出版社网站下载,或通过 E-mail 向肖宏启老师索取: xiaohongqi2000@163.com。

(5) 在教材中使用“▲思考”标志,提出问题拓展学生思维。在教学中恰到好处地启发学生的思维。

(6) 为避免 C 语言中数组的第一个元素的下标为 0 给学习和讲授带来的不便,本书在没有特别申明的地方均不使用 C 语言中数组下标为 0 的元素。

本书由肖宏启整体构思,在多位教师长期从事数据结构课程教学的经验基础上,经多次反复磋商和共同讨论定稿,是多位作者共同合作的产物。魏怀明副教授编写了第 5 章并详细审阅了全书,陈美成副教授、韦军博士审阅了该书并提供了许多宝贵的意见,陈元春副教授提供了本书的大部分习题。本书的第 2 章和第 4 章由陈锐编写,第 6 章由廖银花编写,第 7 章及附录部分由刘昌明编写,其余章节均由肖宏启编写。全书由肖宏启统稿、修改。本书编写过程中参考了许多作者的大量文献资料和国内外优秀教材,清华大学出版社的编辑对本书的出版给予了大力支持和帮助,作者谨此一并致以诚挚的谢意。

本教材讲课时数可为 60~72 学时。上机时数可灵活安排。教师可根据学时数、专业和学生的实际情况选讲应用举例及分析一些较难的例子。

由于编写教材的时间紧张,难免存在疏漏,敬请读者批评指正。

作者

2016 年 3 月

# 目 录

---

<b>第 1 章 绪论</b> .....	1
1.1 什么是数据结构 .....	1
1.1.1 数据结构起源 .....	1
1.1.2 数据结构研究的内容 .....	3
1.1.3 学习数据结构的必要性 .....	5
1.1.4 如何学好数据结构 .....	5
1.2 数据的逻辑结构 .....	6
1.2.1 基本概念 .....	6
1.2.2 逻辑结构的描述 .....	7
1.3 数据的存储结构 .....	8
1.4 算法和算法分析 .....	9
1.4.1 算法特性 .....	9
1.4.2 影响算法效率的因素 .....	10
1.4.3 算法效率的评价 .....	11
上机实训 .....	14
习题 .....	14
<b>第 2 章 线性表</b> .....	18
2.1 内存静态分配 .....	18
2.2 线性表的定义与运算 .....	18
2.2.1 线性表的定义 .....	18
2.2.2 线性表的基本操作 .....	19
2.3 线性表的顺序存储结构 .....	20
2.3.1 顺序表 .....	20
2.3.2 顺序表上基本运算的实现 .....	22
2.4 应用举例及分析 .....	28
上机实训 .....	29
习题 .....	30
<b>第 3 章 链表</b> .....	32
3.1 内存动态分配 .....	32



3.1.1	函数 malloc()	32
3.1.2	函数 free()	36
3.2	线性表的链式存储	37
3.2.1	线性链表	37
3.2.2	单链表上的基本运算	39
3.2.3	循环链表	53
3.2.4	双向链表	54
3.3	顺序表和链表的比较	57
3.4	应用举例及分析	58
	上机实训	65
	习题	66
<b>第4章</b>	<b>栈和队列</b>	<b>68</b>
4.1	栈	68
4.1.1	栈的定义及基本操作	68
4.1.2	栈的顺序存储结构	69
4.1.3	栈的链式存储结构	76
4.2	队列	80
4.2.1	队列的定义及基本操作	80
4.2.2	队列的顺序存储结构	81
4.2.3	队列的链式存储结构	89
4.3	应用举例及分析	95
	上机实训	101
	习题	104
<b>第5章</b>	<b>串和广义表</b>	<b>109</b>
5.1	串的定义和基本运算	109
5.1.1	串的定义	109
5.1.2	串的基本运算	110
5.2	串表示和实现	111
5.2.1	定长顺序存储	111
5.2.2	链式存储	111
5.2.3	堆分配存储	113
5.3	串基本运算的实现	115
5.4	广义表	126
5.4.1	广义表的定义和性质	126
5.4.2	广义表的存储	127
5.5	应用举例及分析	130
	上机实训	133

