



程序员 面试手册

概念、编程问题及面试题

CODING INTERVIEW QUESTIONS

[印] 纳拉辛哈·卡鲁曼希 (Narasimha Karumanchi) 著 爱飞翔 译

精选微软、谷歌和Facebook等世界知名IT公司面试和竞赛中涉及的编程题目
全面涵盖编程基础、架构设计、数据结构及算法等重要主题

针对各类竞赛和面试中的每个难点和考点提供了大量习题和详细解答



机械工业出版社
China Machine Press



程序员面试手册

概念、编程问题及面试题

CODING INTERVIEW QUESTIONS

[印] 纳拉辛哈·卡鲁曼希 (Narasimha Karumanchi) 著 爱飞翔 译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

程序员面试手册：概念、编程问题及面试题 / (印) 纳拉辛哈·卡鲁曼希 (Narasimha Karumanchi) 著；爱飞翔译. —北京：机械工业出版社，2018.3

书名原文：Coding Interview Questions

ISBN 978-7-111-59011-8

I. 程… II. ①纳… ②爱… III. 程序设计—手册 IV. TP311.1-62

中国版本图书馆 CIP 数据核字 (2018) 第 016716 号

本书版权登记号：图字 01-2016-0684

Translation from the English language edition:

Coding Interview Questions, by Narasimha Karumanchi (ISBN: 9788192107516).

Copyright © 2014 by CareerMonk.com.

All Rights reserved No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanic, including photocopying, recording, or by any information storage retrieval system, without permission of Brainy Software, Inc.

Chinese simplified language edition published by China Machine Press.

Copyright © 2018 by China Machine Press.

本书中文简体字版由 CareerMonk Publications 授权机械工业出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

程序员面试手册：概念、编程问题及面试题

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）

责任编辑：缪杰

责任校对：李秋荣

印刷：北京诚信伟业印刷有限公司

版次：2018 年 3 月第 1 版第 1 次印刷

开本：186mm×240mm 1/16

印张：44.25

书号：ISBN 978-7-111-59011-8

定价：99.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88379426 88361066

投稿热线：(010) 88379604

购书热线：(010) 68326294 88379649 68995259

读者信箱：hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问：北京大成律师事务所 韩光 / 邹晓东

华章IT
HZBOOKS | Information Technology



面试与技术考试的成绩受很多因素影响，其中关键的一项因素就是技术水平。这些知识虽然无法完全通过书本获得，但是好的参考书可以把它们有条理地呈现出来，令人尽快发现自己的不足，从而更加明确地去补习某些方面的知识，这有利于提升面试和考试的成绩，而且对实际工作也有所帮助。

Karumanchi 先生的这本书就是一本题目丰富、思路清晰而且解法多样的参考书。

书中的题目涵盖了编程基础、架构设计、网络技术、数据库技术、数据结构及算法等主要知识，而且还涉及了趣味谜题以及非技术的问题。其中数据结构与算法占据较大篇幅，因为这是面试与考试经常涉及的内容，而且能反映开发者的思考能力与技术水平。本书的每一章后面几乎都有数十个问题，多看一些这样的题目，在面对面面试提问的时候，就能够更加从容地应对。

与范围和数量相比，题目的质量同样比较高。有些题目能够指出容易被忽视或误解的知识，还有一些题目则与实际的应用有关，例如模拟图书馆的借书系统，以及模拟商场的电梯系统等。这些题目往往不是单独提出来的，而是与其他题目有所联系，解决这些相似但又有微妙区别的题目能够帮助锻炼思维能力。

本书的讲解风格是较为明晰的。每一章在列出题目之前，都会先讲解题目所涉及的基础知识，使读者能够更加顺畅地切入题目本身。在解答部分，作者会用通俗易懂的语言把解题思路简洁地描述出来。这些思路不仅可以用来解决当前问题，而且还对解决其他题目有所启发。有时只要稍作变换，就可以用来应对同类问题。

