

FOURTH EDITION



Programming Interviews Exposed

Coding Your Way Through the Interview

John Mongan, Noah Kindler, Eric Giguère

PROGRAMMING INTERVIEWS EXPOSED

PREFACE	xxv
INTRODUCTION	xxix
CHAPTER 1 Before the Search	1
CHAPTER 2 The Job Application Process	9
CHAPTER 3 The Phone Screen	19
CHAPTER 4 Approaches to Programming Problems	29
CHAPTER 5 Linked Lists	41
CHAPTER 6 Trees and Graphs	75
CHAPTER 7 Arrays and Strings	99
CHAPTER 8 Recursion	125
CHAPTER 9 Sorting	143
CHAPTER 10 Concurrency	165
CHAPTER 11 Object-Oriented Programming	181
CHAPTER 12 Design Patterns	193
CHAPTER 13 Databases	203
CHAPTER 14 Graphics and Bit Manipulation	223
CHAPTER 15 Data Science, Random Numbers, and Statistics	239
CHAPTER 16 Counting, Measuring, and Ordering Puzzles	259
CHAPTER 17 Graphical and Spatial Puzzles	275
CHAPTER 18 Knowledge-Based Questions	289
CHAPTER 19 Nontechnical Questions	303
APPENDIX Résumés	313
INDEX	333

Programming Interviews Exposed

Fourth Edition

Programming Interviews Exposed

CODING YOUR WAY THROUGH THE INTERVIEW

Fourth Edition

John Mongan

Noah Kindler

Eric Giguère



Programming Interviews Exposed: Coding Your Way Through the Interview, Fourth Edition

Published by
John Wiley & Sons, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Copyright © 2018 by John Wiley & Sons, Inc., Indianapolis, Indiana

Published by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-119-41847-4
ISBN: 978-1-119-41849-8 (ebk)
ISBN: 978-1-119-41848-1 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2018935416

Trademarks: Wiley, the Wiley logo, Wrox, the Wrox logo, Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book.

*To Thuy, the love of my life, who understands me, and
Calvin, who lights up my days.*

—JOHN MONGAN

To Mikey, Alex, Teddy, and Andy.

—NOAH KINDLER

*To my parents, Jean-Claude and Marie-Joëlle, who
encouraged and supported my love of programming.*

—ERIC GIGUÈRE

ABOUT THE AUTHORS

JOHN MONGAN is a self-taught programmer with professional experience as a consultant for several software and pharmaceutical companies. He has three patents on software testing technologies. He holds an MD and a PhD in bioinformatics from UC San Diego, where he worked on supercomputer simulations of protein dynamics. He is currently Assistant Professor and Vice Chair, Informatics of the Department of Radiology and Biomedical Imaging at UC San Francisco. His research focuses on applications of machine learning to radiological data and computerized clinical decision support.

NOAH KINDLER is VP Technology at the security technology company Avira. He leads software design and development teams across several products with a user base of over 100 million.

ERIC GIGUÈRE started programming in BASIC on a Commodore VIC-20 (a long time ago) and was hooked. He holds BMath and MMath degrees in computer science from the University of Waterloo, has extensive professional programming experience, and is the author of several programming books. He currently works as a staff software engineer at Google.

ABOUT THE TECHNICAL EDITORS

WAYNE HEYM, PhD, is a Senior Lecturer in the Department of Computer Science and Engineering for The Ohio State University's College of Engineering. He also collaborates with their Reusable Software Research Group (RSRG). He maintains a strong interest in RSRG's development discipline and language, Reusable Software Language with Verifiability and Efficiency (RESOLVE). He enjoys introducing beginning programmers to the wonders in the art and science of computer programming. He also likes leading programmers into the rich and satisfying realm of the theoretical foundations of computer science.

DAN HILL is a software engineer and software development manager with over 15 years of experience, working on projects that include web development, user interface design, back-end system architecture, databases, security and cryptography, and mobile app development. He has worked for Silicon Valley startups as well as larger technology companies, and has conducted countless programming interviews. He holds BS and MS degrees in computer science from Stanford University.

