



普通高等教育“十一五”国家级规划教材



21世纪大学本科
计算机专业系列教材

刘明军 潘玉奇 主编
蔺永政 袁 宁 编著

程序设计基础(C语言)(第2版)

<http://www.tup.com.cn>

- 国家精品课程主讲教材
- 根据教育部“高等学校计算机科学与技术专业规范”组织编写
- 与美国 ACM 和 IEEE CS Computing Curricula 最新进展同步



清华大学出版社



普通高等教育“十一五”国家级规划教材

21世纪大学本科计算机专业系列教材

国家精品课程主讲教材

程序设计基础 (C语言)

(第2版)

刘明军 潘玉奇 主编

蔺永政 袁 宁 编著



清华大学出版社

北京

TP312C-43
463-2

内 容 简 介

本书以培养编程能力为出发点,以实用性为目标,比较全面地介绍C语言程序设计的基本知识和程序设计的基本方法。全书分为10章,内容涵盖C语言的全部知识点。首先介绍程序设计的基本概念、C语言基础与程序结构,给学习者一个全面的程序概念;然后逐步展开介绍C语言基本内容和程序设计方法;最后给出程序设计实例。

本书是作者多年从事C语言教学的经验积累,可以作为高等学校本科、高职高专“C语言程序设计”课程的教学用书,也可以作为C语言程序设计的自学用书。

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

程序设计基础: C语言/刘明军,潘玉奇主编;蔺永政,袁宁编著. --2版. --北京:清华大学出版社,2014

21世纪大学本科计算机专业系列教材

ISBN 978-7-302-36391-0

I. ①程… II. ①刘… ②潘… ③蔺… ④袁… III. ①C语言—程序设计—高等学校—教材
IV. ①TP312

中国版本图书馆CIP数据核字(2014)第098975号

责任编辑:张瑞庆

封面设计:常雪影

责任校对:焦丽丽

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:三河市中晟雅豪印务有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:24.25

字 数:591千字

版 次:2010年9月第1版 2014年8月第2版

印 次:2014年8月第1次印刷

印 数:1~2000

定 价:39.00元

产品编号:058888-01

21 世纪大学本科计算机专业系列教材编委会

主 任：李晓明

副 主 任：蒋宗礼 卢先和

委 员：(按姓氏笔画为序)

马华东	马殿富	王志英	王晓东	宁 洪
刘 辰	孙茂松	李仁发	李文新	杨 波
吴朝晖	何炎祥	宋方敏	张 莉	金 海
周兴社	孟祥旭	袁晓洁	钱乐秋	黄国兴
曾 明	廖明宏			

秘 书：张瑞庆

责任编辑：李晓明

前言

FOREWORD

C语言是计算机程序设计语言的主流语种。30多年来,C语言不断地发展和完善,逐步成为公认的一种优秀的程序设计语言,有着其他语言不可比拟的优点。

目前C语言教材主要分为两类。一类是以C语法为中心,在介绍语法的基础上,结合程序设计巩固C语言的语法知识。这类教材强调的是语法教学和C语言知识的掌握,而不是C语言的程序设计能力。另一类是案例教材,通过案例学习,兼顾语法教学,通过模仿学习程序设计。相对前一类,这类教材语法教学系统性相对欠缺。本教材是在总结我们建设“C语言程序设计”国家精品课程过程中的经验,认真研究该课程的特点,分析当前出版的C语言程序设计教材的基础上,编写完成的。

作为程序设计教材应该重点培养学生的程序设计能力,同时应该让学生掌握扎实的语法知识。学生创新能力的培养是潜移默化的,作为教材应该在学生创新能力方面加以引导,培养学生发现问题、分析问题和解决问题的能力。

本教材的主要特点如下。

(1) 强化程序设计能力培养。

本教材从实际问题需求出发引出理论,从个体到一般,以点带面。根据程序设计的需要,引出相关的知识点,将知识学习和使用密切结合,加深了理解,也避免了枯燥的学用分离的语法学习,使学习者明确为什么引出这些知识点,强化了知识点在程序设计中的应用。

(2) 注重学生创新思维的培养。

本教材贯穿了提出需要解决的问题、分析问题、引出概念、讲解知识点、程序实现的编写思路。通过给出实际问题,分析问题的特点,引导学生思考,然后给出解决问题的思路。通过潜移默化的作用,培养学生的创新思维和分析问题、解决问题的能力。

