

TURING

Cracking the Coding Interview  
150 Programming Questions and Solutions

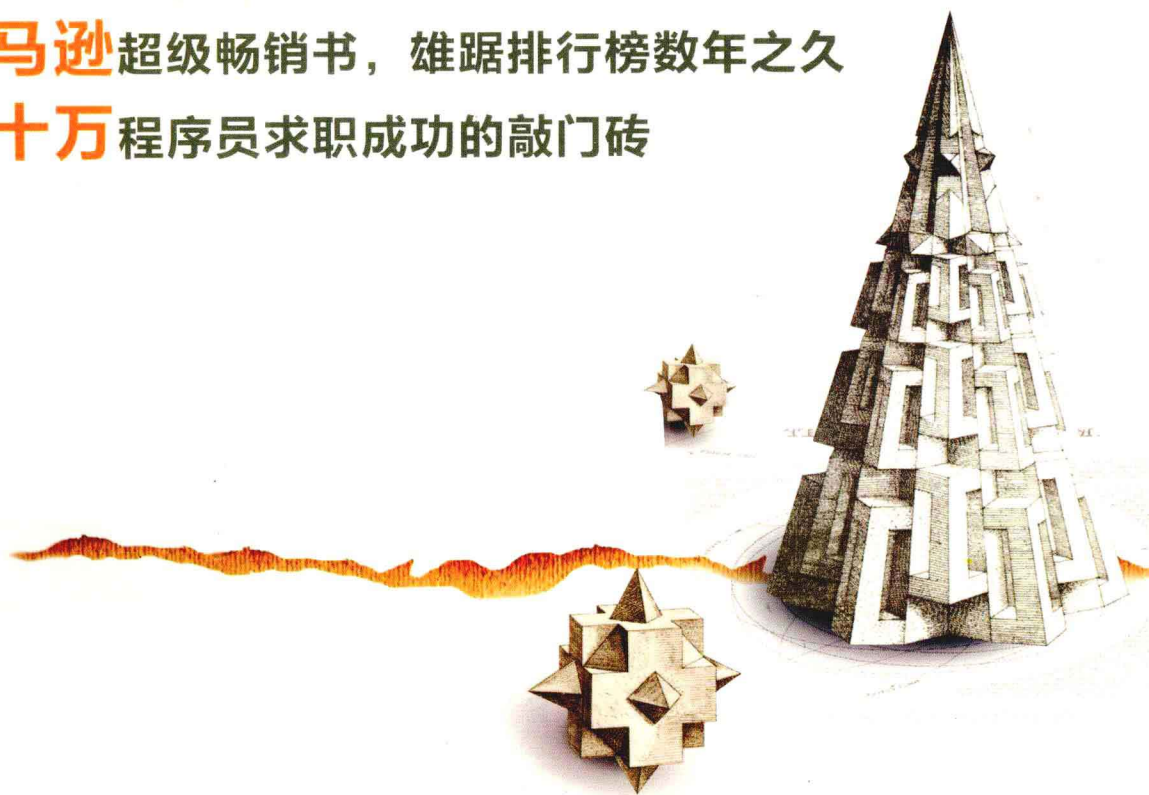
5th  
edition

# 程序员面试金典

第5版

[美] Gayle Laakmann McDowell 著 李琳骁 漆犇 译

全方位揭示**微软、苹果、谷歌**等IT名企招聘秘密  
**亚马逊**超级畅销书，雄踞排行榜数年之久  
**数十万**程序员求职成功的敲门砖



人民邮电出版社  
POSTS & TELECOM PRESS

TURING

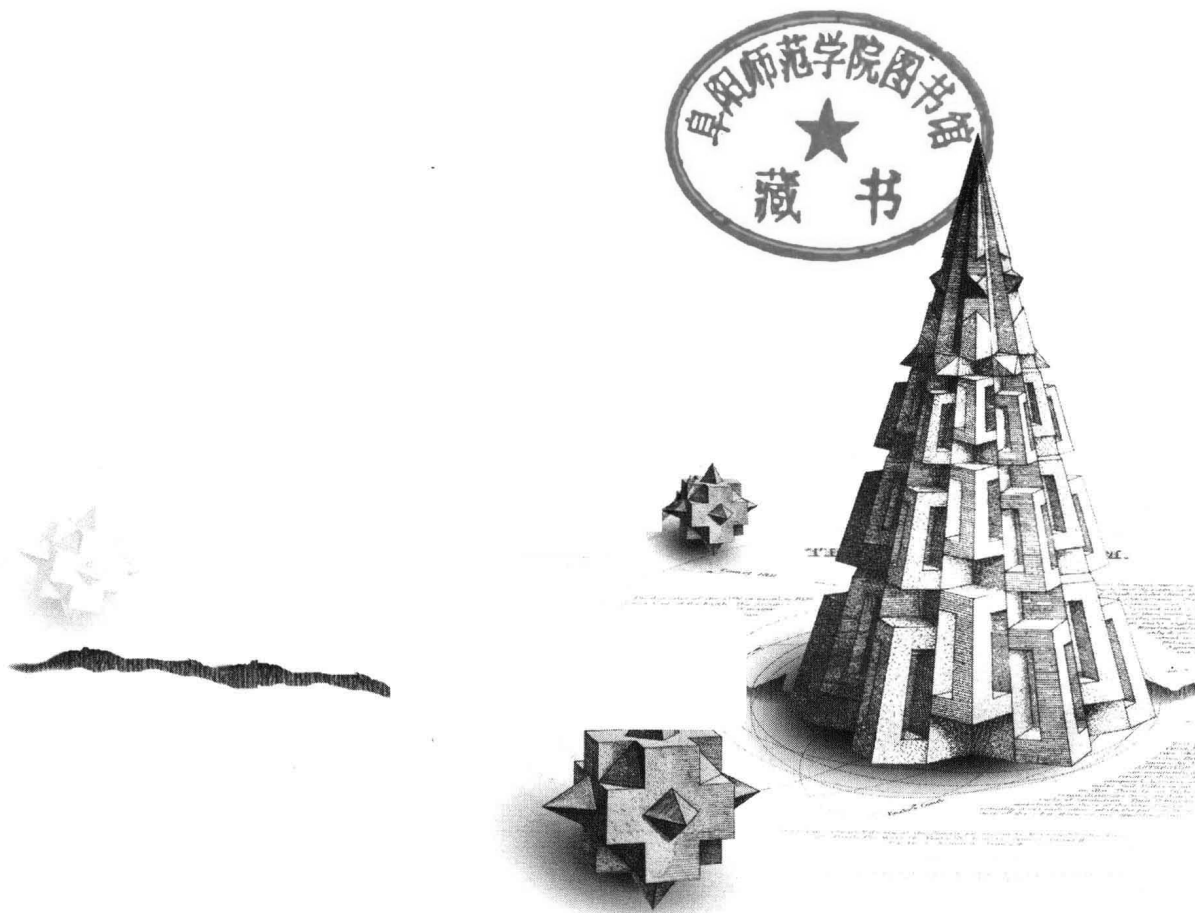
Cracking the Coding Interview  
150 Programming Questions and Solutions

5th  
edition

# 程序员面试金典

第5版

[美] Gayle Laakmann McDowell 著 李琳骁 漆轶 译



人民邮电出版社  
北京

## 图书在版编目(CIP)数据

程序员面试宝典：第5版 / (美) 麦克道尔  
(McDowell, G. L.) 著；李琳骁，漆犇译. — 北京：人  
民邮电出版社，2013. 11

书名原文：Cracking the coding interview:150  
programming questions and solutions, fifth edition  
ISBN 978-7-115-33291-2

I. ①程… II. ①麦… ②李… ③漆… III. ①程序设  
计—工程技术人员—资格考试—自学参考资料 IV.  
①TP311.1

中国版本图书馆CIP数据核字(2013)第245017号