CREDITS

PROJECT EDITOR

Adaobi Obi Tulton

TECHNICAL EDITORS

Wayne Heym and Dan Hill

PRODUCTION EDITOR

Barath Kumar Rajasekaran

COPY EDITOR

Kimberly A. Cofer

PRODUCTION MANAGER

Katie Wisor

**MANAGER OF CONTENT ENABLEMENT
AND OPERATIONS**

Pete Gaughan

MARKETING MANAGER

Christie Hilbrich

BUSINESS MANAGER

Amy Knies

EXECUTIVE EDITOR

Jim Minatel

PROJECT COORDINATOR, COVER

Brent Savage

PROOFREADER

Nancy Bell

INDEXER

Estalita M. Slivoskey

COVER DESIGNER

Wiley

COVER IMAGE

© Paul Bradbury/Getty Images

ACKNOWLEDGMENTS

WE DEEPLY APPRECIATE the efforts of our colleagues at Wiley and Serendipity23 Editorial Services in bringing this revised fourth edition to fruition. The contributions of our project editor, Adaobi Obi Tulton, whose deft edits, organization, and persistence kept us on track, and the personal attention of our executive editor, Jim Minatel, were especially key, and we thank them for their time, work, and assistance.

The quality of this edition has been greatly improved by the work of our technical editors, Wayne Heym and Dan Hill, both of whom have already made important contributions to prior editions. Their thoughtful comments and meticulous review have eliminated numerous errors and oversights and immeasurably improved the clarity of the book. We thank them for their extensive work. We are also grateful to Andrew Taylor for additional review of the new data science material, and Tom Mongan for assistance with proofreading.

No fourth edition would have been possible without the three that preceded it, and the many people who contributed to them. John is particularly grateful for Michael J. Mongan's help in facilitating his participation with the third edition. We thank our third edition editor Maureen Spears, who swiftly and surely overcame the unique challenges that arose in preparation of that edition. Additionally, we thank our original editors, Margaret Hendrey and Marjorie Spencer, for their patience and helpfulness. We are also grateful to our original reviewers and advisors, Dan Hill, Elise Lipkowitz, Charity Lu, Rob Maguire, and Tom Mongan. Dan's contributions in particular were tremendous—the quality of the first edition was vastly improved by his careful and detailed reviews.

CONTENTS

<i>PREFACE</i>	<i>xxv</i>
<i>INTRODUCTION</i>	<i>xxix</i>
CHAPTER 1: BEFORE THE SEARCH	1
Know Yourself	1
Know the Market	3
Basic Market Information	3
What About Outsourcing?	4
Develop Marketable Skills	5
Get Things Done	6
Manage Your Online Profile	7
Summary	8
CHAPTER 2: THE JOB APPLICATION PROCESS	9
Finding and Contacting Companies	9
Finding Companies	9
Getting Referrals	10
Working with Headhunters	10
Contacting the Company Directly	11
Job Fairs	12
Technology-Driven Sites	12
The Interview Process	12
Screening Interviews	12
On-Site Interviews	13
Dress	14
A Recruiter's Role	14
Offers and Negotiation	15
Dealing with Recruiter Pressures	15
Negotiating Your Salary	15
Accepting and Rejecting Offers	17
Summary	17
CHAPTER 3: THE PHONE SCREEN	19
Understanding Phone Screens	19
Phone Screens by Software Engineers	19
Phone Screens by Nontechnical People	20

How to Take a Phone Screen	21
Phone Screen Problems	22
Memory Allocation in C	22
Recursion Trade-Offs	22
Mobile Programming	23
FizzBuzz	23
Reversing a String	24
Removing Duplicates	25
Nested Parentheses	26
Summary	27
CHAPTER 4: APPROACHES TO PROGRAMMING PROBLEMS	29
<hr/>	
The Process	29
The Scenario	29
The Problems	30
Which Languages to Use	30
Interactivity Is Key	31
Solving the Problems	32
The Basic Steps	32
When You Get Stuck	34
Analyzing Your Solution	34
Big- <i>O</i> Analysis	35
How Big- <i>O</i> Analysis Works	36
Best, Average, and Worst Cases	37
Optimizations and Big- <i>O</i> Analysis	37
How to Do Big- <i>O</i> Analysis	38
Which Algorithm Is Better?	38
Memory Footprint Analysis	39
Summary	40
CHAPTER 5: LINKED LISTS	41
<hr/>	
Why Linked Lists?	41
Kinds of Linked Lists	42
Singly Linked Lists	42
Doubly Linked Lists	44
Circular Linked Lists	44
Basic Linked List Operations	44
Tracking the Head Element	44
Traversing a List	46
Inserting and Deleting Elements	46

