SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING
CAREER SERVICES

Guide to Preparing for Technical Interviews

How to solve any technical interview coding question:

1. Clarify question
2. Determine inputs and outputs
3. Determine edge cases
4. Determine the brute force solution
5. Optimize your solution BUD
    a. Bottle necks?
    b. Unnecessary work?
    c. Duplicate work?
6. Test your solution
7. Code your solution
8. Walkthrough and explain your solution

Candidates should prep the following before an interview:

1. Review core algorithms and CS skills.
2. Practice writing code with sample questions, especially if you are out of practice; focus on your strongest language.
    a. Practice on a white board or paper.
    b. Don't use pseudo code. Interviews will want to see you use actual code.
    c. Write in the language in which you feel the most comfortable. If you feel comfortable with a few languages, use the one your interviewer also knows. You're allowed to ask what your interviewer likes.
3. Know what technical skills are on your resume.
4. Practice doing the interview with a friend (especially for a phone interview) or someone from Career Services. Ask a friend or a Career Services staff member to critique you.
5. If you can't write it on a whiteboard, you don't know it. If an interviewer asks you if you're familiar with a sorting algorithm, don't say yes unless you can demonstrate it before her or him.

Interview process:

1. First round phone interview: This will be done with a recruiter and many of the questions will focus on what you want to do, interests, and career path. There aren't too many technical questions asked during this stage.
2. Second round face-to-face interviews: Expect to do multiple rounds of interviews with 2-4 interviewers. Expect to see more advanced technical questions during this round.

How to handle a 30-45 minute technical interview:

1. Short introduction: Be ready to talk about the technical skills you are comfortable with. This is not a time to exaggerate, but you should be confident in sharing your skills.
2. Assessment question: Answer these the best you can. Remember that it is more important to get the high-level correct answer than to write code perfectly, but perfect code is a plus.

3. Test your code: Stand back and make sure that what you wrote is correct. Show the interviewer that the performance and accuracy of your code is important to you.

Typical technical interviews will cover:

1. Object orientation – Do you know what an object is and how to use them?
2. Data structures – This isn't as complicated as it sounds and consists largely of string manipulations and array problems
3. Algorithmic structure – These include linked lists, bit manipulation, sorting and searching.
4. Testing – How would you test a website? A car?
5. Databases – Think SQL. What is a relational database?

Interviewers expect you to:

1. Be familiar with a language so you can talk about it and do a coding problem during the interview. You can use any language you want as long as you know how to code in it!
2. Explain your thought process and how you are solving the problem. You need to expect have a conversation with the interviewer (especially if it is a phone interview!). Do not remain quiet when trying to solve the problem. You should talk through the problems and have a conversation with the interviewer about both the benefits and disadvantages of different solutions to the problem.
3. Spend as long as you can getting to the whiteboard. Be very sure of what you are about to write. Remember: Until you know your *input* and your *output*, you are not ready to write code.
4. If you are stuck, start with a simple solution and then upgrade to a more complex solution. A great engineer is someone who knows a little about a lot, and a lot about a little: Be comfortable in the computer science industry but know that you are not expected to be an expert or to have a PhD. Bring the conversation back to what you know as often as you can.
5. Go beyond giving the interviewer what he or she asked. A rock star candidate talks about subtleties in design, data structures, and algorithms. They also offer a variety of alternatives in different situations.

What to do when you have seen a question before:

If you've heard a question before, admit this to your interviewer. Your interviewer is asking you these questions in order to evaluate your problem-solving skills. If you already know the question, then you aren't giving them the opportunity to evaluate you.

Additionally, your interviewer may find it highly dishonest if you don't reveal that you know the question and, conversely, you'll get big honesty points if you do reveal this (Gayle Laakmann McDowell CCI).

Additional tips:

1. Realize that the interviewer is not out to trick you!
2. Be relaxed and know how to calmly handle the situation/problem.
3. Be polite and sociable. The interviewers want to know if they can work with you.

Sample questions:

1. Can you describe your technical contributions to a project on your resume?
2. How would you sort a two dimensional array of integers on the 0th element?
3. Write a method to replace all the spaces in a string with "%20".
4. How would you test a car?

