Dear fellow engineer,

I'm an engineer here at Facebook and I want to share with you some thoughts on our interview process.

Our interview process is designed around the idea that all programmers that we want to hire, with proper preparation, have the ability to quickly whiteboard solutions to questions that involve describing an algorithm and translating that description into working code. Doing well on this is a strong signal that you're able to understand how to write efficient algorithms, effectively problem-solve, and communicate your thoughts in code clearly.

Unfortunately, we find that many good engineers come to our interviews without preparing for these types of questions. While we think that any good engineer can with practice perform well on these types of coding questions, it's difficult to get a strong signal for someone who did not prepare. In these situations, we often want to hire the candidate but don't have a strong enough signal that the candidate will understand how to write efficient algorithms, effectively problem-solve, and communicate their thoughts in code. And, in the end, while we know many of these engineers could be great, we know that hiring people that cannot do these things effectively is painful for our engineering organization.

So, you should prepare for our coding interviews. I am constantly surprised by how often people come to our interviews underprepared. Even if you're not sure that you want to work at Facebook, you should prepare for your interviews and enter the process committed to doing well. It's usually not enough to read through sample questions and recognize the concepts. You need to prepare by practicing the actual writing of the code, simulating a timed interview environment.

To prepare, you should practice solving interview-style coding questions, without a computer (so with pen and paper, or with a whiteboard) in a timed environment where you give yourself 10-15 minutes to write out the solution to the question. When you have a solution, you should look at it and see if it's something that you would approve if it were submitted to you as a proposed part of your codebase. Make sure that it is correct, that you have taken into account the edge cases, that it is efficient, and that it clearly reflects the ideas that you're trying to express in your code.

You should be able to whiteboard solutions to simple and medium-difficulty programming interview questions in under 15 minutes. 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, just to be totally clear – 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!

Best,
Brian Rosenthal
Engineer, Facebook