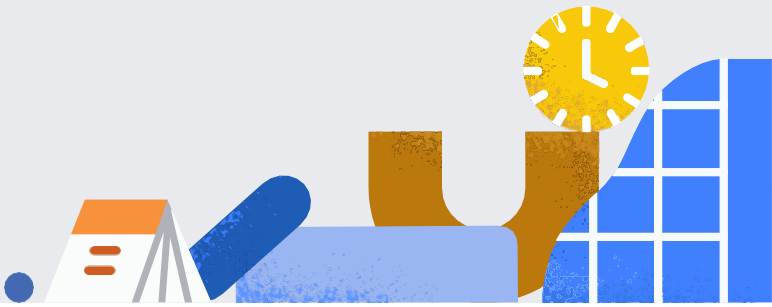
FACEBOOK

AI RESEARCH

Interview Guide



Tips to Keep in Mind

- Familiarize yourself with our 5 Core Values (Be Bold, Focus on Impact, Move Fast, Be Open, and Build Social Value). This is how we work together to make the world more open and connected. We look for people who believe in these values and practice them daily.
- Be yourself! Be open and honest about your successes and failures.
- Be humble and focus on teamwork, leadership, and mentorship qualities.

Welcome to your prep guide for your AI Research Internship or AI Research Engineering Internship interviews at the Facebook company. Our researchers and recruiters put together this guide, so you know what to expect and how to prepare.

Format

Agenda

Your conversations with our team members will be divided into the following time blocks:

- **One (or more) 45-minute** Research Interview/s.
- **One (or more) 45-minute** Coding Interview/s.

Research Interview

What can you expect?

The purpose of this interview is to explore the depth of your knowledge in your research area, usually focused on one of your recently published papers, the novel contributions of your work, evaluation methodology, how your contribution fits in the greater research landscape, and your ability to describe compelling extensions and new directions for future work.

This interview will probe your ability to solve Facebook challenges in an area adjacent to your area of expertise (e.g., NLP, CV, Speech, ML, Robotics, etc.).

Tips to keep in mind

A common mistake is to look at interview problems, recognize them, and sort of understand them, but not to be able to actually code them. So, you should prepare by practicing writing code by hand, without a computer. If you prepare by solving interview-style questions in a timed way, you'll increase the chance that your coding ability will come through during the interview process. Good luck!

Coding interview

What can you expect?

This interview will include having real-world ML skills, but also excelling at common tasks like calculating model metrics or algorithms for data cleanup. The interviewer will be looking for accurate, bug-free, fast, and well-thought-out code. They'll want to hear your thought process throughout, so be sure to provide a narrative as you go through the code. You're welcome to code in whatever language you feel most comfortable but choosing one that is going to assist in getting an optimal solution in the quickest and most efficient manner is key.

Not all interviewers follow the exact same breakdown, but the following is typical:

- **Introductions:** The first five minutes will be an introduction, possibly brief questions about your background.
- **Coding:** The next 30-35 minutes will be one or more coding problems.
- **Ask Us Anything:** We try to reserve the final five minutes for your questions for the interviewer. This part gives you a chance to learn more about Facebook from someone in Engineering and gives your interviewer a chance to learn more about your interests.

How to prep

You should be able to whiteboard solutions from simple to medium difficulty, programming interview questions in under 15 minutes. As one of the interviews is designed for ML practitioners, we'd assume basic familiarity with setting up ML projects, so we'll focus on bare minimum understanding needed to get you started on code.

To prepare, it's not enough to read through sample questions and recognize the concepts. You'll need to practice writing code without a computer, simulating a timed interview environment.
When you have a solution, review it and confirm it's something that you'd approve if it were submitted to you as a proposed part of your codebase. Make sure that it's correct, that you've taken into account the edge cases, it's efficient, and it clearly reflects the ideas that you're trying to express in your code.

In addition to reviewing the CS fundamentals, these tips may be helpful:

- **Understand the problem** you have to solve. It's okay to ask for clarifications or to talk through the problem.
- **Think about different algorithms** and algorithmic techniques (sorting, divide- and-conquer, dynamic programming/memorization, recursion).
- **Think about data structures,** particularly the ones used most often (Array, Stack /Queue, Hashset / Hashmap / Hashtable / Dictionary, Tree / Binary Tree, Heap, Graph, Bloom Filter, etc.).
- **Modifying the problem** or thinking about it in smaller pieces may be helpful.
- **Practice coding using a collaborative coding tool or on a whiteboard if you will be interviewing in person.**

Example questions

Each question has two aspects to it: one part that exercises the ability to solve a problem via code given a specification, and another that evaluates some basic understanding of common ML constructs. We want to see if you can come up with a working solution, while thinking about all the best practices associated with coding for production.

- Write a function that serializes and deserializes a matrix from a file.
- Write a function that calculates the precision and recall for a multi-class problem.

What we're looking for

- Ability to discuss the problem, while keeping in mind various real-world ML implications. For example, if we're talking about the serialization of a matrix problem, real-world scenarios like sparsity or extremely large dimensions would be good things to discuss.
- Ability to think through various alternative solutions to a problem and thinking through space / time complexity. This also includes discussing various data structures and being able to communicate the tradeoffs.
- Ability to convert this solution into a well-organized, executable code that you'd feel comfortable deploying to production (thinking about test cases, good APIs, edge cases).

Appendix / resources

Links to exercises, information, and guides to help you prepare.

Coding Practice

- **CareerCup**
- **CodeKata**
- **Codeforces**
- **HackerRank**
- **TopCoder**

Thanks for taking the time to review this guide and good luck in your interviews – you'll do great!