## 内 容 提 要

本书是原谷歌资深面试官的经验之作，层层紧扣程序员面试的每一个环节，全面而详尽地介绍了程序员应当如何应对面试，才能在面试中脱颖而出。第1~7章主要涉及面试流程解析、面试官的幕后决策及可能提出的问题、面试前的准备工作、对面试结果的处理等内容；第8~9章从数据结构、概念与算法、知识类问题和附加面试题4个方面，为读者呈现了出自微软、苹果、谷歌等多家知名公司的150道编程面试题，并针对每一道面试题，分别给出了详细的解决方案。

本书适合程序开发和设计人员阅读。

- 
- ◆ 著 [美] Gayle Laakmann McDowell
  - 译 李琳骁 漆 犇
  - 责任编辑 丁晓昀
  - 执行编辑 李 鑫 陈婷婷
  - 责任印制 焦志炜
  - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
  - 邮编 100061 电子邮件 315@ptpress.com.cn
  - 网址 <http://www.ptpress.com.cn>
  - 北京艺辉印刷有限公司印刷
  - ◆ 开本：800×1000 1/16
  - 印张：23.25
  - 字数：549千字 2013年11月第1版
  - 印数：1-5 000册 2013年11月北京第1次印刷
  - 著作权合同登记号 图字：01-2012-1951号
- 

定价：59.00元

读者服务热线：(010)51095186转604 印装质量热线：(010)67129223

反盗版热线：(010)67171154

广告经营许可证：京崇工商广字第0021号

# 版权声明

© POSTS & TELECOM PRESS 2013. Authorized translation of the English edition © 2008 CareerCup. This translation is published and sold by permission of Gayle Laakmann McDowell, the owner of all rights to publish and sell the same.

