Coding Interviews

Questions, Analysis & Solutions

Harry He

Apress®

Coding Interviews: Questions, Analysis & Solutions

ISBN 978-1-4302-4761-6

ISBN 978-1-4302-4762-3 (eBook)

President and Publisher: Paul Manning
Lead Editor: Saswata Mishra
Technical Reviewer: Jeffrey Pepper, Robert Hutchinson
Editorial Board: Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Louise Corrigan, Morgan Ertel, Jonathan Gennick, Jonathan Hassell, Robert Hutchinson, Michelle Lowman, James Markham, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Gwenan Spearing, Matt Wade, Tom Welsh
Coordinating Editor: Jill Balzano
Copy Editor: Ann Dickson
Compositor: Apress
Indexer: SPi Global
Artist: SPi Global
Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary materials referenced by the author in this text is available to readers at www.apress.com. For detailed information about how to locate your book's source code, go to www.apress.com/source-code/.

To my wife, Rachel, and our little boy, Lewis.

Contents at a Glance

Contents vii
About the Author xiii
Acknowledgments xv
Introduction xvii

CHAPTER 1: Interview Process 1

CHAPTER 2: Programming Languages 13

CHAPTER 3: Data Structures 33

CHAPTER 4: Algorithms 75

CHAPTER 5: High Quality Code 111

CHAPTER 6: Approaches to Solutions 143

CHAPTER 7: Optimization 187

CHAPTER 8: Skills for Interviews 219

CHAPTER 9: Interview Cases 263

Index 275

Table of Contents

Contents at a Glance v
About the Author xiii
Acknowledgments xv
Introduction xvii

CHAPTER 1: Interview Process 1
Types of Interviews 1
Phone Interviews 1
On-Site Interviews 3
Phases of Interviews 3
Behavior Interview 4
Technical Interview 7
Q/A Time 11
Summary 12

CHAPTER 2: Programming Languages 13
C 13
Palindrome Numbers 16
C++ 17
C++ Concepts 18
Analyzing Execution of C++ Code 18
Implementing a Class or Member Function in C++ 19
Assignment Operator 19
C# 22
Singleton 23
Java 27
Java Keywords 27

Data Containers 29
Thread Scheduler....30
Summary....32

CHAPTER 3: Data Structures....33
Arrays....33
Duplication in an Array....34
Search in a 2-D Matrix....37
String....42
Strings in C/C++....42
Strings in C#....43
Strings in Java....44
Replace Blanks in a String....45
String Matching....49
Linked Lists....53
Print Lists from Tail to Head....54
Sort Lists....56
Loop in List....59
Trees....63
Next Nodes in Binary Trees....64
Binary Search Tree Verification....66
Stack and Queue....70
Build a Queue with Two Stacks....70
Build a Stack with Two Queues....72
Summary....74

CHAPTER 4: Algorithms....75
Recursion and Iteration....75
Fibonacci Sequence....76
Search and Sort....81
Binary Search in Partially Sorted Arrays....84
Majorities in Arrays....87

Backtracking **90**
String Path in Matrix 91
Robot Move 93
Dynamic Programming and Greedy Algorithms **94**
Edit Distance 95
Minimal Number of Coins for Change 98
Minimal Times of Presses on Keyboards 99
Bit Operations **101**
Number of 1s in Binary 102
Numbers Occurring Only Once 105
Summary **109**

CHAPTER 5: High Quality Code **111**
Clearness **111**
Completeness **112**
Test Cases for Completeness 112
Strategies to Handle Errors 113
Power of Integers 114
Big Numbers as Strings 117
Delete Nodes from a List 123
Partition Numbers in Arrays 127
Robustness **132**
k^{th} Node from End 132
Reverse a List 135
Substructures in Trees 138
Summary **141**

CHAPTER 6: Approaches to Solutions **143**
Figures to Visualize Problems **143**
Mirror of Binary Trees 143
Print Matrix in Spiral Order 146
Clone Complex Lists 149