习题	133
<b>第 6 章 树和二叉树</b>	<b>136</b>
6.1 树的定义和术语	136
6.1.1 树的定义	136
6.1.2 基本术语	137
6.1.3 树的存储	138
6.2 二叉树	143
6.2.1 二叉树的定义	143
6.2.2 二叉树的性质	144
6.2.3 二叉树的存储	146
6.2.4 二叉树的建立	148
6.3 遍历二叉树	150
6.3.1 遍历二叉树	150
6.3.2 恢复二叉树	155
6.4 树、森林与二叉树的转换	157
6.4.1 一般树转换为二叉树	157
6.4.2 森林转换为二叉树	158
6.4.3 二叉树转换为树和森林	158
6.5 二叉树的应用举例	160
6.6 哈夫曼树及其应用	162
6.6.1 哈夫曼树的引入	162
6.6.2 哈夫曼树的建立	165
6.6.3 哈夫曼编码	166
上机实训	170
习题	171
<b>第 7 章 图</b>	<b>176</b>
7.1 图的定义和术语	176
7.1.1 图的定义	176
7.1.2 图的相关术语	177
7.1.3 图的基本操作	180
7.2 图的存储表示	181
7.2.1 邻接矩阵	181
7.2.2 邻接表	184
7.2.3 边集数组	187
7.3 图的遍历	188
7.3.1 深度优先搜索	189
7.3.2 广度优先搜索	194

7.4	生成树和最小生成树 .....	199
7.4.1	生成树和生成森林 .....	199
7.4.2	最小生成树 .....	201
7.4.3	普里姆(Prim)算法 .....	201
7.4.4	克鲁斯卡尔(Kruskal)算法 .....	205
7.5	单源最短路径 .....	209
7.5.1	单源最短路径的概念 .....	209
7.5.2	求单源最短路径的方法 .....	210
7.6	AOV网与拓扑排序 .....	214
7.6.1	AOV网与拓扑排序 .....	214
7.6.2	拓扑排序的实现 .....	215
7.7	AOE网与关键路径 .....	219
7.7.1	概述 .....	219
7.7.2	关键路径的确定 .....	219
	上机实训 .....	220
	习题 .....	222
<b>第8章</b>	<b>查找</b> .....	<b>225</b>
8.1	查找的基本概念 .....	225
8.2	静态查找表 .....	227
8.2.1	顺序查找 .....	227
8.2.2	二分查找 .....	229
8.2.3	索引顺序查找 .....	234
8.3	动态查找表 .....	235
8.3.1	二叉排序树定义 .....	235
8.3.2	二叉排序树的插入和生成 .....	236
8.3.3	二叉排序树的删除 .....	238
8.3.4	二叉排序树上的查找 .....	241
8.4	散列表 .....	246
8.4.1	散列表与散列函数 .....	246
8.4.2	散列函数的构造方法 .....	248
8.4.3	处理冲突的方法 .....	250
8.4.4	散列表的查找及分析 .....	252
	上机实训 .....	256
	习题 .....	258
<b>第9章</b>	<b>排序</b> .....	<b>262</b>
9.1	基本概念 .....	262
9.2	插入排序 .....	263

9.2.1	直接插入排序	264
9.2.2	二分插入排序	267
9.2.3	希尔排序	269
9.3	交换排序	272
9.3.1	冒泡排序	272
9.3.2	快速排序	276
9.4	选择排序	280
9.4.1	简单选择排序	280
9.4.2	堆排序	283
9.5	归并排序	288
9.6	各种排序方法的比较	291
	上机实训	292
	习题	293
<b>附录</b>	<b>设计性上机实训</b>	297
	课程设计指导	297
	课程设计参考题目	298
	课程设计(实训)报告参考格式	302
<b>参考文献</b>		305