本书中文简体字版由Gayle Laakmann McDowell授权人民邮电出版社独家出版。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。

版权所有，侵权必究。

# 序

亲爱的读者：

我先做个自我介绍。

我不是什么招聘人员。我是一名软件工程师。正因如此，我深知大家要在面试现场迅速想出精妙算法并在白板上写下完美代码的感受。之所以能感同身受，是因为我与你们有过同样的经历，我参加过谷歌、微软、苹果、亚马逊以及其他诸多公司的面试。

而且，我也当过面试官，让求职者做同样的事情。我还筛选过成千上万份简历，在其中“上下求索”，希望挑出那些或许能在面试难关中脱颖而出的工程师。在谷歌时，我与招聘委员会的同僚有过激烈争辩，探讨某位求职者是否达到了录用要求。我对招聘各环节了如指掌，相关经验也很丰富。

而现在，我亲爱的读者，你也许要在明天、下周或是明年去迎接面试挑战。你可能已经拿到或者正在攻读计算机科学或相关专业的学位。本书并不打算给大家重温有关二叉查找树的基本知识，或者该如何遍历链表。想必你已经掌握这些内容；倘若没有，还请先找些数据结构的基础资料仔细研读。

本书旨在帮助你加深对计算机科学基础知识的理解，并学会该如何运用这些基础知识，成功闯过技术面试这一关。

本书在第四版的基础上做了大量更新，增补篇幅达 200 多页。第五版添补了不少面试题，修订了部分原有题目的解法，并新增了几个章节和其他内容。欢迎访问我们的网站，你可以跟其他求职者互通有无，发现新天地。

与此同时，我也感到无比兴奋，你一定能从本书中学到新的技能。充分的准备会让你在技术和人际沟通技能等诸多方面更进一步。不管最终结果如何，只要拼尽全力，无怨无悔！

请各位读者务必用心研读本书前面的介绍性章节，其中的要点和启示也许可以决定你的面试结果，“录用”与“拒绝”就在一线之间。

此外，切记——面试非易事！根据我在谷歌多年面试的经历，我留意到有些面试官会问一些“简单的”问题，有些则会专挑难题来问。但是你知道吗？面试中碰到简单的问题，也不见得就能轻松过关。完美解决问题（只有极少数求职者能做到！）不是公司录用你的关键，只有题答得比其他求职者更出色才能让你脱颖而出。所以，碰到棘手的难题也不要惊慌，或许其他人一样觉得很难。

请努力学习，不断实践。祝你好运！

盖尔·拉克曼·麦克道尔  
CareerCup.com 创始人兼 CEO  
《金领简历：敲开苹果、微软、谷歌的大门》及本书作者

# 前 言

## 招聘中的问题

讨论完招聘事宜，我们又一次沮丧地走出会议室。那天，我们重新审查了十位“过关”的求职者，但是全都不堪录用。我们很纳闷，是我们太过苛刻了吗？

我尤为失望的是，我推荐的一名求职者也被拒了。他是我以前的学生，以高达 3.73 的平均分（GPA）毕业于华盛顿大学，这可是世界上最棒的计算机专业院校之一。此外，他还完成了大量的开源项目工作。他精力充沛、富于创新、踏实能干、头脑敏锐，不论从哪方面来看，他都堪称真正的极客。

但是，我不得不同意其他招聘人员的看法：他还是不够格。就算我的强力推荐可以让他侥幸过关，在后续的招聘环节还是会失利，因为他的硬伤太多了。

尽管面试官都认为他很聪明，但他答题总是磕磕绊绊的。大多数成功的求职者都能轻松搞定第一道题（这一题广为人知，我们只是略作调整而已），可他却没能想出合适的算法。虽然他后来给出了一种解法，但没有提出针对其他情形进行优化的解法。最后，开始写代码时，他草草地采用了最初的思路，可这个解法漏洞百出，最终还是没能搞定。他算不上表现最差的求职者，但与我们的“录用底线”却相去甚远，结果只能是铩羽而归。

几个星期后，他给我打电话询问反馈意见，我很纠结，不知该怎么跟他说。他需要变得更聪明些吗？不，他其实智力超群。做个更好的程序员？不，他的编程技能和我见过的一些最出色的程序员不相上下。

跟许多积极上进的求职者一样，他准备得非常充分。他研读过 Brian W. Kernighan 和 Dennis M. Ritchie 合著的《C 程序设计语言》，麻省理工学院出版的《算法导论》等经典著作。他可以细数很多平衡树的方法，也能用 C 语言写出各种花哨的程序。

