Coding Challenges Playbook

Estimated read: 10 min

Joseph Borodach | 4/26/23

Introduction

- Don't Forget: You Are Competing
    - Solving the problem does **not** mean you won.
        - Also, this playbook hyper focuses on one portion of the interview process, there are other factors to determining which candidates are chosen.
    - You are competing against the best, most ambitious candidates from around the world.

- This Is **A Playbook for Coding Challenges**
    - Even **though sports players have practiced the same plays over & over again**,
    - When it comes to game time: They are competing against the best & most ambitious players, **and they use a playbook**.
    - Below are the unspoken, but **expected steps** to crush interviews.

- The Goal of Interviews **& Leetcodes**
    - Is **NOT** to solve problems as **quickly** as possible.
    - But, to:
        - Solve problems optimally
            - From start to finish.
        - While engaging interviewer(s).

- How Do I Master Interviewing?
    - *By diminishing the gap between real interviews & leetcodes*.
    - Doing tons of mock interviews until they are not nerve racking.

- Background

    - I gathered these principles after speaking with 100s of engineers.
    - I designed this guide to help me prepare for interviews & perform well during interviews.
    - Most statements will be subjective & disagreement is inevitable.
    - Use your best judgment to decide what you think is correct.
    - Have comments, ideas, feedback - shoot me an email! Joseph Borodach

- General Tips
    - Collaborate - Explain your thought process.
    - Be open minded - Ask for feedback.
    - Most people go too quickly - Err on the side of going slowly.
        - Exception: If the company is notoriously known for wanting working code (i.e. Palantir, Uber) you have to keep an eye on the clock.

The Eleven Sequential Plays 🥇

1. Clarify Expectations Beforehand
    - “Is your priority **working code** or **optimal design & analysis**?”
2. Example
    - Go through an example.
    - This will:
        - Ensure you understand the problem.
        - Help you approach the problem.
        - Avoid wasting time solving the wrong problem.
        - Engage the interviewer.
        - Illustrate communication skills.
3. Observations & Clarifying Questions 🙋
    - Identify key traits & characteristics of the problem.
    - Ask clarifying questions. This will:
        - Help approach the problem.
        - Engage the interviewer(s).

- For example
    - “How does the program accept the input?”

4. Think of Solutions - Take Your Time! 🧘

- Recall - There are 2 goals
    - Solve the problem optimally.
    - Engage the interviewer(s).
- Therefore
    - This is **not** when you want to rush.
        - The hardest part of the interview is coming up with solutions!
    - **Ask for time** to think through some ideas & come up with solutions.
        - This shows intelligence, confidence, & maturity.
    - Then, prepare to **articulate** them to the interviewer(s).
        - You don’t want them to think you’re a genius who doesn’t know how to articulate themself.
- You can take time throughout the interview
    - I personally find that once you speak, the ball starts rolling & it is hard to go back to come up with better solutions.

5. Brute Force Solution 🪠

- Purpose a Brute Force solution.
- Time complexity.
- Space complexity.

6. Optimal Solutions 🔥

- Develop Better Solutions.

7. Design The Best Solution You Have

- Write Pseudo Code.
    - In an interview, you can speak it.
- **Go through ≥ 1 example - Illustrate the solution works**. This will:
    - Help write the actual logic of the solution.
    - Avoid wasting time writing an incorrect solution.
    - Engage the interviewer(s).

8. Write The Solution

- **Use extra informative variables**

        - i.e. write what ‘i’ and ‘j’ represent.
        - This will **help the interviewer understand your code**.
            - Interviewer(s) are not invested in understanding your code.
        - **Make it as easy for them as possible**.
    - **Ask for feedback**
        - “Should I proceed to implement this solution?”
    - **Explain each line of logic before typing it**
        - “I’m gonna type the for-loop structure now”
    - **Speak as you type**
        - This **does not** need to be perfect: It’s just to keep the interviewer engaged & show you know how to work well with others.

9. Analyze The Solution

- **Go through ≥ 1 example → Illustrate the code works**. This will:
    - Help write the actual logic of the solution.
    - Engage the interviewer(s).
- Time.
- Space.
- Edge cases.
- Identify Key Characteristics: In place, etc. 🤸

10. Reflection & Improvements

- Time.
- Space.
- Testing.
- Optimizations.
- Alternative solutions & their trade-offs.
- Related Ideas & Bigger Picture - Get Geeky!
    - Applications in real life, scaling, etc.

11. End Off - “If I had more time, I’d add more documentation & comments, & make the code more readable”

- Everyone likes working with someone who is:
    - Considerate of others who will be reading their code.
    - Always looking to improve.
    - Humble!

Practice & Leetcodes

- When solving a Leetcode, do all the steps above!
  - *Diminish the gap between real interviews & leetcodes*
- Prioritize Problems You're Scared of
  - Focus on problems that are challenging but within reach
    - Don't focus on problems that are too easy for you 🤭
    - Do the problems that scare you: You're familiar with, but terrible at.
  - Start a list of the type of problems you suck at…
    - i.e. Dynamic Programming, Backtracking, etc.
- Can't solve a problem - Do 2 Things
  - Write down the solution to problem
    - Pseudo Code - Summarize how the solution solves the problem.
    - Go through a few examples.
    - Understand the:
      - ➢ Time complexity.
      - ➢ Space complexity.
  - **Add it to the list of problems you suck at**
    - **Pound that type of problem until you can do a couple without help**.
    - Avoid practicing that exact problem again too soon.

Go Get Em!!!

Contact Me: Joseph Borodach

My Leetcode List | Linkedin | Resume