

普通高校本科计算机专业特色教材精选·算法与程序设计

算法设计方法与优化

滕国文 等 编著

清华大学出版社

普通高校本科计算机专业特色教材精选·算法与程序设计

算法设计方法与优化

滕国文 丛飏 李闯 张伟 宫耀勤 滕泰 编著



清华大学出版社
北京

内 容 简 介

本书简要介绍了算法设计、分析和优化的基础知识,重点放在算法设计方法的学习上,通过大量的典型例题讲解了常用算法设计方法(共10种):求值法、累加法、累乘法、递推法、递归法、枚举法、分治法、贪心法、回溯法和动态规划法,最后通过实例给出算法设计综合应用。每个例题从问题描述、问题分析、算法说明、算法设计、运行结果和算法优化6个方面进行讲解。

本书结合实例、内容丰富、深入浅出、结构清晰,可以作为高等院校计算机及其相关专业本科生和研究生算法设计课程的教材,也适合IT技术人员和计算机编程爱好者学习参考。

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

算法设计方法与优化 / 滕国文等编著. —北京: 清华大学出版社, 2013

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 978-7-302-33201-5

I. ①算… II. ①滕… III. ①电子计算机—算法设计—高等学校—教材 IV. ①TP301.6

中国版本图书馆CIP数据核字(2013)第163189号

责任编辑:袁勤勇 薛 阳

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:李红英

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 刷 者: 北京富博印刷有限公司

装 订 者: 北京市密云县京文制本装订厂

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 16.75 字 数: 380千字

版 次: 2013年9月第1版 印 次: 2013年9月第1次印刷

印 数: 1~2000

定 价: 29.00元

产品编号: 052137-01

前言

PREFACE

一些著名的计算机科学家在有关计算机科学教育的论述中认为,计算机科学是一种创造性的思维活动,其教育必须面向设计。算法被公认为是计算机科学的基石。因此,算法设计这门课,在计算机科学与技术学科中占核心地位。对于计算机专业的学生,学会读懂算法、设计算法,应该是一项最基本的要求。通过对计算机算法系统的学习与研究,理解和掌握算法设计的主要方法,培养对算法优化和分析的能力,将为运用计算机解决实际问题奠定坚实的基础。

用计算机解决问题的核心是算法设计,算法设计的关键是要掌握一些常用的算法设计方法和抽象的计算思维方式。

在本书各章的讨论中,首先介绍一种算法设计方法的基本思想,然后将计算机经典问题和算法设计方法很好地结合起来,运用该算法设计方法去解决这些经典问题,并给出用C语言描述的具体算法。通过各种算法设计方法在求解不同问题中的应用进行比较,牢固掌握算法设计技术的基本策略;通过对不同的算法设计方法在同一问题上的应用进行比较,更容易体会到算法设计方法的思想,锻炼逻辑思维能力,达到融会贯通的效果。

全书共分为12章。第1章简要介绍了算法设计、分析和优化的基础知识,第2~11章系统讲解了10种常用的算法设计方法,分别为:求值法、累加法、累乘法、递推法、递归法、枚举法、分治法、贪心法、回溯法和动态规划法,第12章是算法设计综合应用。

本书的第1、12章由滕国文执笔;第2、3章由宫耀勤执笔;第4、5章由李闯执笔;第6、11章由滕泰执笔;第7、8章由丛飏执笔;第9、10章由张伟执笔。2010级学生(张天骥、徐悦、姜波、张丽、杨昌宇、周凯、李金刚、张梦琳、罗春龙、王琪、曹宇和李少军)和硕士研究生(滕硕、董亚群、刘洋、张菁、曾轩、肖春英和张雷)参加了部分代码编写和程序调试工作,夏凤琴、温毓铭和2012级部分学生参与了书稿的校对工作,作者谨此一并致以诚挚的谢意!全书由滕国文教授统稿、审阅和整理后定稿。

在本书的编写过程中,作者参阅并借鉴了国内外诸多同行的文章和著作,这里不一一列举、标明,在此向他们致以谢意!