(3) 突出实用性和趣味性。

在例题的选择上力求实用性和趣味性,以此提高学生的应用程序设计的能力和学习兴趣。内容的组织编排强化实践教学,突出编程能力培养。所有例题不是简单地给出程序,而是首先分析问题,提出解题思路,再给出解决方案。将算法和数据结构结合起来,培养学生的编程能力。

(4) 强调学用结合和规范化编程

学习的目的是为了使用。因此,知识点的学习紧密结合使用,知识点基本上采用了学了即用的原则。一方面加强了知识点的理解和巩固,另一方面知道这些知识点在什么地方用和如何用。避免为了学习而学习,学而不用的问题。努力引导学生养成良好的编程习惯,编写风格优美、可读性好、易于维护的程序代码。

2.4.1 C语言的运算符

编写一本精品教材绝非易事,尽管我们力图贯彻突出程序设计能力的培养和启迪创新思维的思想,但是由于水平有限,还有许多不尽如人意的地方。另外,在编写过程中,由于时间紧迫,难免存在这样或那样的问题和不足,敬请同行和读者提出宝贵意见,以便我们在以后的版本中改进。

本教材由济南大学C语言课程组组织,参加编写的有刘明军、潘玉奇、蔺永政、袁宁等。董吉文教授及课程组的其他老师在教材编写过程中提出了一些很好的建议,在此表示感谢。

北京大学的李晓明教授审阅了全书,提出了非常中肯和宝贵的意见。对全书的定稿给予了很大帮助。

本书配备完整的教学课件和案例源代码,有需要的老师可登录课程网站下载,课程网站网址为 <http://c.ujn.edu.cn>;也可从清华大学出版社网站下载,网址为 <http://www.tup.com.cn>。作者联系邮箱为 ise_panyq@ujn.edu.cn。

作者

2014年3月于济南



CONTENTS

第1章 程序设计概述	1
1.1 计算机软件	1
1.1.1 程序设计语言	1
1.1.2 程序设计	2
1.1.3 数据结构	3
1.1.4 算法	3
1.2 算法的表示方法	5
1.2.1 自然语言表示法	5
1.2.2 流程图表示法	6
1.2.3 N-S图表示法	7
1.2.4 伪代码表示法	8
1.3 程序设计方法	9
1.3.1 引言	9
1.3.2 结构化程序设计	9
1.3.3 面向对象的程序设计	10
1.3.4 面向服务的程序设计	10
1.4 小结	11
第2章 C语言基础	12
2.1 C语言的发展历程	12
2.2 C程序的特点及开发环境	13
2.2.1 C程序的组成及特点	13
2.2.2 C程序的风格	14
2.2.3 C程序的开发环境	17
2.3 C语言的基本标识	18
2.3.1 C语言基本语法成分	18
2.3.2 C语言数据类型	20
2.4 C语言的运算符与表达式	28
2.4.1 C语言的运算符	28



2.4.2	C 语言的表达式	29
2.4.3	算术运算符与算术表达式	30
2.4.4	赋值运算符与赋值表达式	32
2.4.5	逗号运算符与逗号表达式	35
2.5	C 语句和 C 程序结构	35
2.5.1	C 语句	35
2.5.2	C 程序结构	37
2.6	数据的输入、输出与简单程序设计	39
2.6.1	字符数据的输入输出	40
2.6.2	数据的格式化输入输出	40
2.6.3	简单的 C 程序设计	48
2.7	编译预处理命令	49
2.7.1	文件包含	50
2.7.2	宏定义	52
2.7.3	条件编译	56
2.7.4	编译优化	58
2.8	汇编与链接	59
2.8.1	汇编	59
2.8.2	链接	59
2.9	小结	60
第 3 章 程序的控制结构		61
3.1	关系运算与逻辑运算	61
3.1.1	关系运算	61
3.1.2	逻辑运算	62
3.2	分支结构	64
3.2.1	单分支结构	64
3.2.2	双分支结构	65
3.2.3	多分支结构	66
3.2.4	if 语句的嵌套	71
3.2.5	条件运算符	73
3.3	循环结构	75
3.3.1	循环的引出	75
3.3.2	while 循环	76
3.3.3	do-while 循环	77
3.3.4	for 循环	79
3.3.5	几种循环的比较	84
3.4	break 和 continue 语句	86
3.4.1	break 语句	86