Linked List Problems	48
Stack Implementation	48
Maintain Linked List Tail Pointer	54
Bugs in removeHead	60
Mth-to-Last Element of a Linked List	62
List Flattening	65
List Unflattening	68
Null or Cycle	70
Summary	73
CHAPTER 6: TREES AND GRAPHS	75
<hr/>	
Trees	75
Binary Trees	77
Binary Search Trees	78
Heaps	80
Common Searches	80
Breadth-First Search	80
Depth-First Search	81
Traversals	81
Graphs	82
Tree and Graph Problems	83
Height of a Tree	83
Preorder Traversal	84
Preorder Traversal, No Recursion	85
Lowest Common Ancestor	87
Binary Tree to Heap	88
Unbalanced Binary Search Tree	91
Six Degrees of Kevin Bacon	93
Summary	97
CHAPTER 7: ARRAYS AND STRINGS	99
<hr/>	
Arrays	99
C and C++	100
Java	101
C#	102
JavaScript	102
Strings	102
C	103
C++	104
Java	104

C#	105
JavaScript	105
Array and String Problems	105
Find the First Nonrepeated Character	106
Remove Specified Characters	109
Reverse Words	112
Integer/String Conversions	116
From String to Integer	116
From Integer to String	118
UTF-8 String Validation	121
Summary	124
CHAPTER 8: RECURSION	125
<hr/>	
Understanding Recursion	125
Recursion Problems	129
Binary Search	129
Permutations of a String	131
Combinations of a String	134
Telephone Words	137
Summary	142
CHAPTER 9: SORTING	143
<hr/>	
Sorting Algorithms	143
Selection Sort	144
Insertion Sort	145
Quicksort	146
Merge Sort	148
Sorting Problems	149
The Best Sorting Algorithm	150
Stable Selection Sort	153
Multi-Key Sort	155
Make a Sort Stable	156
Optimized Quicksort	158
Pancake Sorting	161
Summary	163
CHAPTER 10: CONCURRENCY	165
<hr/>	
Basic Thread Concepts	165
Threads	165
System Threads versus User Threads	166

Monitors and Semaphores	166
Deadlocks	167
A Threading Example	168
Concurrency Problems	170
Busy Waiting	170
Producer/Consumer	172
The Dining Philosophers	175
Summary	179
CHAPTER 11: OBJECT-ORIENTED PROGRAMMING	181
<hr/>	
Fundamentals	181
Classes and Objects	181
Construction and Destruction	182
Inheritance and Polymorphism	183
Object-Oriented Programming Problems	184
Interfaces and Abstract Classes	184
Virtual Methods	186
Multiple Inheritance	188
Resource Management	189
Summary	191
CHAPTER 12: DESIGN PATTERNS	193
<hr/>	
What Are Design Patterns?	193
Why Use Design Patterns?	193
Design Patterns in Interviews	194
Common Design Patterns	194
Singleton	195
Builder	195
Iterator	197
Observer	197
Decorator	197
Design Pattern Problems	198
Singleton Implementation	198
Decorator versus Inheritance	201
Efficient Observer Updates	202
Summary	202
CHAPTER 13: DATABASES	203
<hr/>	
Database Fundamentals	203
Relational Databases	203

SQL	204
NoSQL	208
Object Databases	209
Hybrid Key-Value/Column Databases	209
Database Transactions	210
Distributed Databases	211
Database Problems	212
Simple SQL	212
Company and Employee Database	212
Max, No Aggregates	215
Three-Valued Logic	216
School Schemata	218
Summary	222
CHAPTER 14: GRAPHICS AND BIT MANIPULATION	223
Graphics	223
Bit Manipulation	224
Binary Two's Complement Notation	224
Bitwise Operators	225
Optimizing with Shifts	226
Graphics Problems	226
Eighth of a Circle	227
Rectangle Overlap	229
Bit Manipulation Problems	232
Big-Endian or Little-Endian	233
Number of Ones	235
Summary	237
CHAPTER 15: DATA SCIENCE, RANDOM NUMBERS, AND STATISTICS	239
Probability and Statistics	240
Descriptive and Inferential Statistics	241
Confidence Intervals	242
Statistical Tests	242
Artificial Intelligence and Machine Learning	244
Random Number Generators	245
Data Science, Random Number and Statistical Problems	246
Irreproducible Results	247
Study More; Know Less	249
Roll the Dice	251
Calculate Pi	254
Summary	258