把思路讲清楚之后，作者会给出解法，而且有时会从不同的思考方向给出多种解法。这可以令人看到同一道题的各种解法之间的区别，从而认清其利弊及适用场合，也可以令人在各种问题的解决办法之间体会相似之处，从而将多个知识点融会贯通。

开发者可以通读此书，也可以将其作为常备的参考手册，随时翻看，以求温故而知新。

翻译本书的过程中，我得到机械工业出版社华章公司诸位编辑与工作人员的帮助，在此深表谢意。

最后要说明的是，书中的范例代码只是为了解释思路而写的，具体运用时，可能需要酌情调整。译文对作者所用的某些分析方法、结论以及实现代码，只做了必要的技术处理，其余均保留原样。代码方面的问题，可访问原书网站 www.careermonk.com，以联系作者。针对文本翻译方面的问题，可访问 github.com/jeffreybaoshenlee/ciq-errata/issues 留言，或发邮件至 eastarstormlee@gmail.com，给我以批评和指教。

爱飞翔

2017年9月

亲爱的读者：

请先别急着翻过这一页。我知道，有很多人是不看前言的，但是这次，强烈建议大家读一读接下来的几段话，因为这里面有一些和其他前言不同的地方。

本书的前提是，读者已经掌握了一些计算机科学的基本知识。它的主要目标并不是提供与数据结构和算法有关的定理及证明，而是以一种逐步完善的方式来告诉大家：我们可以使用那些复杂度各不相同的办法来解决同一个编程问题。也就是说，针对每一个编程问题，笔者都会按照复杂度递减的顺序来讲解各种解法。总的来说，本书中囊括了各种编程解决方案，可以用来有效地应对面试、考试及校园招聘。

求职者如果能够读完本书，并很好地理解其中的内容，那么笔者保证：你肯定能够反过来给面试官提出几个难题。这正是本书想要达成的效果。

本书对于攻读工程学位及工程硕士的学生也会很有帮助。书中的每一章都会把重心放在编程问题及对这些问题所做的分析上，而不会只谈理论。笔者会在每章的开头讲解阅读此章所需的基础理论知识，然后列出一组问题。全书大约有 700 道算法题，而且每道题都有解答。

本书的大多数章节都会专注于问题本身，并对这些问题做出分析，而不会过多地聚焦在理论上面。各章开头会给出该章所要求的基本理论，然后提出问题。

书中的许多问题都有不止一种解法，而且这些解法的复杂度也各不相同。我们首先会讲解一种蛮力解法 (brute force solution)，然后逐渐改进，以求找到该问题的最佳解法 (best solution)。笔者尽量使大家能够理解每个算法所耗费的时间，以及它所占据的内存量。

你至少应该将本书通读一遍，以求对其中的全部话题都有所了解。然后可以根据自己的需要，直接跳到某一章，以参考其内容。尽管本书经过多次校对，但错误在所难免，本书勘误会更新在 www.CareerMonk.com 网站上。你可以定期查阅此网站，以了解最新的勘误、习题及解决方案。也欢迎你发送邮件到 Info@CareerMonk.com，给笔者提出宝贵的建议。

祝万事顺心、阅读愉快！

——Narasimha Karumanchi

印度理工学院孟买校区技术硕士、CareerMonk.com 创始人

致 谢 *Acknowledgements*

妈妈、爸爸，我不知道应该怎样对你们表示感谢才好。你们不遗余力地养育我，使我在稳定的环境中成长；你们始终坚持传统的美德，使我学会了赞美并热爱生活。对我来说，你们是最棒的父母，也是最好的榜样。我从你们身上懂得一个道理：有志者，事竟成！

这本书能够问世，得益于很多人的帮助。感谢他们为完善本书内容所做的努力。然而在这之前，我必须说：我已经尽力修正了审阅者所指出的错误，并且尽可能准确地描述了相关的协议与机制。如果还有尚未改正之处，那么应该归咎于我。

首先，要感谢在写书过程中提供支持的诸君，感谢你们能够与我讨论问题、阅读书稿、撰写内容、提出评价，并允许我引用相关的意见。同时感谢你们为编辑、校对及设计工作提供协助。尤其要感谢下面几位：