3.4.2	continue 语句	88
3.5	goto 语句	89
3.6	小结	90
3.7	程序举例	91
第4章 数组		99
4.1	一维数组	99
4.1.1	一维数组的引出	99
4.1.2	一维数组的定义与引用	100
4.1.3	一维数组的初始化	102
4.1.4	一维数组的应用	103
4.2	二维数组	104
4.2.1	二维数组的引出	104
4.2.2	二维数组的定义与引用	105
4.2.3	二维数组的初始化	107
4.2.4	二维数组的应用	108
4.3	字符数组	109
4.3.1	字符数组的引出	109
4.3.2	字符数组的定义和使用	110
4.3.3	字符串	111
4.3.4	字符数组的应用	118
4.4	小结	120
4.5	程序举例	121
第5章 函数		128
5.1	函数的引出	128
5.2	函数定义与调用	130
5.2.1	函数的定义与调用	130
5.2.2	函数声明与函数原型	134
5.3	函数参数传递	136
5.3.1	简单变量作为函数参数	136
5.3.2	数组作为函数参数	137
5.3.3	函数定义与带参数的宏定义的区别	144
5.4	函数的嵌套调用	145
5.5	递归与分治算法	146
5.5.1	递归函数	146
5.5.2	分治算法	151
5.6	局部变量与全局变量	155

5.6.1	局部变量	155
5.6.2	全局变量	156
5.7	变量的存储类别	160
5.7.1	auto 变量	160
5.7.2	static 变量	160
5.7.3	register 变量	162
5.7.4	extern 变量	163
5.8	内部函数与外部函数	165
5.8.1	内部函数	165
5.8.2	外部函数	166
5.9	小结	167
5.10	程序举例	167
第6章	指针	172
6.1	指针定义与使用	172
6.1.1	指针的引出	172
6.1.2	指针变量的定义	175
6.1.3	指针变量的使用	175
6.2	指针与函数	179
6.2.1	指针作为函数参数	179
6.2.2	函数返回指针	182
6.2.3	指向函数的指针	184
6.3	指针与数组	185
6.3.1	一维数组与指针	185
6.3.2	数组名作为函数参数	188
6.3.3	二维数组与指针	192
6.4	指针与字符串	195
6.5	指针数组与多级指针	199
6.5.1	指针数组的定义和引用	199
6.5.2	多级指针	200
*6.5.3	指针数组作为 main 函数的形参	201
6.6	指针与动态内存分配	205
6.6.1	内存管理	205
6.6.2	void 类型指针	206
6.6.3	动态内存分配和释放函数	206
*6.7	指针的深层应用	208

6.7.1	指针访问特定内存区域	208
6.7.2	指针类型的强制转换	209
6.7.3	指针的安全问题	209
6.8	小结	210
6.9	程序举例	211
第7章 结构体与链表		216
7.1	结构体的引出	216
7.2	结构体变量	218
7.2.1	结构体变量的定义	218
7.2.2	结构体变量的引用和初始化	220
7.3	结构体数组	223
7.3.1	结构体数组的定义	223
7.3.2	结构体数组的初始化	224
7.3.3	结构体数组的使用	224
7.4	结构体类型的指针变量	226
7.4.1	指向结构体变量的指针	226
7.4.2	指向结构体数组的指针	228
7.5	结构体与函数	228
7.5.1	结构体变量作为函数参数	228
7.5.2	指向结构体变量的指针作为函数参数	229
7.5.3	函数返回值为结构体类型	232
7.6	链表	233
7.6.1	链表引出	233
7.6.2	链表的建立	234
7.6.3	链表的输出	235
7.6.4	链表的删除操作	237
7.6.5	链表的插入操作	239
7.7	共用体和枚举类型	241
7.7.1	共用体	241
7.7.2	枚举类型	244
7.8	类型定义符 typedef 的用法	247
7.9	小结	248
7.10	程序举例	248
第8章 文件		254
8.1	文件概述	254
8.1.1	文件的分类	254