我不得不遗憾地告诉他：光是看这些书还远远不够。这些经典学院派著作教会了人们错综复杂的研究理论，对程序员的面试却助益不多。为什么呢？容我稍稍提醒你一下：即使从学生时代起，你的面试官们其实都没怎么接触过所谓的红黑树（Red-Black Trees）算法。

要顺利通过面试，就得“真枪实弹”地做准备。你必须演练真正的面试题，并掌握它们的解题模式。

这本书就是我在根据自己在顶尖公司积累的第一手面试经验提炼而成的精华。我曾经与数百名

求职者有过“交锋”，本书可以说是我面试几百位求职者的结晶。同时，我还从成千上万求职者与面试官提供的问题中精挑细选了一部分。这些面试题出自许多知名的高科技公司。可以说，这本书囊括了 150 道世界上最好的程序员面试题，都是从数以千计的好问题中挑选出来的。

## 我的写作方法

本书重点关注算法、编码和设计问题。为什么呢？尽管面试中也会有“行为问题”，但是答案会随个人的经历而千变万化。同样，尽管许多公司也会考问细节（例如，“什么是虚函数？”），但通过演练这些问题而取得的经验非常有限，更多地是涉及非常具体的知识点。本书只会述及其一些中一些问题，以便你了解它们“长”什么样。当然，对于那些可以拓展技术技能的问题，我会给出更详细的解释。

## 我的教学热情

我特别热爱教学。我喜欢帮助人们理解新概念，并提供一些学习工具，从而充分激发他们的学习热情。

我第一次“正式”的教学经验是在美国宾夕法尼亚大学就读期间，那时我才大二，担任本科计算机科学课程的助教（TA）。我后来还在其他一些课程中担任过助教，最终在大学里推出了自己的计算机科学课程，也就是给大家教授一些实际的“动手”技能。

在谷歌担任工程师时，培训和指导“Nooglers”（意指谷歌新员工。没错，他们就是这么称呼新人的！）是我最喜欢的工作之一。后来，我还利用“20%自由支配时间”在华盛顿大学教授计算机科学课程。

《程序员面试宝典》、《金领简历》和 CareerCup.com 网站都能充分体现我的教学热情。即便是现在，你也会发现我经常出现在 CareerCup.com 上为用户答疑解惑。

请加入我们的行列吧！

Gayle Laakmann McDowell



# 致 谢

生命中很多事情都离不开团队合作，这本书也不例外。在创作本书的过程中，我得到了很多人的帮助，尽管涌泉都难以回报，我还是想在此聊表寸心。

首先，我要感谢我的丈夫约翰，他是最坚强的后盾，让我有勇气将此书一改再改并接连修订了五版。如果没有他的支持，我很可能就做不到这一点。

其次，我要感谢家母，她让我认识到编程无比重要，而写出优美流畅的文字更为重要。毫无疑问，她是一位无与伦比的工程师、企业家，最重要的是，她是一位伟大的母亲。

接下来我要感谢诸多好友的鼓励，尤其是加尔顿·英格里绪。不管是我需要帮助还是听我发牢骚，她总是默默陪在我身边，一如既往地支持我。

最后，我还要感谢那些给予回复与建议的读者：谢谢你们！我要特别感谢维尼特·萨哈和普拉雷·瓦玛，他们细致入微地审阅了书中的每一道题，用心之极，佩服不已。相信你们的同事和主管一定会为有这么出色的伙伴而骄傲。

再次深表谢意！

# 目 录

第 1 章 面试流程	1	第 6 章 技术面试题	27
1.1 概述	1	6.1 技术准备	27
1.2 面试题的来源	2	6.2 如何应对	29
1.3 准备时间表与注意事项	3	6.3 算法题的五种解法	31
1.4 面试评估流程	4	6.4 怎样才算好代码	34
1.5 答题情况	5	第 7 章 录用通知及其他	39
1.6 着装规范	6	7.1 如何处理录用与被拒的情况	39
1.7 十大常见错误	6	7.2 如何评估录用待遇	40
1.8 常见问题解答	8	7.3 录用谈判	41
第 2 章 面试揭秘	9	7.4 入职须知	42
2.1 微软面试	10	第 8 章 面试题	44
2.2 亚马逊面试	10	8.1 数组与字符串	45
2.3 谷歌面试	11	8.2 链表	47
2.4 苹果面试	12	8.3 栈与队列	49
2.5 Facebook 面试	13	8.4 树与图	51
2.6 雅虎面试	14	8.5 位操作	54
第 3 章 特殊情况	15	8.6 智力题	57
3.1 有工作经验的求职者	15	8.7 数学与概率	59
3.2 测试人员及 SDET	15	8.8 面向对象设计	64
3.3 项目经理与产品经理	16	8.9 递归和动态规划	66
3.4 技术主管与部门经理	17	8.10 扩展性与存储限制	69
3.5 创业公司的面试	18	8.11 排序与查找	73
第 4 章 面试之前	19	8.12 测试	78
4.1 积累相关经验	19	8.13 C 和 C++	83
4.2 构建人际网络	20	8.14 Java	89
4.3 写好简历	21	8.15 数据库	93
第 5 章 行为面试题	23	8.16 线程与锁	98
5.1 准备工作	23	8.17 中等难题	104
5.2 如何应对	25	8.18 高难度题	105