CHAPTER 16: COUNTING, MEASURING, AND ORDERING PUZZLES 259

Tackling Brainteasers	259
Beware of Assumptions	260
Don't Be Intimidated	261
Beware of Simple Problems	262
Estimation Problems	262
Brainteaser Problems	263
Count Open Lockers	263
Three Switches	265
Bridge Crossing	266
Heavy Marble	269
Number of American Gas Stations	273
Summary	274

CHAPTER 17: GRAPHICAL AND SPATIAL PUZZLES 275

Draw It First	275
Graphical and Spatial Problems	276
Boat and Pier	276
Counting Cubes	278
The Fox and the Duck	282
Burning Fuses	283
Escaping the Train	286
Summary	287

CHAPTER 18: KNOWLEDGE-BASED QUESTIONS 289

Preparation	289
Problems	290
C++ versus Java	291
Friend Classes	292
Argument Passing	292
Macros and Inline Functions	294
Inheritance	295
Garbage Collection	296
32-Bit versus 64-Bit Applications	297
Network Performance	298
Web Application Security	298
Cryptography	301
Hash Tables versus Binary Search Trees	301
MapReduce	302
Summary	302

CHAPTER 19: NONTECHNICAL QUESTIONS	303
Why Nontechnical Questions?	303
Questions	304
“What Do You Want to Do?”	304
“What Is Your Favorite Programming Language?”	305
“What Is Your Work Style?”	306
“What Can You Tell Me About Your Experience?”	306
“What Are Your Career Goals?”	306
“Why Are You Looking to Change Jobs?”	306
“What Salary Are You Expecting?”	307
“What Is Your Salary History?”	310
“Why Should We Hire You?”	310
“Why Do You Want to Work for This Company?”	311
“Do You Have Any Questions for Me?”	311
Summary	311
APPENDIX: RÉSUMÉS	313
The Technical Résumé	313
A Poor Example	313
Sell Yourself	317
Keep It Short	317
List the Right Information	318
Be Clear and Concise	319
Relevant Information Only	320
Use Reverse Chronological Ordering	321
Always Proofread	321
An Improved Example	321
Managers and Senior Developers	323
Tailor the Résumé to the Position	329
Sample Résumé	329
INDEX	333

PREFACE

Solving problems that are presented in programming interviews requires a separate skillset from what you need to be a good programmer. Just like anything else, you probably won't be very good at this when you first start, but you can develop and improve your skills just as we did. This book is the first step in that process; through this book we leverage your programming expertise to rapidly turn you into an expert at programming interviews.

Since the first edition, *Programming Interviews Exposed* has effectively established a new topic area of programming books, and now a multitude of websites, blogs, and forums provide advice and sample questions. With all that available, why should you invest your time and money in this book?

Our focus continues to be on teaching you the techniques and approaches you need to be successful in programming interviews. We reinforce these by illustrating the thought process that leads to the solution of each of the problems we present, and show you how to move forward when you're stuck. These skills overlap with general coding skills, but they're not the same; we've seen great coders crash and burn in programming interviews because they haven't developed their interview skills. Early in our careers we crashed and burned a couple times ourselves, but you can avoid that by beginning your preparation with this book. Once you've learned the skills taught in this book you'll continue to learn by applying them to the problems you find in other books and on the web, but this is the book you want to start with.

One thing that never changes is that to become good at solving programming interview questions, you have to do more than passively read about them: you need to practice them. You'll get a lot more out of this book if you work out as much of each solution as you can on your own before you read about it.

Although the content of the book has expanded significantly since the first edition and the languages employed have shifted, we've stayed true to the goals and approach we set out then, described in the original preface, which follows.

PREFACE TO THE FIRST EDITION

If you're like us, you don't usually read prefaces. This one has some useful information in it, though, so we hope you'll make an exception. If you're still tempted to skip the preface, here's what you really need to know: You'll get as much out of this book as you put into it. If you read this book cover to cover, you'll learn something, but not nearly as much as you would if you take some time trying to work through the problems on your own before you read the answers.

This book will help prepare you for the interviews you will face when seeking a job in programming, development, technical consulting, or any other field that warrants a programming interview. Programming interviews bear little resemblance to those described in traditional job-hunting and interview books. They consist almost entirely of programming problems, puzzles, and technical

questions about computers. This book discusses each of the kinds of problems you are likely to encounter and illustrates how they are best approached using questions from real interviews as examples.