8.1.2	文件类型指针	255
8.1.3	文件操作的基本步骤	256
8.2	文件的打开与关闭	257
8.2.1	文件打开函数	257
8.2.2	文件关闭函数	258
8.3	文件的读写	259
8.3.1	字符读写函数	259
8.3.2	字符串读写函数	261
8.3.3	数据块读写函数	263
8.3.4	格式化读写函数	266
8.4	文件的定位	267
8.4.1	复位函数	267
8.4.2	随机移动函数	268
8.4.3	取当前位置的函数	270
8.5	文件检测函数	271
8.5.1	feof 函数	271
8.5.2	ferror 函数	274
8.5.3	clearerr 函数	274
8.6	小结	275
8.7	程序举例	275
第9章	位运算	279
9.1	位运算符	279
9.2	位运算符的运算规则	279
9.2.1	按位与运算符	279
9.2.2	按位或运算符	281
9.2.3	按位异或运算符	281
9.2.4	按位取反运算符	283
9.2.5	左移运算符	283
9.2.6	右移运算符	283
9.2.7	位运算中的类型转换	284
9.2.8	位运算的复合赋值运算符	285
9.3	小结	285
9.4	程序举例	285
第10章	综合程序设计	288
10.1	电子万年历系统	288
10.1.1	系统设计要求	288

10.1.2	系统总体设计	288
10.1.3	源程序代码	292
10.1.4	程序运行结果	296
10.2	集合基本运算系统	298
10.2.1	系统设计要求	298
10.2.2	系统总体设计	298
10.2.3	源程序代码	302
10.2.4	程序运行结果	312
10.3	图书借阅管理系统	316
10.3.1	系统设计要求	316
10.3.2	系统总体设计	317
10.3.3	源程序代码	322
10.3.4	程序运行结果	344
附录 A	ASCII 码表	352
附录 B	C 运算符的优先级与结合性	355
附录 C	常用标准库函数	357
C.1	stdio. h 中的常用函数	357
C.2	math. h 中的常用函数	361
C.3	stdlib. h 中的常用函数	363
C.4	string. h 中的常用函数	366
C.5	ctype. h 中的常用函数	368
C.6	conio. h 中的常用函数	370
C.7	time. h 中的常用函数	371
参考文献		373

第1章

程序设计概述

程序(program)是计算任务的处理对象和处理规则的描述。任何以计算机为处理工具的任务都是计算任务。处理对象是数据(如数字、文字、图形、图像和声音等,它们只是表示,而无含义)或信息(数据及有关的含义)。处理规则一般指处理动作和步骤。程序设计(programming)是设计、编制和调试程序的方法与过程。程序设计是软件构造活动中的重要组成部分,程序设计往往以某种程序设计语言为工具,程序设计是一项创造性的工作,是伴随着计算机应用和程序设计语言的发展而发展起来的一门学科。对于不同的计算机语言,其程序设计的基本方法是大致相同的。本书以C语言程序设计为主线,介绍程序设计的基本概念和基本方法。

本章主要内容:简单介绍计算机软件、程序设计语言的分类,重点讨论程序设计的方法步骤、数据结构的概念、算法及其描述。介绍程序设计的一般方法,介绍结构化程序设计、面向对象程序设计以及面向服务程序设计的特点。通过本章的学习,对程序设计的基本概念和方法有基本了解,了解算法和数据结构的概念,掌握算法的表示方法。

1.1 计算机软件

计算机软件(computer software)是计算机系统上的程序及其文档。程序是计算任务的处理对象和处理规则的描述;文档是为了便于了解程序所需要的简明资料。

1.1.1 程序设计语言

软件语言(software language)是用于书写计算机软件的语言。它主要包括需求定义语言、功能性语言、设计性语言、实现性语言以及文档语言等。

程序设计语言(programming language)是用于书写计算机程序(习惯上指实现高级语言程序)的语言。语言的基础是一组记号和一组规则。根据规则由记号构成的记号串的总体就是语言。在程序设计中,这些记号串就是程序。程序设计语言的基本成分如下。

- (1) 数据成分:用于描述程序所涉及的数据。
- (2) 运算成分:用于描述程序中所包含的运算。
- (3) 控制成分:用于表达程序中的控制构造。
- (4) 传输成分:用于表达程序中数据的传输。

按照语言与硬件的关联程度不同,有低级语言和高级语言之分。计算机程序设计语言

的发展,经历了从机器语言、汇编语言到高级语言的历程,大量的应用程序开发通常采用高级语言,机器语言和汇编语言通常用于与机器硬件关联密切的程序设计。

1. 机器语言

机器语言(machine language)是用二进制代码表示的计算机能直接识别和执行的一种机器指令的集合。它是计算机的设计者通过计算机的硬件结构赋予计算机的操作功能。机器语言具有灵活、直接执行和速度快等特点。不同型号的计算机的机器语言是不相通的,按着一种计算机的机器指令编制的程序,一般情况下不能在另一种计算机上执行。

