Hi Pranam,

Thank you for applying to us.

As promised on call, I wanted to share a quick preview of the interview process and a bunch of resources that might be handy! This might be a long email, but these links and resources might be really helpful.

Overall process:
Recruiter Prescreen → Phone Interview *(1~2 sessions)* → Onsite Interview *(4 sessions)* → Hiring Committee Review → Offer Review → Offer Delivery *(Woohoo!)*

Understand Google Interview Process

- Ask a Google Engineer — What is the Interview Process at Google? (2 min)
- How to prepare for a Google Engineering Interview (7 min)
- Candidate Coaching Session for Technical Interviewing (45 min)
- Example of a Coding Interview (24 min)

Step 1. Refresh yourself with CS fundamentals

- We do expect you to know a lot about algorithms and data structures and especially be able to implement them into your solutions - there is a great bigocheatsheet that may also help you!
- Steve Yegge's Blog
- Google Style Guides (C++, Python, Java; Android, Javascript)
- Geek for Geeks - Study algorithms and data structure
- System Design resources:
  - System design interview tip!
  - Hired in Tech

Step 2. Coding Practice + CS review

When you practice, do not use an IDE. You need to be able to write legible, compilable code without help with regards to layout, or spelling of standard library class/method names. I suggest solving similar style algorithmic/ DS problems on a google document or on paper to simulate a real interview. Several sites that provide similar problems to those typically asked in the interview are:

- Leetcode
- Code Jam Kickstart - great practice to get a glimpse of Google coding and algorithm questions. Code Jam hosts online Kickstart rounds that give participants the opportunity to test and grow their coding abilities! Participate in one—or join them all! You can also learn how to solve those problems by reading our analysis
- Codeforces
- HackerRank
- Topcoder
- ACM-ICPC
- Code Jam

Additional Notes/ Optional Studies:

- **Study Materials:**
  - **Courses: can choose one of below**
    - Coursera - Algorithms, Part 1
    - Coursera - Algorithms, Part 2
    - Udacity - Intro to Algorithms

    - MIT Open courseware - Introduction to Algorithms (not efficient)
  - **Youtube Channels:**
    - Tushar Roy - Coding made Simple
    - Interview Prep
    - Google Recruiters Share Technical Interview Tips (30 min)
    - Google Technical Interview Workshop - Singapore (2 hours):
  - **Books:**
    - Beautiful code
    - Elements of Programing Interview
    - Coding Interview University
    - Cracking the Coding Interview

- **Interview behaviors to note**
  - Talk through your thought process about the questions you are asked. In all of Google's interviews, our engineers are evaluating not only your technical abilities but also how you approach problems and how you try to solve them.
  - Ask clarifying questions if you do not understand the problem or need more information. Many of the questions asked in Google interviews are deliberately underspecified because our engineers are looking to see how you engage the problem. In particular, they are looking to see which areas leap to your mind as the most important piece of the technological puzzle you've presented.
  - Think about ways to improve the solution you'll present. In many cases, the first answer that springs to mind isn't the most elegant solution and may need some refining. It's definitely worthwhile to talk about your initial thoughts to a question, but jumping immediately into presenting a brute force solution will be received less well than taking time to compose a more efficient solution.
  - Access to a computer at the time of interview is required as you will be requested to write and share code. Interviewers use Google Docs to facilitate coding in real time.

Couple of pointers to bear in mind during the interviews;

- **Coding rounds:** Emphasis on Syntax, Naming, Organisation and testing. Please take care to use the best features of the language. Ensure a high level of code readability (use good variable names etc). Be particular about boundary conditions and incorporate the same in your answer. Do test your code independent of the interviewer hinting to do so.

- **Data Structures:** Emphasis on right tools for the job, varieties of DS (Lists, Hashtables, Stacks, Queues, Graphs etc). Don't force a question into a structure that doesn't fit.
- **Algo**: Binary search, sorting, Recursion, Dynamic Programming, Time & Space complexity. Try to start simple and improve.
- A simple elegant solution is preferred. You are expected to hit the optimal solution, and be able to discuss alternate methods and their tradeoffs. Time complexity analysis is expected.

- **Efficacy:** Fast solutions with minimum guidance is expected.

- **Remember to think out loud** - ask for clarifications - check corner cases - pay attention to the interviewer's hints/ clues

As discussed, **leetcode.com**, **spoj.com** and interview bit should give a lot of exposure to a wide array of problems. **Codeforces.com (type B and C)** discusses competitive programming problems. While the way they are positioned is different from interview style, the concepts and level of difficulty is similar.

Interview Prep Videos: Interview experience   coding interview   coding interview prep

Kindly go through the reference docs / videos.

Should you have any questions, give a shout!
--