

Introduction

Something's Wrong

We walked out of the hiring meeting frustrated, again. Of the ten “passable” candidates we reviewed that day, none would receive offers. Were we being too harsh, we wondered?

I, in particular, was disappointed. We had rejected one of *my* candidates. A former student. One who I had referred. He had a 3.73 GPA from the University of Washington, one of the best computer science schools in the world, and had done extensive work on open source projects. He was energetic. He was creative. He worked hard. He was sharp. He was a true geek, in all the best ways.

But, I had to agree with the rest of the committee: the data wasn't there. Even if my emphatic recommendation would sway them to reconsider, he would surely get rejected in the later stages of the hiring process. There were just too many red flags.

Though the interviewers generally believed that he was quite intelligent, he had struggled to develop good algorithms. Most successful candidates could fly through the first question, which was a twist on a well known problem, but he struggled to develop his algorithm. When he came up with one, he failed to consider solutions that optimized for other scenarios. Finally, when he began coding, he flew through the code with an initial solution, but it was riddled with mistakes that he then failed to catch. Though he wasn't the worst candidate we'd seen by any measure, he was far from meeting “the bar.” Rejected.

When he asked for feedback over the phone a couple of weeks later, I struggled with what to tell him. Be smarter? No, I knew he was brilliant. Be a better coder? No, his skills were on-par with some of the best I'd seen.

Like many motivated candidates, he had prepared extensively. He had read K&R's classic C book and he'd reviewed CLRS' famous algorithms textbook. He could describe in detail the myriad of ways of balancing a tree, and he could do things in C that no sane programmer should ever want to do.

I had to tell him the unfortunate truth: those books aren't enough. Academic books prepare you for fancy research, but they're not going to help you much in an interview. Why? I'll give you a hint: your interviewers haven't seen Red-Black Trees since *they* were in school either.

To crack the coding interview, you need to prepare with *real* interview questions. You must practice on *real* problems, and learn their patterns.

Cracking the Coding Interview is the result of my first-hand experience interviewing at top companies. It is the result of hundreds of conversations with candidates. It is the result of the thousands of candidate- and interviewer- contributed questions. And it's the result of seeing so many interview questions from so many firms. Enclosed in this book are 150 of the best interview questions, selected from thousands of potential problems.

My Approach

The focus of ***Cracking the Coding Interview*** is algorithm, coding and design questions. Why? Because while you can and will be asked behavioral questions, the answers will be as varied as your resume. Likewise, while many firms will ask so-called “trivia” questions (e.g., “What is a virtual function?”), the skills developed through practicing these questions are limited to very specific bits of knowledge. The book will briefly touch on some of these questions, to show you what they’re like, but I have chosen to allocate space where there’s more to learn.

My Passion

Teaching is my passion. I love helping people understand new concepts, and giving them tools so that they can excel in their passions.

My first experience “officially” teaching was in college at the University of Pennsylvania, when I became a teaching assistant for an undergraduate Computer Science course during my second year. I went on to TA for several other courses, and eventually launched my own CS course at the university focused on “hands-on” skills.

As an engineer at Google, training and mentoring “Nooglers” (yes, that’s really what they call new Google employees!) were some of the things I enjoyed most. I went on to use my “20% time” to teach two Computer Science courses at the University of Washington.

Cracking the Coding Interview and **CareerCup.com** reflect my passion for teaching. Even now, you can often find me “hanging out” at CareerCup.com, helping users who stop by for assistance.

Join us.

Gayle Laakmann

Behind the Scenes

For many candidates, interviewing is a bit of a black box. You walk in, you get pounded with questions from a variety of interviewers, and then somehow or other you return with an offer... or not.

Have you ever wondered:

- » How do decisions get made?
- » Do your interviewers talk to each other?
- » What does the company really care about?

Well, wonder no more!

CareerCup sought out interviewing experts from five top companies - Microsoft, Google, Amazon, Yahoo and Apple - to show you what really happens “behind the scenes.” These experts will walk us through a typical interview day and describe what’s taking place outside of the interviewing room, and what happens after you leave.

Our interviewing experts also told us what’s different about their interview process. From bar raisers (Amazon) to Hiring Committees (Google), each company has its own quirks. Knowing these idiosyncrasies will help you to react better to a super-tough interviewer, or to avoid being intimidated when two interviewers show up at the door (Apple!).

In addition, our specialists offered insight as to what their company stresses in their interviews. While almost all software firms care about coding and algorithms, some companies focus more than others on specific aspects of the interview. Whether this is because of the company’s technology or its history, now you’ll know what and how to prepare.

So, join us as we take you behind the scenes at Microsoft, Google, Amazon, Yahoo and Apple...

Behind the Scenes | The Microsoft Interview

Microsoft wants smart people. Geeks. People who are passionate about technology. You probably won't be tested on the ins and outs of C++ APIs, but you will be expected to write code on the board.