由于作者水平有限,加之学科理论与技术发展日新月异,书中疏漏谬误之处在所难免,恳请广大读者指正。

作 者

2013年6月

目 录

CONTENTS

第 1 章 算法概述	1
1.1 算法与问题求解	1
1.1.1 算法的定义	1
1.1.2 问题求解	2
1.2 算法的要素和特性	3
1.2.1 算法的要素	3
1.2.2 算法的基本特性	4
1.3 算法的描述	5
1.3.1 基本控制结构的描述	5
1.3.2 C 算法描述约定	7
1.4 算法分析	9
1.4.1 算法的评价标准	9
1.4.2 算法的时间复杂度	10
1.4.3 算法的空间复杂度	11
1.5 算法的优化	12
1.5.1 全局优化	12
1.5.2 局部优化	13
1.5.3 算法优化中的注意事项	14
第 2 章 求值法	17
2.1 算法设计思想	17
2.2 典型例题	17
2.2.1 求最大值	17
2.2.2 求平均分	19
2.2.3 判断闰年	21
2.2.4 素数	23
2.2.5 孪生数	25

2.2.6 巧排螺旋阵	27
2.2.7 百灯判熄	31
2.3 小结	33
习题 2	33
第 3 章 累加法	35
3.1 算法设计思想	35
3.2 典型例题	35
3.2.1 自然数求和	35
3.2.2 自然数倒数求和	37
3.2.3 统计及格人数	38
3.2.4 计算 π 值	41
3.2.5 统计对称数	42
3.2.6 数列求和	45
3.2.7 判断天数	46
3.3 小结	48
习题 3	49
第 4 章 累乘法	51
4.1 算法设计思想	51
4.2 典型例题	51
4.2.1 求 n 的阶乘	51
4.2.2 计算偶数积	52
4.2.3 求阶乘之和	54
4.2.4 大整数阶乘	56
4.2.5 国王奖赏问题	58
4.2.6 计算 e 值	60
4.3 小结	62
习题 4	62
第 5 章 递推法	65
5.1 算法设计思想	65
5.2 典型例题	66
5.2.1 兔子繁殖问题	66
5.2.2 最大公约数问题	68
5.2.3 猴子吃桃问题	70
5.2.4 杨辉三角问题	71
5.2.5 穿越沙漠问题	74

5.2.6 分西瓜问题	75
5.3 小结	77
习题 5	77
第 6 章 递归法	81
6.1 算法设计思想	81
6.2 典型例题	81
6.2.1 母牛繁殖问题	81
6.2.2 输出各位数字	83
6.2.3 最大值问题	85
6.2.4 计算 x 的 n 次幂	87
6.2.5 数组逆置	89
6.2.6 汉诺塔问题	91
6.3 小结	93
习题 6	93
第 7 章 枚举法	97
7.1 算法设计思想	97
7.2 典型例题	97
7.2.1 百鸡问题	97
7.2.2 水仙花数	100
7.2.3 完数	102
7.2.4 可逆素数	104
7.2.5 串匹配问题	107
7.2.6 最小公倍数问题	111
7.2.7 狱吏问题	113
7.3 小结	116
习题 7	116
第 8 章 分治法	119
8.1 算法设计思想	119
8.2 典型例题	119
8.2.1 折半查找	119
8.2.2 金块问题	121
8.2.3 寻找第二的问题	124
8.2.4 归并排序	127
8.2.5 大整数乘法	131
8.2.6 二叉树遍历	133

8.3 小结	138
习题 8	138
第 9 章 贪心法	141
9.1 算法设计思想	141
9.2 典型例题	142
9.2.1 找零钱问题	142
9.2.2 最优装载	145
9.2.3 哈夫曼编码	149
9.2.4 单源最短路径	155
9.2.5 埃及分数问题	158
9.2.6 多机调度问题	162
9.3 小结	165
习题 9	166
第 10 章 回溯法	169
10.1 算法设计思想	169
10.2 典型例题	170
10.2.1 八皇后问题	170
10.2.2 图着色问题	174
10.2.3 桥本分数式	176
10.2.4 高逐位整除数	180
10.2.5 直尺刻度分布问题	183
10.2.6 素数环问题	186
10.2.7 伯努利装错信封问题	190
10.3 小结	194
习题 10	195
第 11 章 动态规划	197
11.1 算法设计思想	197
11.2 典型例题	198
11.2.1 数塔问题	198
11.2.2 矩阵连乘问题	202
11.2.3 最长公共子序列问题	207
11.2.4 最长上升子序列问题	211
11.2.5 陪审团问题	215
11.3 小结	218
习题 11	218