At this point you may be wondering who we are and what gives us the authority to write this book. We're both recent graduates who've been through a lot of interviews in the past few years. We've interviewed for jobs ranging from technical consulting with large established companies to writing device drivers for startups. This book is based on the experiences and observations we've taken from those interviews—what yielded offers and what didn't. We believe that this is the best possible basis for a book like this. Rather than give you some HR exec's idea of how interviewing should be done or a head hunter's impression of how it might be done, we will tell you what interviews are really like at America's top software and computer companies and what you need to do to get the job you want.

NOTE *For the record, we don't think that the way interviewing is done today is necessarily the way it should be done. The current paradigm puts too much emphasis on the ability to solve puzzles and familiarity with a relatively limited body of knowledge, and it generally fails to measure a lot of the skills that are critical to success in industry.*

To that end, we haven't made up any of the questions in this book. Every last one of them has been lifted from a recent interview. The distributions of problem type and difficulty are similar to what you should expect to encounter in your interviews. We must emphasize that the problems presented in this book are a representative sample of the questions asked in interviews, not a comprehensive compilation. Reading this book straight through and memorizing the answers would completely miss the point. You may be asked some of the questions that appear in this book, but you should not expect that. A large and constantly changing body of questions is asked, and any intelligent interviewer who has seen this book will never again use any of the questions that appear here. On the other hand, interview questions encompass relatively few topic areas and types of questions, and these rarely change. If you work on learning to solve not just the specific problems we present, but the types of problems we present, you'll be able to handle anything they throw at you in an interview.

We've taken a couple of steps to facilitate the objective of improving your problem-solving skills. First, where appropriate, we provide reviews of important topics before we present questions on those topics. Second, instead of merely giving answers to the problems, we illustrate the problem-solving process from beginning to solution. We've found that most textbooks and nearly all puzzle books take a different approach to examples: they begin with a problem, go immediately to the answer, and then explain why the answer is correct. In our experience, the result is that the reader may understand the particular answer and why it's right, but is left with no clue as to how the author came up with that solution or how a similar problem might be solved. We hope that our step-by-step approach to solutions will address this issue, helping you to understand not only the answers but also how you arrive at the answers.

Learning by watching is never as effective as learning by doing. If you want to get the most out of this book, you will have to work out the problems yourself. We suggest the following method:

1. After you read a problem, put the book down and try to work out the solution.
2. If you get stuck, start reading the solution. We never blurt out the answer at the beginning, so you don't have to worry that we're going to give away the entire solution.
3. Read just far enough to get the hint you need, and then put down the book and keep working.
4. Repeat this as necessary.

The more of the solution you work out yourself, the better your understanding will be. In addition, this method closely resembles the actual interview experience, where you will have to solve the problems yourself, but the interviewer will give you hints when you get stuck.

Programming is a difficult and technical art. It would be impossible to teach everything you need to know about computers and programming in one book. Therefore, we've had to make some assumptions about who you are. We assume that you have a background in computers equivalent to at least the first year or two of a computer science degree. Specifically, we expect that you are comfortable with programming in C, that you've had some experience with object-oriented programming in C++ or perhaps Java, and that you know the fundamentals of computer architecture and computer science theory. These are effectively the minimum requirements for a general development job, so most interviewers will have similar expectations. If you find yourself lacking in any of these areas, you should seriously consider seeking more education before starting your job search and interviews.

It's also possible that you have a great deal more computer knowledge and experience than what we've described as the minimum requirements. If so, you may be particularly interested in some of the more advanced topics included. However, don't ignore the basic topics and questions, no matter how much experience you have. Interviewers tend to start with the fundamentals regardless of what's on your résumé.

We have made every effort to ensure that all of the information in this book is correct. All of the code has been compiled and tested. Nevertheless, as you probably know all too well from your own programs, a few bugs and errors are inevitable. As we become aware of such problems, we will post corrections.

We're confident that you'll find this book useful in getting the job you want. We hope that you may also find it an entertaining exploration of some clever puzzles in your chosen profession. If you'd like to tell us about your reaction to our book, share your thoughts on any particular problem or topic, or provide a problem from one of your recent interviews, we'd love to hear from you.

Go find a killer job!