## 本章内容概要：

计算机科学是一门研究数据表示和数据处理的科学。计算机在发展初期,其应用范围是数值计算,所处理的数据都是整型、实型、布尔型等简单数据,并以此为加工对象进行数值型程序的设计。后来,随着电子技术的发展,计算机逐渐进入到商业、制造业等其他领域,从而广泛地应用于数据过程和过程控制。与此相对应,计算机处理的数据也不再是简单的数值,而是字符串、图像、图形、声音、视频等复杂的数据。数据是计算机化的信息,它是计算机可以直接处理的最基本和最重要的对象。无论是进行科学计算或数据处理、过程控制以及对文件的存储和检索及数据库技术等计算机应用领域中,都是对数据进行加工处理的过程。因此,要设计出一个结构好效率高的程序,必须研究数据的特性及数据间的相互关系及其对应的存储表示,数据结构就是一门研究这些问题的课程,并利用这些特性和关系设计出相应的算法和程序。本章将介绍数据结构和算法分析的基本概念。

## 1.1 什么是数据结构

### 1.1.1 数据结构起源

首先,我们用两个简单的 C 语言程序范例来说明数据结构与程序设计的关联性,表 1.1 是 10 次 C 语言课程的测验成绩,请读者先思考一下,然后设计一个小程序计算这 10 次测验的总分和平均分。

表 1.1 测验成绩

测 验	成 绩
1	81
2	90
3	80
4	59
5	70
6	85
7	92
8	84
9	100
10	78

## 【程序 1-1】

```

/* ===== */
/*   程序: 1-1.c                               */
/*   计算总分和平均分                         */
/* ===== */

void main()
{
    int t1,t2,t3,t4,t5;                        /* 各次的成绩 */
    int t6,t7,t8,t9,t10;
    int sum;                                   /* 总分      */
    int average;                               /* 平均分    */

    t1 = 81; t2 = 90; t3 = 80; t4 = 59; t5 = 70;
    t6 = 85; t7 = 92; t8 = 84; t9 = 100; t10 = 78;
    sum = t1 + t2 + t3 + t4 + t5 + t6 + t7 + t8 + t9 + t10; /* 计算总分 */
    average = sum / 10;                               /* 计算平均分 */
    printf("输出总分: %d\n", sum);                    /* 输出总分 */
    printf("输出平均: %d\n", average);                /* 输出平均分 */
}

```

程序运行结果:

```

输出总分:819
输出平均:81

```

这是一个很简单的 C 语言程序,相信并不需要多费笔墨来解释。如果读者想到的方法是程序 1-1,必须很抱歉地建议您,需要先加强一下 C 语言的程序设计功力。不过也不必因此而感到难过或自责,因为大多数人都要经历这么一段学习程序设计的过程。

程序 1-1 使用数个内存变量存储考试成绩。这种方法的扩充性不是很好,因为如果考试的次数改变了,增加成为 15 次或减少成为 8 次,整个程序就需要很大的修改。事实上,从测验次数和成绩的关系可得知,用数组结构保存测验的成绩是一种更好的方法。程序 1-2 中正是使用这种方法。

## 【程序 1-2】

```

/* ===== */
/*   程序: 1-2.c                               */
/*   计算总分和平均分                         */
/* ===== */

void main()
{
    int t[10] = { 81,90,80,59,70,85,92,84,100,78 };
    int sum;                                   /* 总分      */
    int average;                               /* 平均分    */
}

```

```

int i;

sum = 0;                                /* 设置总分初值 */
for ( i = 0; i < 10; i++)
    sum += t[i];                          /* 计算总分 */
average = sum / 10;                       /* 计算平均分 */
printf("输出总分: %d\n", sum);           /* 输出总分 */
printf("输出平均: %d\n", average);      /* 输出平均分 */
}

```

程序运行结果:

```

输出总分:819
输出平均:81

```

相信读者在比较程序 1-1 和程序 1-2 后,一定可以发现程序 1-2 使用的方法比较好。可是为什么读者会认为程序 1-2 比较好?是经验吗?事实上,这是因为程序 1-2 使用了比较好的数据结构来解决问题。虽然这两个程序都可以正确地解决问题。但是采用不同的方式保存成绩数据,进而形成不同的程序设计方式。所以如何选择最佳的数据结构来解决程序问题,这就是为何需要学习数据结构的重要原因。

