Technical Interview Tips

Suggestions from IST Peer Career Assistant, Ryan Feng ’19
IST: Design and Development

FORMAT

• Your interviewer will start with a behavioral question (Tell me about yourself, what are some projects you’ve worked on?).
• Then, the interviewer will ask you an easy level coding question.
  • You must write your work out for them.
  • In-person will be on a whiteboard.
  • Over the phone or on skype will be on a “Google doc-type” application where the interviewer watches what you’re typing.
• Your interviewer will give you clues and suggestions throughout the process.
• You should create some solution, even if you don’t fully write/code it out.
• If you finish soon, you may get more coding questions of increasing difficulty (easy -> hard).

PROCCESS FOR EACH CODING QUESTION

• Ask questions, gather background info.
• Think and describe a brute force solution. Don’t worry about test cases or optimizations yet.
• Write pseudo code. Don’t worry about syntax/spelling.
• Think of test cases where the pseudo code might break (ex. Out of bounds index error).
• Revise pseudo code.
• Start writing the actual code. Ask interviewer if you forget syntax (ex .toString() .charAt() ).
• Walk through test cases and edge cases:
  • What if the number of users is 1 million?
  • What if a string is null or 1 char long?
  • What if a string accidentally has a number or space in it?
  • Watch out for difference in length of string vs index count (String “Maria” has length 5 but index goes from 0-4).
• Improve time and space complexities:
  • Reduce number of for loops, especially if time complexity is O(n^2).
  • Try to reach O(n logn) time complexity.
  • Store calculations to avoid duplicate work.
STRATEGIES

• Ask questions right away. Gather background information and any constraints. The interviewer is supposed to give you clues and more info (collaborative).
• Think out loud! Show your thought process.
• Always think about the user and their situation.
• Stay organized and keep your writing neat.
• If you freeze-up or find your mind blanking:
  • Retrace your steps or discuss with your interviewer.
  • Go back to who are the users and what do they want?
• Treat you interviewer like a team-member. You are collaborating to create a solution.
• Your interviewer is not there to criticize you or make you mess up. You are demonstrating how you code in a group setting.
• Watch out for nested ‘for loop’ and greedy algorithms.

HOW TO PREPARE

• Use resources from companies:
  • https://news.microsoft.com/life/how-to-ace-a-technical-interview-at-microsoft/
  • https://www.facebook.com/careers/life/preparing-for-your-software-engineering-interview-at-facebook/
  • https://grouplens.org/blog/preparing-for-a-google-technical-interview/
  • https://www.geeksforgeeks.org/top-10-algorithms-in-interview-questions/

• Study data structures (Linked Lists, Binary Trees, Hash Tables):
  • Know why you should use one data structure over the other (searching, deletion, insertion speed).
  • Know the methods specific to each data structure (in-order, post-order traversal for binary trees).

• Study search algorithms (Breadth 1st search, Depth 1st search, Dijkstra’s algorithm).
• Study sorting algorithms (bubble sort, quick sort, heap sort).
• Study Time Complexity/Big-O notation and how to reduce it in your code.
• Study Space Complexity and how to reduce it in your code.
• Buy “Cracking the Coding Interview” and practice problems every week.
• Practice coding challenges on Hackerrank and LeetCode every week.
• Practice whiteboarding and coding questions with friends who also have technical interviews.
• Take every tech interview opportunity you can get, even if you think you are not prepared!
• Keep up with emerging tech news.