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.