- 来自印度理工学院孟买校区、在 dataRPM 公司担任架构师的 Mohan Mullapudi
- 在瞻博网络担任高级顾问的 Navin Kumar Jaiswal
- 来自印度理工学院坎普尔校区、在明导国际公司任职的 A. Vamshi Krishna
- 在雅虎任职的 Hirak Chatterjee
- 拥有技术学士学位、在 HCL 担任技术主管的 Kondrakunta Murali Krishna
- StockMonks 公司的创始人 Chaganti Siva Rama Krishna Prasad
- NotionPress 公司的联合创始人 Naveen Valsakumar
- Nagarjuna 大学技术与科学学院的讲师、任职于 MLG 的 Ramanaiah

最后，我还要感谢 Guntur Vikas 学院的董事、Y. V. Gopala Krishna Murthy 教授、Ayub Khan 教授 (ACE Engineering Academy)、T. R. C. Bose (APTransco 的前主管)、Ch. Venkateswara Rao VNR Vignanjyothi (Hyderabad 的工程学院)、Ch. Venkata Narasaiah (IPS)、Yarapathineni Lakshmaiah (Gurazala 的 Manchikallu)，以及在研究期间为笔者及家人提供支持的每一位朋友。

Contents 目录

译者序	1.2.5
前言	2.2.5
致谢	3.2.5
第1章 编程基础	1
1.1 变量	1
1.2 数据类型	1
1.3 数据结构	2
1.4 抽象数据类型	3
1.5 内存与变量	3
1.6 指针	4
1.6.1 指针的声明	4
1.6.2 指针的使用	5
1.6.3 指针的操纵	6
1.6.4 数组与指针	7
1.6.5 动态内存分配	7
1.6.6 函数指针	7
1.7 参数传递的方式	8
1.7.1 实际参数与形式参数	8
1.7.2 参数传递的语义	8
1.7.3 各种编程语言所支持的参数传递方式	9
1.7.4 按值传递	9
1.7.5 按结果传递	10
1.7.6 有可能发生的参数冲突	10
1.7.7 按值-结果传递	11
1.7.8 按引用传递(别名机制)	11
1.7.9 按名称传递	12
1.8 绑定	12
1.8.1 静态绑定(前期绑定)	13
1.8.2 动态绑定(后期绑定)	13
1.9 作用域	13
1.9.1 静态作用域	13
1.9.2 动态作用域	14
1.10 存储类别	15
1.10.1 存储类别为 auto 的变量	15
1.10.2 存储类别为 extern 的变量	16
1.10.3 存储类别为 register 的变量	18
1.10.4 存储类别为 static 的变量	19
1.11 存储空间的安排	19
1.12 编程方式	22
1.12.1 无结构的编程	22
1.12.2 过程式的编程	22
1.12.3 模块式的编程	22
1.12.4 面向对象的编程	23
1.13 面向对象编程的基本概念	23

