UBER Engineering

Backend Engineering Phone Screen Preparation Guide

Format and Expectations

Phone Interview Format

- Introductions (10 min)
- Coding Exercises (30-40 min)
    - Live coding exercise on HackerRank. Solve 1-2 technical exercises and write working code to implement the solution.
- Relevant Experience and Questions (10-15 min)

What to expect in our backend phone interviews

- General problem solving skills
- Architecture
- Common data structures and algorithms
- Software Design (OOP)
- Runtime analysis (Big-O)
- Open-Ended Discussion: Complex challenges faced, best/worst designs seen, performance analysis and optimization, testing, ideas for improving existing products

Tips

At Uber, we believe in collaboration and sharing ideas

- When coding, make sure you fully understand what's being asked before diving in. It's always OK to have the interviewer clarify.
- Always let your interviewer know what you're thinking, describe how you want to tackle solving each part of the question. He or she will be as interested in your thought process as your solution. If you're stuck, let your interviewer know. They may help clarify the problem to get you back on the right track.
- Make sure you talk through your process out loud. We're looking for clean, optimized code, and we want to know your way of getting there.
- We love candidates who show strong commitment and passion, as well as those who are team players. We want people to be excited about working at Uber, not just because of the company, but because they enjoy the product, or have ideas on how to make us better.

Interviewers will be looking at the approach to questions as much as the answer

- Does the candidate listen carefully and comprehend the question? (+)
- Are the correct questions asked before proceeding? (+)
- Are hints heard and heeded? (+)
- Does the candidate enjoy finding multiple solutions before choosing the best one? (+)
- Are new ideas and methods of tackling a problem sought? (+)

- Is the candidate inventive and flexible in their solutions and open to new ideas? (+)
- Can questioning move up to more complex problem solving? (+)
- Is brute force used to solve a problem? (-)
- Are things assumed without first checking? (-)

Backend Engineering Specifics

We expect our backend engineers to be proficient in algorithms, problem solving, and designing scalable fault tolerant architectures. This includes being an expert at your programming language of choice.

- Brush up on your Data Structures including: ArrayLists, Hash Tables, Trees, Graphs, Linked Lists, Stacks, and Queues. Potentially Heaps.
- Brush up on your Algorithms: Sorting, Breadth-First Search, Binary Search, Tree Insert/Find.
- Brush up on concepts like Big O and Recursion
- Avoid studying up on Tree Balancing, Memorization/Dynamic Programming,
- Heap Implementation, and Topological.

Remember to **ALWAYS** test your code.

Preparation Videos

Gayle Laakmann McDowell, author of Cracking the Coding Interview, has prepared a series of videos specifically for Uber candidates to help you prepare, walking you through the entire process step-by-step. Take the time to work through these: Gayle gives away the secrets to the whole process!

Practice Algorithmic Problem Solving

Uber interviews focus on algorithmic problem solving. While this may seem extraneous for front-end candidates, many of our web applications are complex enough to require solving this type of challenge in your day-to-day work. Even good candidates get tripped up because they are rusty or unfamiliar with this style of interview question. Set aside some time and practice answering some questions using HackerRank or Coderpad. Here are some online resources for this kind of question:

- Cracking the Coding Interview Github
- CareerCup Solutions
- LeetCode Questions

Read about this process firsthand from a recent hire to Uber Engineering!

- Why join Uber
- Engineering blog
- Uber Open Source