---

第9章 解题技巧	107	9.10 扩展性与存储限制	241
9.1 数组与字符串	108	9.11 排序与查找	255
9.2 链表	117	9.12 测试	269
9.3 栈与队列	131	9.13 C和C++	274
9.4 树与图	146	9.14 Java	284
9.5 位操作	163	9.15 数据库	290
9.6 智力题	175	9.16 线程与锁	296
9.7 数学与概率	179	9.17 中等难题	306
9.8 面向对象设计	192	9.18 高难度题	331
9.9 递归和动态规划	221	索引	358

## 第 1 章

# 面试流程

- 概述
- 面试题的来源
- 准备时间表与注意事项
- 面试评估流程
- 答题情况
- 着装规范
- 十大常见错误
- 常见问题解答

### 1.1 概述

大多数公司的面试流程其实都大同小异。本章会简述面试流程，以及企业到底想招募什么样的人才。这些信息将指导你如何做好面试准备，以及在面试过程中和面试结束后该如何应对。

收到面试通知后，你通常得先经历一次筛选面试（screening interview），一般通过电话进行。顶尖高校的应届毕业生则可能需要参加现场的筛选面试。

不要因“筛选面试”这个词儿而掉以轻心，筛选面试也很有可能涉及编码与算法问题，要求不见得比现场面试低。如果不确定它是不是技术筛选面试，不妨问问招聘助理面试官是什么来头，若是工程师，那十有八九会与技术相关。

许多公司会在面试中运用在线同步文档编辑系统，但也有可能让你直接在纸上写好代码，然后在电话里念给他们听。有些面试官甚至还会给你留“家庭作业”，或是要求你用电子邮件将写好的代码发给他们。

在现场面试（on-site interview）之前，通常会有一两轮筛选面试。现场面试大概有4到6轮，其中一轮可能是午餐面试。当然，午餐面试比较随意，面试官一般不会问你技术问题，甚至不会纳入面试评价范畴。但同时，这也是难得的好机会，你可以跟面试官探讨自己感兴趣的问题，了解公司的企业文化。其他几轮面试主要涉及技术问题，包括编码和算法等。此外，你可能还要回答与简历相关的问题。

面试结束后，面试官们会聚在一起讨论你的表现，或者提交书面评价。大多数情况下，公司招聘人员都会在一周内给你回复，告知应聘进展。

要是已经望穿秋水等了一个多星期，你也可以主动询问进展。就算招聘人员没有回应，也并不表示你被拒了（至少大的高科技公司是这样，其实几乎所有公司都是如此）。我再重复一次：没有回应表示你的应聘结果还是未知数。当然，人们都希望招聘方在得出最终结论时，及时通知求职者。

拖拖拉拉的情况确实有。等不及的话，不妨问问相关招聘人员，但务请有礼有节。招聘人员和我们一样，他们很忙，有些人会因此容易忘事。

### 1.2 面试题的来源

求职者经常会问我，某些公司最近都喜欢问哪些面试题？他们总以为面试题会应时而变。实际上，公司本身对面试题并没有什么倾向，这完全取决于面试官的个人喜好。容我解释一下。

在大公司里，面试官通常需要先参加一些面试培训课程。在谷歌，担任面试官之前，我先参加了一次由外部公司提供的专门培训。培训课程为期一天，有一半时间侧重于法律层面的事务，比如，面试官不能探问求职者的婚姻状况，不得询问种族，等等。另一半时间则在探讨如何应对“刺头”求职者，比如当问及编码问题或其他令求职者认为是在“羞辱”自己的问题时，要是求职者“暴跳如雷”，该怎么应对。培训过后，我又实地观摩了两次真正的面试，然后就开始独自面试了。

就是这样。我们受过的培训也不过如此，其实所有公司都大同小异。

根本就不存在什么“谷歌官方面试题清单”，也从来没有人要求我一定要问哪些特定的问题，或者必须避开哪些话题。

那我的面试题从何而来呢？其实，来源和大家一样。

面试官也当过求职者，他们会借用自己当年被拷问过的题目。又或者，有些面试官也会彼此交换题库。还有些人喜欢上网找问题，比如CareerCup.com网站。有些面试官也可能从上述渠道收集面试题，并或多或少做些调整。