Examples to Simplify Problems 152
Stack with Min Function 152
Push and Pop Sequence of Stacks 157
Print Binary Trees Level by Level 159
Paths in Binary Trees 165
Divide and Conquer 168
Traversal Sequences and Binary Trees 168
Binary Search Trees and Double-Linked Lists 174
Permutation and Combination 179
Summary 185

CHAPTER 7: Optimization 187
Time Efficiency 187
Median in a Stream 188
Minimum *k* Numbers 191
Intersection of Sorted Arrays 194
Greatest Sum of Sub-Arrays 196
Digit 1 Appears in Sequence from 1 to *n* 198
Concatenate an Array to Get a Minimum Number 201
Space-Time Trade-Off 203
Ugly Numbers 204
Hash Tables for Characters 207
Reversed Pairs in Array 213
First Intersection Node in Two Lists 216
Summary 218

CHAPTER 8: Skills for Interviews 219
Communication and Learning Skills 219
Communications Skills 219
Learning Skills 220

Knowledge Migration Skill 220
Time of Occurrences in a Sorted Array 221
Application of Binary Tree Traversals 223
Sum in Sequences 227
Reversing Words and Rotating Strings 233
Maximum in a Queue 236
Mathematical Modeling Skill 241
Probabilities of Dice Points 241
Last Number in a Circle 243
Minimum Number of Moves to Sort Cards 246
Most Profit from Stock 249
Divergent Thinking Skills 251
Calculating 1+2+…+*n* 252
Implementation of +, -, *, and / 255
Final/Sealed Classes in C++ 259
Array Construction 261
Summary 262

CHAPTER 9: Interview Cases 263
Integer Value from a String 263
The Interviewer's Comments 267
Lowest Common Parent Node in a Tree 269
The Interviewer's Comments 273

Index 275

About the Author

Harry He has been a senior software engineer at Cisco since September 2010. His primary work involves development of Cesium, which is a platform for Cisco to monitor and control hardware quality of its partners (OEM/ODM). Prior to joining Cisco, Harry was associated with Autodesk and Microsoft for development of Civil 3D and Winforms respectively. Over the years, he has interviewed many candidates for different corporations where he developed his interest in coding interview questions. He has written dozens of blogs on this topic.

Harry's published works include a book on programming interview questions in Chinese, which was released in December 2011 with PHEI, China. He has exhaustive knowledge, experience, and understanding of code-related questions and interviews.

Acknowledgments

The prototype of this book is my blogs about coding interview problems. Thanks to the readers of the blogs, whose encouragement helped me make the decision to write this book.

Many friends and colleagues helped to review the first draft: Wesley Miao from Autodesk, Min Yang from Amazon, Aldrin Lee from Cisco, Jiakai Liu and Huai Wang from Facebook, Xiang Fan, Chao Tian, Pung Xu, and Bi Xue from Microsoft. They found many errors and made improvements that were invaluable additions to this text.

Thanks to the folks at Apress, who include (but are not limited to): Saswata Mishra, Jeffrey Pepper, Ann Dickson, and Jill Balzano. Their comments and revisions made this book much better.

Great thanks to my family members. My parents helped take care of the whole family. My father began to learn to cook in his 60s, and now he provides truly delicious meals. Moreover, I began to work on this book shortly after my little boy Lewis's birth. His smile and baby babble gave me great pleasure while going through the hard experience of writing a book. And, most of all, to Rachel, my wife and the love of my life. The day I finally found her was the best day in my memory and I am sure will always be so in the future.

Harry He
Shanghai, China
November 2012

Introduction

I used to be one of those who searched through the Internet to prepare for interviews of well-known companies. The information was scattered over lots of web sites, and it was not an easy task to collect coding interview problems and solutions systematically. In order to facilitate my own interview preparation, as well as others', I began to write blogs about programming problems and their solutions.

After I wrote dozens of blogs, I found that there were common strategies to solve various coding interview problems. Therefore, I gradually realized that it might be a good idea to summarize the strategies in a book. With one-year of writing and revising, as well many friends' encouragement and help, now this book is in your hands or perhaps on your screen.