用机器语言编写程序,编程人员要首先熟记所用计算机的全部指令代码和代码的涵义,需要自己处理每条指令和每一数据的存储分配和输入输出,需要记住编程过程中每步所使用的工作单元处在何种状态。机器语言程序全是0和1的指令代码,程序的编写、调试和修改非常复杂。

2. 汇编语言

汇编语言(assembly language)是面向机器的程序设计语言。在汇编语言中,用助记符代替操作码,用地址符号或标号代替地址码,方便记忆和使用。使用汇编语言编写的程序,机器不能直接识别,需要将汇编语言翻译成机器语言,这种起翻译作用的程序称为汇编程序,把汇编程序翻译成机器语言的过程称为汇编。

汇编语言是面向机器的低级语言,保持了机器语言的优点,具有直接和简洁的特点,可有效地访问、控制计算机的各种硬件设备,目标代码简短,占用内存少,执行速度快。

3. 高级语言

高级语言(high-level language)是一种更接近于自然语言的计算机程序设计语言,具有很强的描述能力,高级语言不直接依赖于具体的计算机硬件。使用高级语言编写的程序不能直接运行,需要将其转换为机器语言才能运行,通常的转换方式有解释和编译两种。

在计算机发展历史上,先后出现过几百种高级语言。其中,影响较大、使用较普遍的有FORTRAN、ALGOL、COBOL、BASIC、Lisp、PL/1、Pascal、C、PROLOG、Ada、C++、Visual C++、Visual Basic、Java等高级语言。

1.1.2 程序设计

程序设计是目标明确的智力活动,是根据计算机要完成的任务,提出相应的要求,在此基础上设计数据结构和算法,然后再编写相应的程序代码并测试该代码运行的正确性,直到能够得到正确的运行结果为止。

由于程序是软件的本体,软件的质量主要通过程序的质量来体现,在软件研究中,程序设计的工作非常重要,内容涉及有关的基本概念、工具、方法及方法学等。程序设计一般分为以下5个步骤。

(1) 方案确定:程序将以数据处理的方式解决客观世界中的问题,因此进行程序设计首先应该将实际问题用数学语言描述出来,形成一个抽象的、具有一般性的数学问题。也就是建立问题的抽象数学模型,然后制定解决该模型所代表的数学问题的算法。数学模型精确地阐述了模型本身所涉及的各种概念、已知条件、所求结果,以及已知条件与所求结果之间的联系等各方面的信息。数学模型是进一步确定解决所代表的数学问题算法的基础。利用数学模型和算法的结合得到问题的解决方案。

(2) 算法描述: 具体的解决方案确定后, 需要对所采用的算法进行描述, 算法的初步描述可以采用自然语言方式、程序流程图、N-S图、伪代码等。算法描述应简单明确, 能够比较明显地展示程序设计思想, 是进行程序调试的重要参考。

(3) 数据描述: 根据程序设计的目标及对数据的处理要求, 确定所处理数据的表示方式, 即数据结构。算法和数据结构密切相关, 两者应相互结合。

(4) 编写程序: 使用计算机系统提供的某种程序设计语言, 根据上述算法描述和数据结构, 将已设计好的算法表达出来, 使得非形式化的算法转变为形式化的由程序设计语言表达的算法, 这个过程称为程序编制(编码)。程序的编写过程需要反复调试才能得到可以运行且结果“正确”的程序。

(5) 程序测试: 程序编写完成后必须经过科学的、严格的测试, 才能最大限度地保证程序的正确性。同时, 通过测试可以对程序的性能作出评估。

程序设计是很讲究方法的, 一个好的设计方法能够大大提高程序的高效性、合理性。程序设计是软件开发工作的重要部分, 而软件开发是工程性的工作, 所以要有规范。

1.1.3 数据结构

数据结构(data structure)是计算机存储、组织数据的方式。数据结构是指相互之间存在一种或多种特定关系的数据元素的集合。通常情况下, 精心选择的数据结构可以带来更高的运行速度或者存储效率。数据结构往往与高效的检索算法和索引技术有关。数据结构一般包括以下3个方面内容。

(1) 数据元素之间的逻辑关系, 又称数据的逻辑结构(logical structure)。数据的逻辑结构是从逻辑关系上描述数据, 与数据的存储无关, 是独立于计算机的。数据的逻辑结构可以看作是从具体问题抽象出来的数学模型。