就算真有公司给面试官准备好问题清单，这种情况也并不多见。面试官通常也会自行挑选问题，而且大家往往会有五六个常用的备选题目。

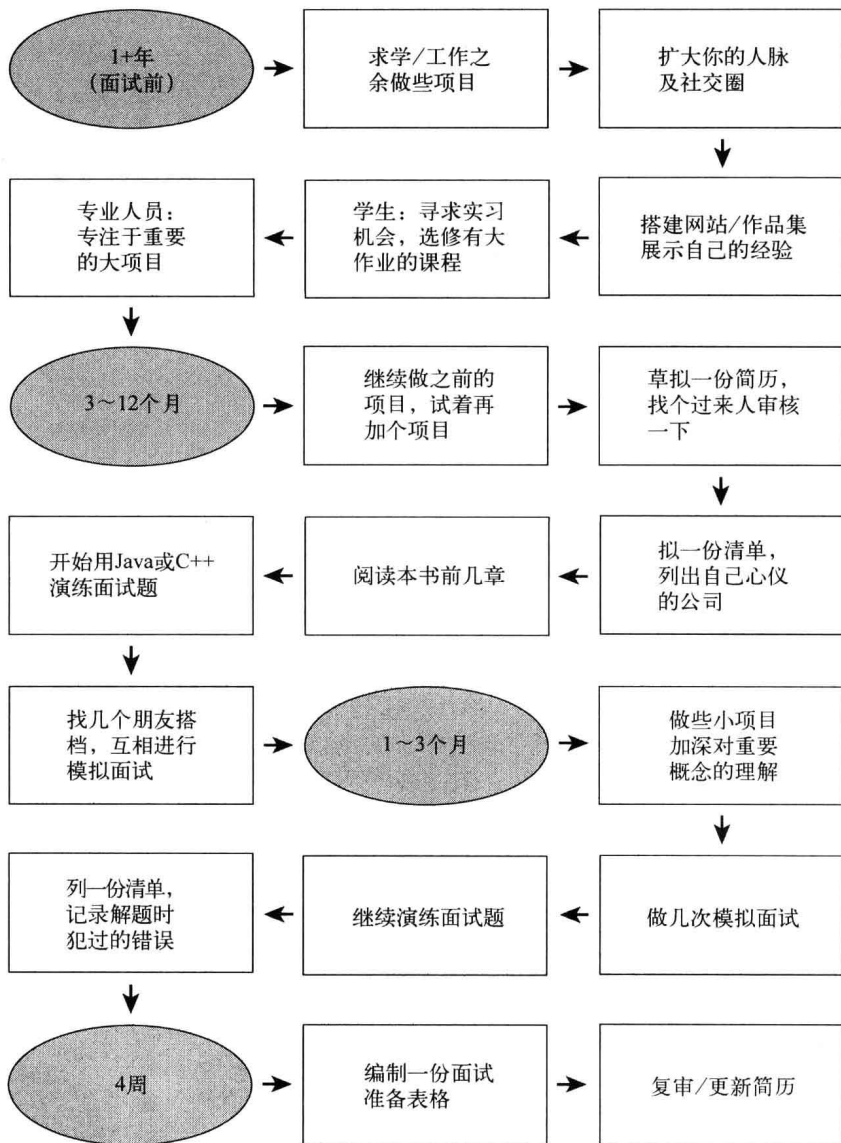
因此，下次在你想知道谷歌“最近”都问些什么问题的时候，不妨先停下来想一想。谷歌与亚马逊的面试题其实没什么不同，他们需要的都是软件开发人才。至于面试题是不是“最近流行的”也就更无关紧要了。万变不离其宗，因为这本来就得靠面试官自己去把握。