数据结构这门课程的起源是程序设计的经验累积,自从计算机(电脑)发明以来,科技发展一日千里,计算机知识随处可得,目前的一般读者根本无法理解早期程序设计师的辛酸。程序设计对于早期的程序设计师来说,是一种艺术而不是技术。因为参考的数据难得,每一位程序设计师在追求计算机知识的过程中,都曾经经历过一段非常艰苦的时光。逐渐地,由这些先辈所留下来宝贵的程序设计经验能够确实有效地解决一些程序问题,而这些方法便成为了一门学问,这就是“数据结构(Data Structure)”这门课程的起源。

### 1.1.2 数据结构研究的内容

早期人们都把计算机理解为数值计算工具,使用计算机的目的主要是处理数值计算问题。当我们使用计算机来解决一个具体问题时,一般需要经过下列几个步骤:首先要从该具体问题抽象出一个适当的数学模型,然后设计或选择一个解此数学模型的算法,最后编制程序、运行并调试程序,直到实际问题被解决。

由于当时所涉及的运算对象是简单的整型、实型或布尔类型数据,所以程序设计者的主要精力是集中于程序设计的技巧上,而无须重视数据结构。随着计算机软、硬件的发展和应用领域的不断扩大,计算机处理的对象更多地是非数值计算问题,这类问题涉及的数据结构更为复杂,数据元素之间的相互关系一般无法用数学方程式加以描述,此时解决这类问题的关键不再是数学分析和计算方法,而是必须建立相应的数据结构来进行描述,分析问题中所用到的数据是如何组织的,研究数据之间的关系如何,进而设计出合适的数据结构,才能有效地解决问题。下面所列举的就是属于这一类的具体问题,用以说明什么是数据结构和数据结构所研究的内容。

例 1.1 某班级学生学籍档案的管理,如表 1.2 所示。

表 1.2 某班级学生学籍档案信息

学号	姓名	性别	年龄	入学成绩
2009001	张三	男	21	80
2009002	刘薇薇	女	23	90
2009003	马琳	男	22	78
⋮	⋮	⋮	⋮	⋮
2009050	李明	男	22	85

在这个班级学生学籍档案信息表中,共有 50 个学生。我们可以把表中每个学生的信息看成一条记录并称之为一个节点,表中的每个节点由 5 个数据项组成。该学生学籍档案信息表由 50 个节点组成,每个节点排列有先后次序,形成一种线性关系。这是一种典型的数据结构,我们称这种数据结构为线性表。

对该表的主要操作有:在给出学号时,如何在表中快速查找到所对应的学生的信息;若有学生退学,如何删除该学生的记录;若有新生入学该班级时,如何在表中插入一条新记录。这些都是数据结构要研究的内容。

例 1.2 航天学院教学行政机构示意图,如图 1.1 所示。

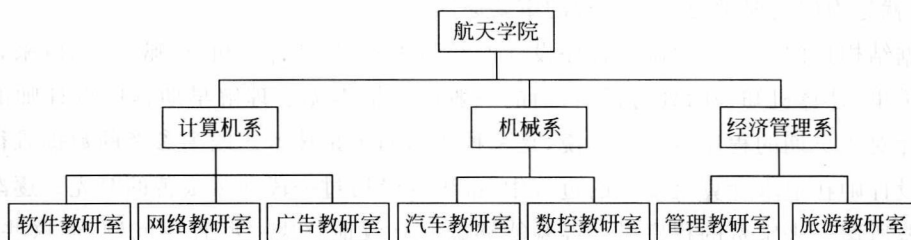


图 1.1 航天学院教学行政机构示意图

对于航天学院的教学行政机构,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中间节点,把每个系的教研室看成树叶。树中的每个节点可以包含较多的信息,节点之间的关系不再是顺序的,而是分层、分叉的一对多的非线性结构。这也是常用的一种数据结构,我们称之为树形结构,如图 1.1 所示。树形结构的主要操作有遍历、查找、插入和删除等。