1.13.1	类与对象	24	2.4.9	Perl 语言的运算符	101
1.13.2	封装	24	2.4.10	条件语句	110
1.13.3	抽象	25	2.4.11	循环	113
1.13.4	数据隐藏	25	2.4.12	子例程	115
1.13.5	多态	25	2.4.13	字符串操作	117
1.13.6	继承	26	2.4.14	包 / 模块	118
1.13.7	继承的类型	26	2.5	Python	118
1.13.8	动态绑定	27	2.5.1	什么是 Python	118
1.13.9	消息传递	28	2.5.2	布尔类型	119
			2.5.3	整数	119
			2.5.4	字符串	119
			2.5.5	列表与元组	121
			2.5.6	函数	122
			2.5.7	把代码包装成模块	123
第 2 章	脚本语言	83	第 3 章	与设计有关的面试题	124
2.1	解释器与编译器	83	3.1	术语介绍	124
2.1.1	编译器	83	3.2	技巧	125
2.1.2	解释器	84	3.3	可供练习的其他设计问题	179
2.1.3	编译器与解释器的区别	84	第 4 章	操作系统的概念	180
2.2	什么是脚本语言	84	4.1	术语介绍	180
2.3	shell 脚本编程	85	4.2	与操作系统概念有关的问题	183
2.3.1	命令的重定向与管道	85	第 5 章	计算机网络的基础知识	188
2.3.2	变量	86	5.1	介绍	188
2.3.3	命令行参数	87	5.2	局域网与广域网	188
2.3.4	命令替换	88	5.3	数据包分割与多路复用	189
2.3.5	算术扩展	88	5.4	终端设备	190
2.3.6	控制结构	88	5.5	中介设备	190
2.3.7	函数	92	5.6	集线器、交换机与路由器的定义	191
2.4	Perl	94	5.7	介质	192
2.4.1	从“Hello world!”程序开始	94	5.8	端对端网络与客户端 / 服务器网络	192
2.4.2	Perl 的命令行参数	95			
2.4.3	Perl 的数据类型与变量	95			
2.4.4	引用	98			
2.4.5	声明变量	98			
2.4.6	变量的作用域	99			
2.4.7	字符串字面量	99			
2.4.8	Perl 的标准输入端	100			

5.9	互联网是如何运作的	193	8.15	三种表示法的性质	227
5.10	OSI 模型与 TCP/IP 模型的区别	196	8.16	常用的对数公式与求和公式	227
5.11	客户端 / 服务器结构与互联网	197	8.17	分治算法的主定理	227
5.12	ARP 与 RARP	198	8.18	与分治算法的主定理有关的问题	228
5.13	子网	199	8.19	递减式递推 (减而治之) 算法的主定理	229
5.14	路由器的工作原理	200	8.20	另一种递减式递推 (减而治之) 算法的主定理	229
5.15	单播、广播、多播	201	8.21	与算法分析有关的问题	230
5.16	tracert/traceroute 及 ping 命令的工作原理	202			
5.17	什么是 QoS	203			
第 6 章	数据库概念	204	第 9 章	递归与回溯	240
6.1	术语介绍	204	9.1	介绍	240
6.2	与数据库概念有关的问题	206	9.2	什么是递归	240
第 7 章	智力题	213	9.3	为什么要用递归的办法解决问题	240
7.1	智力题	213	9.4	递归函数的格式	241
第 8 章	算法介绍	217	9.5	演示递归调用时的内存占用情况	241
8.1	什么是算法	217	9.6	递归与迭代	242
8.2	为什么要做算法分析	218	9.7	运用递归时的注意事项	243
8.3	算法分析的目标	218	9.8	递归算法举例	243
8.4	什么是运行时间分析	218	9.9	与递归有关的问题	243
8.5	怎样对比不同的算法	218	9.10	什么是回溯	245
8.6	什么是增长率	219	9.11	回溯算法举例	245
8.7	几种常见的增长形式	219	9.12	与回溯有关的问题	245
8.8	算法分析的类型	220	第 10 章	链表	248
8.9	渐近表示法	221	10.1	什么是链表	248
8.10	大 O 表示法	221	10.2	将链表用作抽象的数据类型	248
8.11	大 Ω 表示法	222	10.3	为什么要用链表	249
8.12	大 Θ 表示法	223	10.4	数组概述	249
8.13	算法分析为什么又叫渐近分析	225	10.5	比较链表、数组与动态数组	250
8.14	渐近分析指南	225	10.6	单链表	251
			10.7	双链表	256
			10.8	循环链表	261