In a typical interview, you'll show up at Microsoft at some time in the morning and fill out initial paper work. You'll have a short interview with a recruiter where he or she will give you a sample question. Your recruiter is usually there to prep you, and not to grill you on technical questions. Be nice to your recruiter. Your recruiter can be your biggest advocate, even pushing to re-interview you if you stumbled on your first interview. They can fight for you to be hired - or not!

During the day, you'll do four or five interviews, often with two different teams. Unlike many companies, where you meet your interviewers in a conference room, you'll meet with your Microsoft interviewers in their office. This is a great time to look around and get a feel for the team culture.

Depending on the team, interviewers may or may not share their feedback on you with the rest of the interview loop.

When you complete your interviews with a team, you might speak with a hiring manager. If so, that's a great sign! It likely means that you passed the interviews with a particular team. It's now down to the hiring manager's decision.

You might get a decision that day, or it might be a week. After one week of no word from HR, send them a friendly email asking for a status update.

Definitely Prepare:

"Why do you want to work for Microsoft?"

In this question, Microsoft wants to see that you're passionate about technology. A great answer might be, "I've been using Microsoft software as long as I can remember, and I'm really impressed at how Microsoft manages to create a product that is universally excellent. For example, I've been using Visual Studio recently to learn game programming, and its APIs are excellent." Note how this shows a passion for technology!

What's Unique:

You'll only reach the hiring manager if you've done well, but if you do, that's a great sign!

Behind the Scenes | The Amazon Interview

Amazon's recruiting process usually begins with one or two phone screens in which you interview with a specific team. The engineer who interviews you will usually ask you to write simple code and read it aloud on the phone. They will ask a broad set of questions to explore what areas of technology you're familiar with.

Next, you fly to Seattle for four or five interviews with one or two teams which have selected you based on your resume and phone interviews. You will have to code on a whiteboard, and some interviewers will stress other skills. Interviewers are each assigned a specific area to probe and may seem very different from each other. They can not see other feedback until they have submitted their own and they are discouraged from discussing it until the hiring meeting.

Amazon's "bar raiser" interviewer is charged with keeping the interview bar high. They attend special training and will interview candidates outside their group in order to balance out the group itself. If one interview seems significantly harder and different, that's most likely the bar raiser. This person has both significant experience with interviews and veto power in the hiring decision. You will meet with your recruiter at the end of the day.

Once your interviewers have entered their feedback, they will meet to discuss it. They will be the people making the hiring decision.

While Amazon's recruiters are excellent at following up with candidates, occasionally there are delays. If you haven't heard from Amazon within a week, we recommend a friendly email.

Definitely Prepare:

Amazon is a web-based company, and that means they care about scale. Make sure you prepare for questions in "Large Scale." You don't need a background in distributed systems to answer these questions. See our recommendations in the System Design and Memory Limits Chapter.

Additionally, Amazon tends to ask a lot of questions about object oriented design. Check out the Object Oriented Design chapter for sample questions and suggestions.

What's Unique:

The Bar Raiser, who is brought in from a different team to keep the bar high.

Behind the Scenes | The Google Interview

There are many scary stories floating around about Google interviews, but it's mostly just that: stories. The interview is not terribly different from Microsoft's or Amazon's. However, because Google HR can be a little disorganized, we recommend being proactive in communication.

A Google engineer performs the first phone screen, so expect tough technical questions. On your on-site interview, you'll interview with four to six people, one of whom will be a lunch interviewer. Interviewer feedback is kept confidential from the other interviewers, so you can be assured that you enter each interview with a blank slate. Your lunch interviewer doesn't submit feedback, so this is a great opportunity to ask honest questions.

Written feedback is submitted to a hiring committee of engineers to make a hire/no-hire recommendation. Feedback is typically broken down into four categories (Analytical Ability, Coding, Experience and Communication) and you are given a score from 1.0 to 4.0 overall.

The hiring committee understands that you can't be expected to excel in every interview, but if multiple people raise the same red flag (arrogance, poor coding skills, etc), that can disqualify you. A hiring committee typically wants to see one interviewer who is an "enthusiastic endorser." In other words, a packet with scores of 3.6, 3.1, 3.1 and 2.6 is better than all 3.1s. Your phone screen is usually not a strong factor in the final decision.

The Google hiring process can be slow. If you don't hear back within one week, politely ask your recruiter for an update. A lack of response says nothing about your performance.

Definitely Prepare:

As a web-based company, Google cares about how to design a scalable system. So, make sure you prepare for questions from "System Design and Memory Limits" Additionally, many Google interviewers will ask questions involving Bit Manipulation, so please brush up on these questions.

What's Different:

Your interviewers do not make the hiring decision. Rather, they enter feedback which is passed to a hiring committee. The hiring committee recommends a decision which can be—though rarely is—rejected by Google executives.