例 1.3 城市之间建立通信网络的问题,如图 1.2 所示。

在  $n$  个城市之间建立通信网络,要求在任意两个城市之间都有直接或间接的通信线路;在已知某些城市之间直接通信线路预算造价的情况下,使网络造价最低。

当  $n$  很大时,这样的问题只能用计算机来求解。我们用图 1.2(a)中描述的关系来说明:图中的每个小圆圈表示一个城市,两个圆圈之间的连线表示对应城市之间的通信线路,连线上的数值表示该通信线路的造价。这一描述结构为图状结构,利用计算机可以求出满足要求的最小造价通信网络,如图 1.2(b)所示。在如图 1.2(a)所示这种数据结构中,数据之间的关系是多对多的非线性关系,我们称这种数据结构为图形结构。

由以上三个例子可见,描述这类非数值计算问题的数学模型不再是数学方程,而是诸如



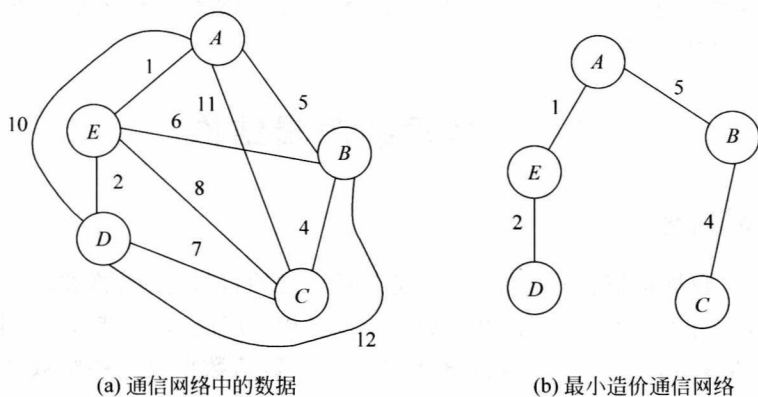


图 1.2 用图描述通信网络问题

表、树、图之类的数据结构。因此,可以说数据结构课程主要是研究非数值计算的程序设计问题中所出现的计算机操作对象以及它们之间的关系和操作的学科。

学习数据结构的目的是为了了解计算机处理对象的特性,将实际问题中所涉及的处理对象在计算机中表示出来并对它们进行处理。与此同时,通过算法训练来提高我们的思维能力,通过程序设计的技能训练来促进我们的综合应用能力和专业素质的提高。

### 1.1.3 学习数据结构的必要性

随着计算机运行速度的加快和存储(内存)容量的不断加大,有的人可能认为程序的运行效率变得越来越不重要了。然而,计算机的功能越强大,人们就越想去尝试解决更加复杂的问题。而更复杂的问题需要更大的计算量,这使得对高效率的程序的需求更加明显,工作越复杂就越偏离人们的日常经验。因此,学习数据结构,掌握各种前人设计的算法的运行效率并自己设计高效率的算法是非常必要的。

数据结构不仅是计算机专业教学计划中的核心课程之一,而且已逐步成为非计算机专业的主要选修课程之一。数据结构与数学、计算机硬件和软件的关系十分密切,是介于数学、计算机硬件和计算机软件之间的一门核心课程。在计算机科学中,数据结构不仅是一般非数值计算程序设计的基础,而且是设计和实现汇编语言、编译程序、操作系统、数据库系统以及其他系统程序和大型应用程序的重要基础。打好数据结构课程的扎实基础,对于学习计算机专业其他课程,如编译原理、数据库系统原理、计算机网络基础等都十分有益。

### 1.1.4 如何学好数据结构

本课程(以本书为教材)的先导课程是“C 语言程序设计”。要想学好本课程,首先要求读者已经掌握了基本的 C 语言知识,掌握了模块化程序设计的基本思想,能够利用 C 语言熟练编写一些简单的程序。同时要求读者比较熟悉 C 语言编程环境,能够熟练地编辑、调试及运行 C 语言程序。

在学习本课程时,读者首先要掌握各种基本的数据结构,并对各种数据结构的逻辑特性和物理特性(存储结构)都要有足够的认识。对基于各种数据结构的常见操作及其算法要重点掌握,并要了解评价某个具体算法优劣的方法。