10.9	节省内存的双链表	266	13.10	异或树	379
10.10	松散链表	268	13.11	二叉搜索树	380
10.11	跳跃链表	273	13.12	平衡二叉搜索树	395
10.12	与链表有关的问题	276	13.13	AVL 树	396
第 11 章	栈	295	13.14	其他形式的树	413
11.1	什么是栈	295	13.14.1	红黑树	413
11.2	怎样使用栈	296	13.14.2	伸展树	414
11.3	将栈用作抽象数据类型	296	13.14.3	扩充树(增强树)	414
11.4	栈的运用	296	13.14.4	区间树(区段树)	415
11.5	实现	297	13.14.5	替罪羊树	416
11.6	对比各种实现方式	302	第 14 章	优先级队列与堆	418
11.7	与栈有关的问题	303	14.1	什么是优先级队列	418
第 12 章	队列	324	14.2	将优先级队列用作抽象数据 结构	418
12.1	什么是队列	324	14.3	运用	419
12.2	如何使用队列	324	14.4	实现	419
12.3	将队列用作抽象数据类型	325	14.5	堆与二叉堆	420
12.4	异常	325	14.6	二叉堆	421
12.5	运用	325	14.7	与优先级队列和堆有关的问题	428
12.6	实现	326	第 15 章	图算法	442
12.7	与队列有关的问题	331	15.1	介绍	442
第 13 章	树	337	15.2	术语表	442
13.1	什么是树	337	15.3	图的运用	446
13.2	术语表	337	15.4	将图用作抽象的数据结构	446
13.3	二叉树	339	15.4.1	邻接矩阵	446
13.4	二叉树的类型	339	15.4.2	邻接列表	447
13.5	二叉树的性质	340	15.4.3	邻接集合	449
13.6	遍历二叉树	342	15.4.4	表示图的方法的对比	449
13.7	泛化树(N 叉树)	362	15.5	图的遍历	449
13.8	通过线索二叉树来遍历	369	15.5.1	深度优先搜索(DFS)	450
13.9	表达式树	376	15.5.2	广度优先搜索(BFS)	454

15.5.3 对比 DFS 与 BFS	456	第 17 章 搜索	500
15.6 拓扑排序	457	17.1 什么是搜索	500
15.7 最短路径算法	458	17.2 为什么要搜索	500
15.8 最小生成树	465	17.3 各种类型的搜索	500
15.9 与图算法有关的问题	469	17.4 在无序的数据中执行线性搜索	501
第 16 章 排序	475	17.5 在已经排好序 / 有序的数组中 执行线性搜索	501
16.1 什么是排序	475	17.6 二分搜索	501
16.2 为什么要排序	475	17.7 对比几种基本的搜索算法	502
16.3 排序算法的分类方式	475	17.8 符号表与哈希	502
16.3.1 按照比较的次数来分类	475	17.9 字符串搜索算法	502
16.3.2 按照交换操作的次数来 分类	476	17.10 与搜索有关的问题	503
16.3.3 按照内存使用量来分类	476	第 18 章 选择算法	530
16.3.4 按照是否递归来分类	476	18.1 什么是选择算法	530
16.3.5 按照是否稳定来分类	476	18.2 通过排序来选择	530
16.3.6 按照适应性来分类	476	18.3 基于分区的选择算法	531
16.4 其他的分类方式	476	18.4 线性选择算法——中位数的 中位数算法	531
16.5 冒泡排序	477	18.5 把最小的 k 个元素找出来	531
16.6 选择排序	478	18.6 与选择算法有关的问题	531
16.7 插入排序	479	第 19 章 符号表	541
16.8 希尔排序	481	19.1 介绍	541
16.9 归并排序	483	19.2 什么是符号表	541
16.10 堆排序	485	19.3 实现符号表	542
16.11 快速排序	485	19.4 比较实现符号表的各种方式	543
16.12 树排序	488	第 20 章 哈希	544
16.13 线性时间的排序算法	489	20.1 什么是哈希	544
16.14 计数排序	489	20.2 为什么要使用哈希	544
16.15 桶排序	490	20.3 将哈希表用作抽象数据结构	544
16.16 基数排序	490	20.4 哈希技术的原理	545
16.17 拓扑排序	491		
16.18 外部排序	491		
16.19 与排序有关的问题	492		