What to expect in interviews with the Big 4:

Google

The process for interns and full-time hires at Google is different. For interns, you will have two phone screenings and in each one you will be given 45 minutes to solve a coding problem through a shared text editor. If you get high scores from both of your interviews, the hiring manager will make a decision whether or not to hire you and then you will go into team matching phase. If you perform great on one interview and poorly on the other, you will be offered a third phone interview that will serve as the decider.

For full-time hires, if you pass the phone screenings, you will then be flown onsite to do four or five interviews with various engineers throughout Google, who will ask questions similar to the questions you had in your phone interviews. If you pass this and the hiring manager, you then move to the team matching phase. In the team matching phase, you will review various teams that have open spots within Google and pick the teams you are interested in. The teams will also do the same, and if you and a team have both chosen each other, then you will get an offer for that team.

Facebook

The process for interns and full-time hires is the same at Facebook. You will first have a phone screening, where you will be presented with a warmup question (5-10 minutes) then you will be given the core question (30 minutes). Questions for the screening interview will be almost identical to questions you find on Leetcode.com, so you are expected to solve them very fast and explain your thought process.

If you pass the phone screening, Facebook will fly you onsite to Menlo Park where you will have four or five interviews with various engineers throughout Facebook. Each engineer will have a different area they are grading you on: Design, Behavioral, or Coding. Each engineer them submits feedback to a hiring manager who will decide whether or not you are a good fit for Facebook.

Microsoft

Microsoft interviews interns and full-time hires in a similar way. You will have a phone screening interview if they like what they see on your resume. If you pass the phone screening they will fly you onsite for a day of interviews. During the day you will complete four or five interviews, often with different teams. If one of the teams you interviewed with is interested in having you as a member of their team, they will contact the hiring manager, who will make the final decision. You should hear back within a week of your interviews and if you haven't, you will want to send a friendly reminder to your recruiter.

Amazon

Amazon's recruiting process typically begins with a phone screen in which a candidate interviews with a specific team. A small portion of the time, a candidate may have two or more interviews, which can indicate either that one of their interviewers wasn't convinced or that they are being considered for a different team or profile. The engineer who interviews you will usually ask you to write simple code via a shared document editor. Next, you will fly to Seattle (or whichever office you're interviewing for) for four or five interviews with one or two teams that 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.

The "bar raiser" interviewer is charged with keeping the interview bar high. They attend special trainings 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. Remember, just because you seem to be struggling more in this interview doesn't mean you're actually doing worse. Your performance is judged relative to other candidates; it isn't evaluated on a simple "percent correct" basis. Once your interviewers have entered their feedback, they will meet to discuss it. These are the individuals who will make the hiring decision. While Amazon's recruiters are usually excellent at following up with candidates, occasionally there are delays. If you haven't heard from Amazon within a week, you might consider sending them a friendly reminder email (Gayle Laakmann McDowell CCI).

Technical interviews with other companies:
If you are interviewing with any tech company, you can expect a technical interview process similar to the Big 4, unless the company is a startup. Often, startups have much more loosely defined interview processes and focus more on cultural fit. For non-tech companies, you can expect the technical interview to be fairly basic, with a focus on some entry level programming challenges like a palindrome checker, finding duplicates in an array, and FizzBuzz.

Resources for interview prep:

- Websites
  - HackerRank (great entry level website)
  - LeetCode (you might see these exact questions in interviews)
  - GeeksforGeeks (great explanations of complicated topics)
  - Pramp (allows you to practice phone interviews with real people)
- Books
  - Cracking the Coding Interview (great if you are short on time)
  - Introduction to Algorithms by CLRS (one of the most famous books in CS)
- Additional resources
  - Hangout on Air: Candidate coaching session—tech interviewing. Retrieved from http://www.google.com/about/jobs/lifeatgoogle/hangout-on-air-tech-interviewing.html
  - Moore, J. Hangout on air: Google recruiters share technical interview tips. Retrieved from http://www.google.com/intl/en/about/jobs/lifeatgoogle/hangouts-on-air-google-recruiters-sharetechnical-interview-tips.html