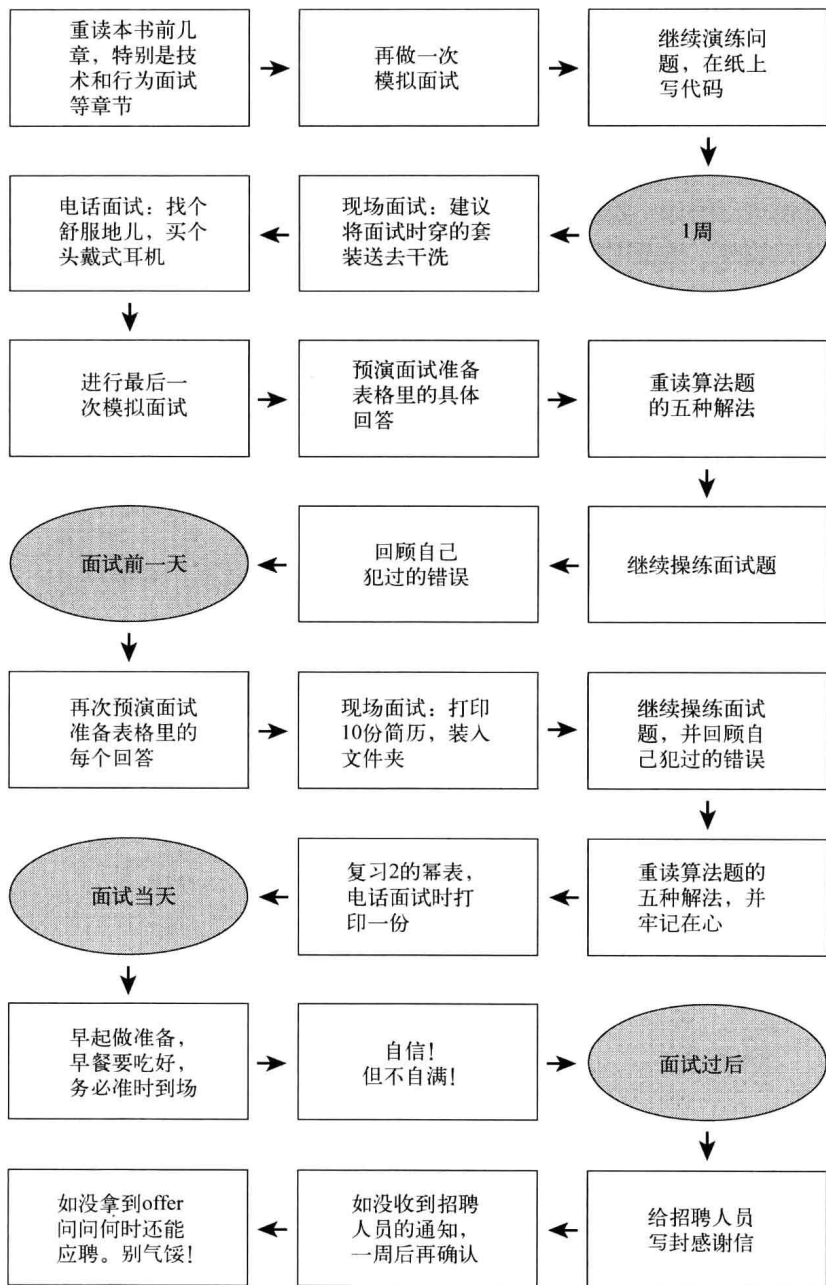
当然，总体上，不同的公司在风格上存在差异。互联网公司往往会提些系统设计方面的问题，而那些使用数据库的公司则明显偏爱数据库方面的问题。然而，大部分面试题无外乎就是数据结构 and 算法之类的，任何公司都会问到。

## 1.3 准备时间表与注意事项

“台上一分钟，台下十年功”，事实上你的面试表现取决于你的功底——离不开多年的积淀。你需要刚好具备能为公司所用的技术经验，然后还要准备好在面试中解决实际的技术问题。下面的时间表和流程图可以给你一些启发。

如果你起步比较晚，也不用担心。“尽人事，知天命”，请安心准备，祝你好运！





### 1.4 面试评估流程

招聘人员可能会告诉你，他们主要考查四个方面：工作经验、企业文化契合度、编程技能及

分析能力。这四个方面相辅相成，但在决定录用与否时，分量最重的通常还是编程技能和分析能力（或者看你是否聪明）。这也是为什么本书的主要篇幅都在探讨如何提升编程与算法技能。

当然，虽说编程与算法技能往往最为重要，但并不表示你可以忽视其他两个方面。

一旦进入大型科技公司的面试环节，你之前的工作经验就不是特别重要了，但它可能会左右面试官对你的看法。比如，如果你说起以前写的某个复杂程序的精彩之处，面试官很有可能会想：“哇，她可真聪明！”一旦他认定你智力超群，可能就会下意识地忽略你所犯的小错误。总之，面试并不会十分精确，对某些“软问题”做好充分准备会大有裨益。

创业公司比大公司更看重企业文化契合度（或你的个性，主要看是否与公司合拍）。举个例子，如果公司的企业文化鼓励员工独立做决定，那么喜欢听从指导的人就不太适合了。

此外，求职者因为过于自大、巧辩或抵触而被淘汰的情况也并不少见。我就遇到过，有位求职者对我提问的用词吹毛求疵，并抱怨这导致他解题不太顺利，后来他还对我的引导方式心生不满。这种“抵触心太重”的表现其实也是一个警示，果然，其他面试官对他的感觉也很不好。最后他被淘汰了。谁会愿意跟这种人一起共事呢？