第 12 章 综合应用	223
12.1 Fibonacci 数列	223
12.2 π 值求法	227
12.3 循环赛日程表	230
12.4 最大子段和问题	239
12.5 背包问题	246
习题 12	256

第 1 章

算法概述

CHAPTER

1.1 算法与问题求解

用计算机解决实际问题时,最终要编制程序,根据著名的计算机专家沃思(Niklaus Wirth)提出的著名公式:

算法+数据结构=程序

可以看出,要编制程序,必须确定算法和数据结构。

1.1.1 算法的定义

算法就是为了求解问题而给出的指令序列,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,而程序是算法的一种实现。计算机按照程序逐步执行算法,实现对问题的求解。简单来说,算法可以看成是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某一个(类)问题。

算法设计的重点就是把人类找到的求解问题的方法、步骤,以过程化、形式化、机械化的形式表示出来,以便让计算机执行。

对于一个问题,如果可以通过一个计算机程序,在有限的存储空间内运行有限的时间而得到正确的结果,则称这个问题是算法可解的。但算法不等于程序,也不等于计算方法。当然,程序也可以作为算法的一种描述,但程序通常还需要考虑很多与方法和分析无关的细节问题,这是因为在编写程序时要受到计算机系统环境的限制。通常程序的编制不可能优于算法的设计。

算法是一个十分古老的研究课题,然而计算机的出现为这个课题注入了青春和活力,使算法的设计和分析成为计算机科学中最为活跃的研究热点之一。

1967年,D. E. Knuth指出“算法是贯穿在所有计算机程序设计中的一个基本概念”,所以,算法被誉为计算机学科的灵魂!

数学大师吴文俊指出：我国传统数学在从问题出发以解决问题为主旨的发展过程中建立了以构造性与机械化为其特色的算法体系，这与西方数学以欧几里得《几何原本》为代表的所谓公理化演绎体系正好遥遥相对……肇始于我国的这种机械化体系，在经过明代以来几百年的相对消沉后，由于计算机的出现，已越来越为数学家所认识与重视，势将重新登上历史舞台。吴文俊创立的几何定理的机器证明方法（世称吴方法），用现代的算法理论，使得中国古代传统算法发扬光大，享有很高的国际声誉，也受到大家的高度关注。

1.1.2 问题求解

用计算机解决实际问题，就是在计算机中建立一个解决问题的模型。在这个模型中，计算机内部的数据表示了需要被处理的实际对象，包括其内在的性质和关系；处理这些数据的程序；模拟对象领域中的求解过程。通过解释计算机程序的运行结果，便得到了实际问题的解。下面给出用计算机求解问题的一般步骤。

1. 问题分析

这个阶段的任务是弄清题目提供的已知信息和所要解决的问题是什么。完整地理解和描述问题是解决问题的关键。要做到这一点，必须注意以下一些问题：在未经加工的原始表达中，所用的术语是否都明白其准确定义？题目提供了哪些已知信息？还可以得到哪些潜在的信息？题目中做了哪些假定？题目要求得到什么结果？等等。针对每个具体问题，必须认真审查问题的有关描述，深入分析，以加深对问题的准确理解。

2. 数学模型的建立

用计算机解决实际问题必须建立合适的数学模型。因为在现实问题面前，计算机是无能为力的。对一个实际问题建立数学模型，可以考虑这样两个基本问题：最适合于此问题的数学模型是什么？是否有已经解决了的类似问题可以借鉴？

建立数学模型是最关键且较困难的一步，涉及四个世界和三级抽象。四个世界分别是现实世界（客观世界）、信息世界（概念世界）、数据世界、计算机世界。三级抽象分别是：现实世界到信息世界的抽象，建立信息模型或概念模型；信息世界到数据世界的抽象，将信息数据转化建立数据模型；数据世界到计算机世界的抽象，建立存储模型在计算机中实现。