Distinguishing Features

This book analyzes coding problems from interviewers' perspectives. There are many tips about the expected behaviors in this book, which are based on my own experiences as an interviewer at Autodesk, Microsoft, and Cisco. Moreover, many interview questions have different solutions. This book evaluates various solutions from an interviewer's point of view. When you read the problem analyses, you will get the idea as to why some solutions are better than others, and you will grasp the capabilities required to the assure the quality of your code through completeness, robustness, and efficiency.

This book not only solves more than 100 interview problems, but also summarizes common strategies to conquer complex problems. When I analyzed and solved dozens of coding interview problems, I found that there are many general strategies that are quite helpful to solve other similar problems during interviews. For example, if an interview problem is quite complex, we may divide it into several small subproblems, and then solve the subproblems recursively. We can also utilize hash tables implemented with arrays to solve many interview problems about strings. Similar problems are grouped in sections in this book. Pay attention to the similarities among problems in a section and the general ideas to solve them. When you meet new but similar problems at your interviews, you may reapply the strategies illustrated in this book.

Sample questions in this book are real interview problems frequently met in the famous IT companies. The coding interview is the most important phase of the whole interview process in many companies, such as Facebook, Google, and Microsoft. The sample questions collected in this book are the most typical ones adopted by interviewers in these companies. Don't be discouraged when you find that the problems in this book are not easy because interviews in big companies are not easy for most software engineers at first. You will find that there are relatively few problems that truly test the capabilities of programmers in meaningful ways. So, while you may not get a problem directly from this book, you should attain the skills required to handle whatever an interviewer can dish out. When you gradually master the strategies to solve problems summarized in this book, your capabilities to develop code and solve complex problems will be improved, and you will feel confident when interviewed by the Facebooks and Googles of the world.

Source code to solve sample interview problems along with a complete set of test cases to each problem is included. After candidates finish writing code, many interviewers will ask them to design some test cases to test their own code. Some candidates, especially fresh graduates, do not have clear ideas about how to design test cases. When you finish reading this book, you should know how to improve code quality with functional test cases, boundary test cases, performance test cases, and so on.

Summary of Chapters

The first chapter focuses on the interview process. A typical interview process can be divided into two phases: phone interviews (including phone-screen interviews) and on-site interviews. Usually there are three steps in each round of interview, which are the behavioral interview, technical interview, and general Q/A. Tips are provided for each stage of interviews.

The next three chapters cover basic programming knowledge. Technical interview questions on four popular programming languages (C, C++, C#, and Java) are discussed in Chapter 2. The most common data structures (including arrays, strings, lists, trees, stacks, and queues) and algorithms (including search, sort, backtracking, dynamic programming, greedy algorithms, and bit operations) are discussed in Chapter 3 and Chapter 4 respectively.

Chapter 5 discusses three factors of high quality code. Interviewers usually expect candidates' code to fulfill the functional requirements as well as cover corner cases and handle invalid inputs gracefully. After reading this chapter, you should get the idea so that you will write clear, complete, and robust code.

Three strategies to solve difficult coding interview problems are provided in Chapter 6. If hard problems are met during interviews, candidates should figure out solutions before they write code. After reading this chapter, you may get three strategies to solve problems: figures to visualize problems, step-by-step analysis on examples to simplify problems, and divide-and-conquer strategies to break complex problems into manageable pieces.

The topic of Chapter 7 is performance optimization. If there are multiple solutions to a problem, usually interviewers expect the most efficient one. The strategies to improve time efficiency and make trade-off decisions between time and space are discussed with several sample coding interview questions.

Chapter 8 summarizes various skills for interviews. Interviewers usually pay close attention to candidates' communication and learning skills. Additionally, many interviewers like to examine candidates' skills of reapplying knowledge, mathematical modeling, and divergent thinking.

Chapter 9 closes this book with two interview cases, which highlight good behavior expected by interviewers and the most common mistakes made by candidates.

Downloading the Code

The code for the examples shown in this book is available on the Apress web site, `www.apress.com`. A link can be found on the book's information page under the Source Code/Downloads tab. This tab is located underneath the Related Titles section of the page.