interviews is that I only focused on the technical aspects. Yes, technical competency is a significant part of the interview, and rightly so, but these interviews are also largely designed to assess the candidate's communication, problem-solving, and testing abilities. Here's a breakdown of the four main coding signals that the interviewers are generally looking for.

Technical - Ability to quickly and correctly implement a working solution, be proficient in the chosen programming language, and write clean and straightforward code that follow the best coding practices like using meaningful names, avoiding duplications, etc. A book I'd highly recommend for the best software engineering practices and principles is Clean Code by Robert C. Martin.

Communication - Asking good clarifying questions and disambiguating the problem, explaining the overall approach and algorithm you are going to use, explaining the code while you are coding, and convincing the interviewer of the choices you are making. I am currently working on a list of all the clarifying questions that I commonly asked the interviewers and will post them shortly, in case you are interested.

Problem-solving - Approaching the problem systematically and logically, quickly understanding what the question is asking/requiring you to do, bringing unique prespective and thinking outside the box, proposing multiple solutions to the problem and discussing their trade-offs, having a thorough understanding of time and space complexity, and proactively pruning the code and improving the solution.