3. 算法设计

算法设计是指设计求解某一特定类型问题的一系列步骤，并且这些步骤是可以通过计算机的基本操作来实现的。算法设计要同时结合数据结构的设计，简单来说数据结构的设计就是选取存储方式，因为不同的数据结构的设计将导致算法的差异很大。算法的设计与模型的选择更是密切相关的，但同一模型仍然可以有不同的算法，而且它们的有效性可能有相当大的差距。

算法设计方法也称算法设计技术或算法设计策略，是设计算法的一般性方法，可用于解决不同计算领域的多种问题。虽然设计算法尤其是设计出好的算法是一件非常困难的工作，但是设计算法也不是没有方法可循，人们经过几十年来的工作，总结和积累了许多行之有效的方法，了解和掌握这些方法会给我们解决问题提供一些思路。本书讨论的算法设计方法已经被证明是对算法设计非常实用的通用技术，包括求值法、累加法、累乘法、递推法、递归法、枚举法、分治法、贪心法、回溯法和动态规划法等。这些算法设计方法构

成了一组强有力的工具,可用于大量实际问题的求解。

4. 算法表示

对于复杂的问题,确定算法后可以选择一种算法描述方式来准确表示算法。算法的描述方式有很多,如传统流程图、盒图、PAD图、伪码和高级语言等,其中高级语言是最理想的描述算法的方法,因此,本书选择C语言来表示算法。

5. 算法分析

算法分析的目的首先是为了对算法的某些特定输入,估算该算法所需的内存空间和运行时间,其次是为了建立衡量算法优劣的标准,用以比较同一类问题的不同算法。一般来说,一个好的算法首先应该是比同类算法的时间效率高,算法的时间效率用时间复杂度来度量。

6. 算法实现

算法实现就是指编码,也就是平常所说的编程序,即将算法设计“转译”成某种计算机语言的表述形式,才能够在计算机上执行。编码的目的是使用选定的程序设计语言,把算法描述翻译成为用该语言编写的源程序(或源代码)。源程序应该正确可靠、简明清晰,而且具有较高的效率。

在把算法转变为程序的过程中,虽然现代编译器提供了代码优化功能,但是,仍然需要一些技巧,比如,在循环之外计算循环中的不变式、合并公共子表达式等。

7. 程序调试

程序调试也称为算法测试,其任务首先是发现和排除在前几个阶段中产生的错误,经测试通过的程序才可投入运行,在运行过程中还可能发现隐含的错误和问题,因此,还必须在使用中不断地维护和完善。

算法测试的实质是对算法应完成任务的实验证实,同时确定算法的使用范围。测试方法一般有两种:白盒测试,对算法的各个分支进行测试;黑盒测试,检验对给定的输入是否有指定的输出。

8. 结果整理文档编制

结果整理时,要对计算结果进行分析,看其是否符合实际问题的要求,如果符合,问题得到解决,可以结束;如果不符合,说明前面的步骤一定存在问题,必须返回,从头开始逐步检查,找出错误并重新设计,这个循环过程也可能重复多次。

编制文档的目的是让别人理解你编写的算法。首先要将代码编写清楚,代码本身就是文档,同时还有代码的注释。另外还包括算法的流程图,自顶向下各研制阶段的相关记录,算法的正确性证明(论述),算法测试过程、结果,对输入输出的要求及格式的详细描述等。

1.2 算法的要素和特性

1.2.1 算法的要素

算法由操作、控制结构和数据结构三要素组成。

1. 操作

算法实现平台尽管有许多种类,它们的函数库、类库也有较大差异,但是必须具备的

最基本的操作功能是相同的。这些操作包括以下几方面。

- (1) 算术运算：加法、减法、乘法、除法等运算。
- (2) 关系比较：大于、小于、等于、不等于等运算。
- (3) 逻辑运算：与、或、非等运算。
- (4) 数据传送：输入、输出、赋值等操作。