20.5 哈希技术的组成要素	546	21.5.2 构建有限状态机时的 注意事项	568
20.6 哈希表	546	21.5.3 匹配算法	568
20.7 哈希函数	547	21.6 KMP 算法	569
20.8 负载因子	547	21.6.1 前缀表	569
20.9 冲突	547	21.6.2 匹配算法	571
20.10 冲突解决技术	548	21.7 Boyce-Moore 算法	573
20.11 单独链接法	548	21.8 适合用来保存字符串的数据 结构	573
20.12 开放定址	548	21.9 用哈希表来保存字符串	574
20.12.1 线性探测	548	21.10 用二叉搜索树来存放字符串	574
20.12.2 二次探测	549	21.11 前缀树	574
20.12.3 二次哈希	550	21.11.1 什么是前缀树	574
20.13 比较各种冲突解决技术	550	21.11.2 为什么要使用前缀树	575
20.14 哈希技术如何把复杂度降为 $O(1)$	551	21.11.3 声明前缀树	575
20.15 哈希技术	551	21.11.4 向前缀树中插入字符串	576
20.16 哪些问题不适合用哈希表解决	551	21.11.5 在前缀树中查找字符串	576
20.17 Bloom 过滤器	552	21.11.6 用前缀树来表示字符串 有什么缺点	577
20.17.1 工作原理	552	21.12 三元搜索树	577
20.17.2 选择合适的哈希函数	553	21.12.1 声明三元搜索树	577
20.17.3 设置长度合适的位向量	553	21.12.2 向三元搜索树中插入 字符串	578
20.17.4 空间方面的优势	553	21.12.3 在三元搜索树中查找 字符串	580
20.17.5 时间方面的优势	554	21.12.4 显示三元搜索树中的全部 字符串	580
20.17.6 实现	554	21.12.5 在三元搜索树中查找最长 的字符串	581
20.18 与哈希有关的问题	554	21.13 比较二叉搜索树、前缀树及 三元搜索树	581
第 21 章 字符串算法	565	21.14 后缀树	581
21.1 介绍	565	21.14.1 前缀与后缀	582
21.2 字符串匹配算法	565		
21.3 蛮力法	566		
21.4 Rabin-Karp 字符串匹配算法	566		
21.5 用有限状态机来实现字符串 匹配算法	567		
21.5.1 状态机的运作过程	568		

21.14.2	规律	582	23.3	哪些问题适合用贪婪算法求解	595
21.14.3	什么是后缀树	582	23.4	贪婪算法是否能应对所有的问题	596
21.14.4	构建后缀树	582	23.5	贪婪算法的优点与缺点	596
21.14.5	运用后缀树	585	23.6	可以运用贪婪算法的场合	596
21.15	与字符串有关的问题	585	23.7	理解贪婪算法	596
第 22 章	算法设计技巧	591	23.8	与贪婪算法有关的问题	599
22.1	介绍	591	第 24 章	分治算法	606
22.2	分类	591	24.1	介绍	606
22.3	按实现方式分类	591	24.2	什么是分治策略	606
22.3.1	递归算法与迭代算法	591	24.3	分治技术是否能用来解决所有的问题	606
22.3.2	过程式算法与声明式 (非过程式)算法	592	24.4	用示意图来说明分治技术	607
22.3.3	串行算法、并行算法、 分布式算法	592	24.5	理解分治技术	607
22.3.4	确定性的算法与非确定性 的算法	592	24.6	分治技术的优点	608
22.3.5	精确算法与近似算法	592	24.7	分治技术的缺点	608
22.4	按设计方式分类	592	24.8	分治算法的主定理	609
22.4.1	贪婪算法	592	24.9	分治算法的适用场合	609
22.4.2	分治算法	593	24.10	与分治技术有关的问题	609
22.4.3	动态规划算法	593	第 25 章	动态规划	623
22.4.4	线性规划算法	593	25.1	介绍	623
22.4.5	归约(转化并治理)算法	593	25.2	什么是动态规划策略	623
22.5	其他分类方式	594	25.3	什么样的问题适合用动态规划 来解决	624
22.5.1	按研究领域划分	594	25.4	动态规划技术能否应对所有 的问题	624
22.5.2	按复杂程度划分	594	25.5	动态规划的方式	624
22.5.3	随机化的算法	594	25.5.1	自下而上的动态规划	624
22.5.4	分支定界与回溯	594	25.5.2	自上而下的动态规划	624
第 23 章	贪婪算法	595	25.5.3	两种规划方向的对比	624
23.1	介绍	595	25.6	动态规划算法示例	625
23.2	贪婪算法的策略	595			