(2) 数据元素及其关系在计算机存储器内的表示, 称为数据的存储结构(storage structure)。数据的存储结构是逻辑结构用计算机语言的实现(也称为映像), 它依赖于计算机语言。对机器语言而言, 存储结构是具体的。通常, 只在高级语言的层次上讨论存储结构。

(3) 数据的运算, 即对数据施加的操作。数据的运算定义在数据的逻辑结构上, 每种逻辑结构都有一个运算的集合。最常用的检索、插入、删除、更新、排序等运算, 实际上只是在抽象的数据上所施加的一系列抽象的操作。所谓抽象的操作, 是指只知道这些操作是“做什么”, 而无须考虑“如何做”。只有确定了存储结构之后, 才考虑如何具体实现这些运算。

1.1.4 算法

在客观世界中, 做任何事情都有一定的方法和步骤。比如, 要得到某门课程的学分, 就包括选课、听课、完成作业、参加考试等环节。如果考试不及格, 还要按规定补考或重修。这就是获得课程学分的方法步骤。用计算机解决问题也是按照相应的步骤一步一步完成的。这些方法步骤都可以称为算法(algorithm)。广义地说, 算法是为解决问题而采取的方法和步骤。

1. 算法的概念

在程序设计中, 算法是一系列解决问题的清晰指令。也就是说, 能够对一定规范的输

人,在有限时间内获得所要求的输出。算法可以理解为由基本运算及规定的运算顺序所构成的完整的解题步骤。或者看成按照要求设计好的、有限的、确切的计算序列,并且这样的步骤和序列可以解决一类问题。如果一个算法有缺陷或者不适合于某个问题,执行这个算法将不会解决这个问题。不同的问题有不同的算法,不同的算法可能用不同的时间、空间或效率来完成同样的任务。一个算法的优劣可以用空间复杂度与时间复杂度来衡量。

2. 算法的特征

一个算法应该具有以下5个重要的特征。

- (1) 有穷性:一个算法必须保证执行有限步之后结束。
- (2) 确切性:算法的每一步骤必须有确切的定义。
- (3) 可行性:算法原则上能够精确地运行,而且人们用笔和纸做有限次运算后即可完成。

(4) 输入:一个算法有0个或多个输入,以刻画运算对象的初始情况,所谓0个输入是指算法本身定出了初始条件。

(5) 输出:一个算法有一个或多个输出,以反映对输入数据加工后的结果。没有输出的算法是毫无意义的。

计算机科学家尼克劳斯·沃思曾写过一本著名的书《数据结构+算法=程序》,可见算法在计算机科学界与计算机应用界的地位。

【例 1.1】 对给定有序数列{3,5,11,17,21,23,28,30,32,50},查找关键字值为30的数据元素。

算法一:将30按顺序与给定数列逐一地比较,直到找到为止。这种算法最坏的情况是可能需要比较整个序列。

算法二:查找过程中采用跳跃式方式查找,即先以有序数列的中点位置为比较对象,如果要找的元素值小于该中点元素,则将待查序列缩小为左半部分,否则为右半部分。通过一次比较,将查找区间缩小一半。经 $\log_2 n$ 次比较就可以完成查找过程。但是这种算法的缺点也很明显,要求查找数列必须有序,而对所有数据元素按大小排序是非常费时的操作。

可见,不同的算法其效率有着很大的差别,不同的算法适合的场合也不同。

3. 算法设计与分析的常用方法

常用的算法设计与分析方法如下。

(1) 递推法:递推法把问题分成若干步,找出相邻几步的关系,从而达到目的的方法称为递推法。它是利用问题本身所具有的一种递推关系来求解问题,是迭代算法的最基本形式。

(2) 递归法:递归指的是一个过程,即函数不断引用自身,直到引用的对象已知。

(3) 穷举搜索法:穷举搜索法是对可能是解的众多候选解,按某种顺序进行逐一枚举和检验,并从中找出那些符合要求的候选解作为问题的解。

(4) 贪婪法:贪婪法是一种不追求最优解,只希望得到较为满意解的方法。贪婪法一般可以快速地得到满意的解,因为它省去了为找最优解要穷尽所有可能而必须耗费的大量时间。贪婪法常以当前情况为基础作最优选择,而不考虑各种可能的整体情况,所以贪婪法不要回溯。

(5) 分治法:把一个复杂的问题分成两个或更多的相同或相似的子问题,再把子问题