2. 控制结构

一个算法功能的实现不仅取决于所选用的操作，而且还与各操作之间的执行顺序有关。算法中各操作之间的执行顺序称为算法的控制结构。算法的控制结构给出了算法的基本框架，它不仅决定了算法中各操作的执行顺序，而且也直接反映了算法的设计是否符合结构化原则。

算法的基本控制结构有以下三种。

(1) 顺序结构：顺序结构是程序设计中最简单、最常用的基本结构。在该结构中，各操作块按照出现的先后顺序依次执行。它是任何程序的主体基本结构，即使在选择结构或循环结构中，也常以顺序结构作为其子结构。

(2) 选择结构：又称为分支结构，是指程序依据条件所列出表达式的结果来决定执行多个分支中的哪一个分支，进而改变程序执行的流程。依据条件选择分支的结构称为选择结构。

(3) 循环结构：某一类问题可能需要重复多次执行完全一样的计算和处理方法，而每次使用的数据都按照一定的规律在改变。这种可能重复执行多次的结构称为循环结构，又称重复结构。

3. 数据结构

算法操作的对象是数据，数据间的逻辑关系、数据的存储方式及处理方式就是数据结构。它与算法设计是紧密相关的。

有了计算机的帮助，使得许多过去靠人工无法计算的大量复杂问题有了解决的希望。不过，使用计算机进行计算，首先要解决的是如何将处理的对象存储到计算机中，也就是要选择适当的数据结构。

1.2.2 算法的基本特性

算法应具有以下 5 个重要特性。

(1) 输入：一个算法有 0 个或多个外部量作为算法的输入。有些输入量需要在算法执行过程中输入，而有些算法表面上可以没有输入，实际上已被嵌入算法之中。

(2) 输出：一个算法产生至少一个或多个量作为输出。它是一组与输入有确定关系的量值，是算法进行信息加工后得到的结果。

(3) 确定性：算法中的每一条指令必须有确切的含义，无二义性。即每种情况下所应执行的操作在算法中都有确切的规定，使算法的执行者或阅读者都能明确其含义及如何执行。在任何条件下，对于相同的输入只能得到相同的输出。

(4) 有穷性: 是指算法必须能在执行有限步骤后、有限的时间内终止。即每条指令的执行次数和执行时间必须是有限的。

(5) 可行性: 算法描述的操作可以通过已经实现的基本操作执行有限次来实现。就是指算法的每一个步骤, 计算机都能执行。计算机所能执行的动作, 是预先设计好的, 一旦出厂就不会改变。所以, 设计算法时, 应考虑每个步骤必须能用计算机所能执行的操作命令实现。

综上所述, 算法是一组严谨定义运算顺序的规则, 并且每一个规则都是有效的、明确的, 此规则将在有限的次数内终止。

1.3 算法的描述

算法设计者在构思和设计了一个算法之后, 必须清楚准确地将所设计的求解步骤记录下来, 即算法描述。常用的算法描述方法有自然语言、传统流程图、N-S 图、PAD 图、伪代码和高级语言等。本书使用高级语言中的 C 语言来描述算法。

下面首先给出三种基本控制结构的描述, 然后详细列出 C 语言算法描述的约定。

1.3.1 基本控制结构的描述

计算机科学家已经证明只需要使用三种基本控制结构就可以构建解决任何复杂问题的算法。这三种基本控制结构是: 顺序结构、选择结构和循环结构。

为了更好地领会和理解这三种基本控制结构, 下面将 C 语言与 N-S 图两种描述方法对照给出。并做如下约定: S(Statement), S1, S2 代表语句; E(Expression), E1, E2, E3 代表表达式; T(True) 代表逻辑“真”(成立); F(False) 代表逻辑“假”(不成立)。

1. 顺序结构

1) C 语言描述

```
S1  
S2
```

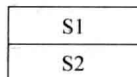


图 1-1

即 S1 语句在前, S2 语句在后。

2) N-S 图描述(图 1-1)

顺序结构是指程序的执行次序与程序的书写次序一致。即先执行 S1, 后执行 S2。

2. 选择结构

1) 简单选择结构

(1) C 语言描述

```
if ( E )  
    S
```

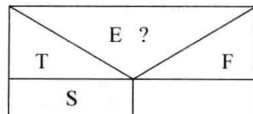


图 1-2

(2) N-S 图描述(图 1-2)