25.7	理解动态规划	625	27.2.7	判断第 K 个二进制位有没有 设置 (或者说是不是 1)	676
25.7.1	斐波那契数列	625	27.2.8	设置第 K 个二进制位 (也就是将其设为 1)	677
25.7.2	求某数的阶乘	627	27.2.9	清除第 K 个二进制位 (也就是将其设为 0)	677
25.7.3	最长的公共子序列	628	27.2.10	切换第 K 个二进制位	677
25.8	与动态规划有关的问题	631	27.2.11	把值为 1 且最靠右的 二进制位设置成 0	677
第 26 章 复杂度类		668	27.2.12	把值为 1 且最靠右的 二进制位标出来	677
26.1	介绍	668	27.2.13	把值为 0 且最靠右的 二进制位标出来	678
26.2	多项式时间 / 指数时间	669	27.2.14	判断某数是不是 2 的幂	678
26.3	什么是判定性问题	669	27.2.15	与 2 的幂相乘	678
26.4	判定过程	669	27.2.16	与 2 的幂相除	678
26.5	什么是复杂度类	669	27.2.17	求出与 2 的幂相除的余数	678
26.6	复杂度类的类型	669	27.2.18	将二进制表示形式反转	678
26.6.1	P 类	669	27.2.19	统计值为 1 的二进制位 个数	679
26.6.2	NP 类	670	27.2.20	创建掩码, 以便将尾部 连续出现的 0 标注出来	680
26.6.3	反 NP 类	670	27.2.21	把奇数位置上的二进制位 与偶数位置上的二进制位 互换	680
26.6.4	P、NP 与反 NP 之间的关系	670	27.2.22	用不做除法的方式来求 平均值	680
26.6.5	NP 困难类	670	27.3	其他编程问题	680
26.6.6	NP 完全类	671	第 28 章 非技术问题		686
26.6.7	P、NP、反 NP、NP 困难与 NP 完全之间的关系	671	28.1	面试技巧	686
26.6.8	P 是否等于 NP	671	28.2	非技术问题举例	688
26.7	归约	672			
第 27 章 其他概念		675			
27.1	介绍	675			
27.2	与位运算有关的技巧	675			
27.2.1	按位与	675			
27.2.2	按位或	675			
27.2.3	按位异或	676			
27.2.4	左移位	676			
27.2.5	右移位	676			
27.2.6	按位取反	676			

编程基础

本章旨在解释编程语言的基础知识。读者将会了解数据类型、指针、作用域的规则、程序的内存布局、参数传递的方式、编程语言的类型，以及与之相关的一些知识。

1.1 变量

在给变量下定义之前，我们首先来谈谈它与传统数学方程之间的联系。大家从小就求解过许多数学方程，例如下面的方程：

$$x^2+2y-2=1$$

大家并不需要关注这个方程式的具体用途，而是要注意：该方程用 x 与 y 这样的名称 (name) 来存放数值 (value) 或数据 (data)，这意味这些名称是一种用来表示数据的占位符号。同理，在计算机科学中，我们也需要用类似的办法来存放数据，这就是变量 (variable) 的用途。

1.2 数据类型

在上面那个方程式里面，像 x 与 y 这样的变量既可以取整数值 (例如 10、20)，又可以取实数值 (例如 0.23、5.5)，还可以仅仅在 0 与 1 之间取值。要想解开这个方程，我们必须确定：其中的变量可以取什么类型的数值，而这正是数据类型 (data type) 一词所要表达的意思。编程语言中的数据类型，是带有预定义值 (predefined value) 的一组数据。如整数 (integer)、浮点数 (floating point unit number)、字符 (character)、字符串 (string) 等，都是数据类型。