Google Interview Preparation Guide

Start with Interview [tips from Google software engineers](https://www.youtube.com/watch?v=XOtrOSatBoY&t=1s)

Google interviews focus very heavily on algorithms and data structures. You'll be expected to know and apply: lists, maps, stacks, priority queues, binary trees, graphs, bags, and sets. You'll need to talk about how they're implemented and why you'd choose one implementation or data structure instead of another. For algorithms you'll want to know greedy algorithms, divide and conquer, dynamic programming, recursion, and brute force search. You'll definitely want to be conversant with big­O notation, time and space complexity, and real world performance of all of this. Most importantly you'll need to be able to pick the right data structure and algorithm for a specific problem.

Suggestions on reading material include:

Algorithm / Data Structure Theory books

Google Style Guides (C++, [Python](https://google.github.io/styleguide/pyguide.html), [Java](https://google.github.io/styleguide/javaguide.html); [Android](https://source.android.com/devices/architecture/hidl/code-style), [Javascript](https://google.github.io/styleguide/javascriptguide.xml))

[Coursera - Algorithms, Part](https://www.coursera.org/learn/algorithms-part1) 1

[Coursera - Algorithms, Part 2](https://www.coursera.org/learn/algorithms-part2)

[Udacity - Intro to Algorithms](https://www.udacity.com/course/intro-to-algorithms--cs215)

[MIT Open courseware - Introduction to Algorithms](https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-spring-2008/)

- Moving onto the practice after theory:

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:

HackerRank

Topcoder

Codeforces

Leetcode

InterviewBit

Kattis

Problems from [Cracking the Coding Interview Book](https://books.google.co.uk/books/about/Cracking_the_Coding_Interview.html?id=anhAXwAACAAJ&hl=en)

This will help get you into the problem-solving mindset.

Here are some extra videos which are definitely worth watching before your interview:

[What's it like to work at Google?](https://youtu.be/n_Cn8eFo7u8)

[Prepare for your Google Interview: Coding](https://youtu.be/6ZZX9iIgFoo)

[Prepare for your Google Interview: System Design](https://youtu.be/Gg318hR5JY0)

[Prepping for Your Google Interview: Leadership](https://youtu.be/2Cr3-et4xkI)

--------------------------------

Here are some more hints for success

1) Interviewers will be looking at the approach to questions as much as the answer:

Does the candidate listen carefully and comprehend the question?

Are the correct questions asked before proceeding?

Are things assumed without first checking?

Are hints heard and heeded?

Is the candidate slow to comprehend / solve problems?

Does the candidate enjoy finding multiple solutions before choosing the best one?

Are new ideas and methods of tackling a problem sought?

Is the candidate inventive and flexible in their solutions and open to new ideas?

Can questioning move up to more complex problem solving?

2) Interview Questions: Interview topics may cover anything on your CV (especially if you have stated that you are an expert!), whiteboard coding questions, building and developing complex algorithms and analyzing their performance characteristics, logic problems, systems design and core computer science principles.

3) How to succeed: At Google, we believe in collaboration and sharing ideas. Most importantly, you'll need more information from the interviewer to analyze & answer the question to its full extent. When asked to provide a solution, first define and frame the problem as you see it. If you don't understand - ask for help or clarification. If you need to assume something - verbally check its a correct Assumption! Describe how you want to tackle solving each part of the question. Always let your interviewer know what you are thinking as he/she will be as interested in your process of thought as your solution.Also, if you're stuck, they may provide hints if they know what you're doing. Finally, listen - don't miss a hint if your interviewer is trying to assist you!

4) What is Google looking for?: "We are not simply looking for engineers to solve the problems they already know the answers to; we are interested in engineers who can work out the answers to questions they had not come across before."

I hope you find this long email helpful. And please feel free to let me know if there is anything I can do to help.