简单选择结构执行时, 先判断表达式 E, 若为非零(T, 成立), 则执行 S; 否则什么都

不做。

2) 一般选择结构

(1) C 语言描述

```
if ( E )
    S1
else
    S2
```

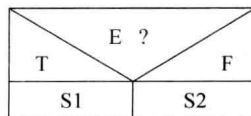


图 1-3

(2) N-S 图描述(图 1-3)

一般选择结构执行时,先判断表达式 E ,若为非零(T ,成立),则执行 $S1$;否则执行 $S2$ 。 $S1$ 与 $S2$ 只执行其一,也必执行其一。

3. 循环结构

1) while 循环

(1) C 语言描述

```
while ( E )
    S
```

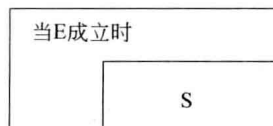


图 1-4

(2) N-S 图描述(图 1-4)

while 循环执行过程:首先判断表达式 E ,若为非零(成立),则执行语句 S ;执行语句 S 后,再返回表达式 E 的判断,如果仍为非零(成立),再次执行语句 S ;如此反复,直到某一次表达式 E 为零(不成立)为止,结束该循环。

2) do-while 循环

(1) C 语言描述

```
do
    S
while ( E );
```

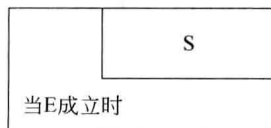


图 1-5

(2) N-S 图描述(图 1-5)

do-while 循环执行过程:首先执行一次语句 S ,然后判断表达式 E ,若为非零(成立)时,则再次执行语句 S ;执行语句 S 后,再返回表达式 E 的判断,如果仍为非零(成立),再次执行语句 S ;如此反复,直到某一次表达式 E 为零(不成立)为止,结束该循环。

3) for 循环

(1) C 语言描述

```
for ( E1; E2; E3 )
    S
```

(2) N-S 图描述(图 1-6)

for 循环执行过程:首先计算表达式 $E1$,之后判断表达式 $E2$,若为非零(成立),则执行语句 S ,并计算表达式 $E3$;然后再返回表达式 $E2$ 的判断,如果仍为非零(成立),再次执行



图 1-6

语句 S,并计算表达式 E3;如此反复,直到某一次表达式 E2 为零(不成立)为止,结束该循环。

1.3.2 C 算法描述约定

1. 算法表示形式

本书中所有的算法都以如下的 C 函数形式表示。

1) 定义格式

[函数返回值类型] 函数名 ([形式参数表列及说明])

```
{  
    声明部分  
    执行语句部分  
}
```

函数的定义主要由函数说明部分和函数体两部分组成,用花括号“{”和“}”括起来的部分为函数体,其前面为函数说明部分。函数中用方括号“[”和“]”括起来的部分为可选项,但函数名之后的圆括号“(”和“)”是不能省略的。函数通常用 return 语句将指定的值返回给调用者。

2) 调用格式

函数名 ([实际参数表列])

3) 形实参结合方式

形式参数与实际参数之间信息传递的主要方式有两种:一是值传递,即将实参表达式的值依次传递给对应的形式参数,而形参的改变不会影响到实参,通常称为单向的传值过程;二是地址传递,即将实参变量的地址依次传递给对应的形式参数,这时对应的形参和实参就具有了相同的地址,也就是说它们共享地址空间,因此,若形参改变,则实参必然随之改变。对此,有人说成是双向传递过程,通过上面的讲解,我们知道这种说法是不对的,或者说的不准确。其实,也是单向的传递过程!

2. 模块化设计

根据结构化程序设计的思想,将一个大任务分解成若干个功能独立的子任务。因此算法可由一个主函数和若干个其他函数组成。C 的函数相当于其他语言中的子程序,用函数来实现特定的功能。一个完整的、可执行的 C 程序文件一般结构如下:

```
[文件包含命令]  
[宏定义命令]  
[用户自定义类型]  
[所有子函数的原型说明]  
[子函数 1 定义]  
...  
[子函数 n 定义]  
[主函数定义]
```