所以，你应该注意以下几点。

- 如果人们都认为你骄傲自大、过于狡辩，或有其他负面评价，那你最好在面试中收敛一下。个性不讨喜的话，哪怕你的表现再好，也可能被拒。
- 准备一些与简历相关的问题。虽然这不是最重要的因素，但也不能掉以轻心。稍微花点时间准备就能起到很好的效果，做到“四两拨千斤”。
- 把主要精力用在编程与算法问题上。

最后，我还是要再强调一遍，面试并不会十分精确。你的表现可能会有失水准，招聘委员会（或不管是谁）有时候也会做出错误判断。就像任何群体一样，招聘委员会也可能被某位主导人物的观点所左右。这也许不公平，但这就是生活。

记住——这次被拒绝并不代表永远。一年内你还可以重新应聘，很多求职者都有过失利后再成功的经历。

不要气馁，失败是成功之母。

## 1.5 答题情况

有则谣传流传甚广且颇具迷惑性：求职者必须答对全部问题才会被录用。事实绝非如此。

首先，面试题的答案很难用“正确”和“错误”去简单评判。我个人在评估求职者的面试表现时，一般不会只看他们答对了几道题。相反，我会考量其最终解法是否最优，用时多久，代码整洁与否。这不只是单纯的是非判断，还要综合考虑很多因素。

其次，你的面试表现还会拿来跟其他求职者作比较。比如说，你用15分钟出色地解决了一道题，而另一个人不到5分钟就搞定了一道比较容易的题，是否就意味着那个人的表现比你好呢？也许是，但也未必。很自然，面试官出的题越简单，他们越是希望你尽快给出最佳答案。但要是题目很难，他们也不会指望你能答得又快又好，毕竟，出点纰漏也是在所难免的。



我在谷歌评估过数千名求职者的面试资料，其中只有一位求职者的面试表现堪称“完美无瑕”。其他人，包括最后被录用的几百个幸运儿，都或多或少犯过一些错。

## 1.6 着装规范

软件工程师一般都穿得比较随意。这一点从面试的着装规范也看得出来。参加面试时，推荐做法是穿得比同级别员工稍好一点。

以下是我给软件工程师（及测试人员）的面试着装建议，意在让大家找到一个“平衡点”：不要穿得过于正式，也不要太随意。其实，有很多人还是穿着牛仔裤和T恤衫参加创业公司或大公司的面试，也不会有什么大问题。毕竟，公司不是看你穿什么，而是看你的编程水平。

	创业公司	微软、谷歌、亚马逊、Facebook等科技巨头	非科技公司（包括银行）
男 性	卡其裤、休闲裤或整洁得体的牛仔裤。Polo衫或礼服衬衫	卡其裤、休闲裤或整洁得体的牛仔裤。Polo衫或礼服衬衫	套装，不打领带（可带一条领带以防万一）
女 性	卡其裤、休闲裤或整洁得体的牛仔裤。大方得体的上衣或毛衣	卡其裤、休闲裤或整洁得体的牛仔裤。大方得体的上衣或毛衣	套装，或得体的休闲裤配整洁的上衣

这些只是指导建议，具体还要参考公司的企业文化。此外，如果你应聘的是项目经理、开发主管或其他管理层职位，面试时最好还是穿得正式一点。

## 1.7 十大常见错误

### 错误一：只在计算机上练习

如果你正准备参加海洋游泳比赛，你会只在泳池里练习吗？应该不会。你得去体验大风大浪及海洋里各种情况带来的影响。所以，你肯定会希望到海洋中实地训练。

在计算机上借助编译器演练面试题就像只在泳池里练习一样。抛开这个环境吧，让我们拿出纸和笔。你可以在写好全部代码并做过人工测试之后，再在计算机上用编译器进行验证。

### 错误二：不做行为面试题演练

很多求职者将全部时间花在演练技术问题上，而忽视了行为面试题。你猜怎么着？面试官可是两者都会考查的。

而且不止于此，你回答行为问题的表现其实还会左右面试官对你技术能力的看法。行为问题的准备工作其实相对比较轻松，而且容易达到事半功倍的效果。用心回顾你以往的项目和经历，然后准备一些小故事。

### 错误三：不做模拟面试训练

假设你要准备一场重大演讲，所有同事和相关人员都将列席，而且它还关乎你的未来。要是只在头脑里无声地练习演讲，到了真正演讲时，你肯定会发狂的。

光是纸上谈兵，不做模拟面试也会陷入同样的境地。如果你是一名工程师，肯定认识不少同行。不妨找个朋友帮你做模拟面试。作为回报，你也可以给他当一回面试官。