How to prepare for coding interviews

Last updated: 2019-06-07
Link to this document: http://bit.ly/prepareforcodinginterview
Author: Tilek Mamutov (CEO of Outtalent; previously a software engineer at Google and a project lead at Google X)
For feedback and questions please email: tilek@outtalent.com

1. If you are applying for a backend software engineer to big tech companies choose one language you want to be comfortable during interviews with among the ones listed here, gain experience using it and learn more about it:
   a. Python
   b. C++
   c. Java

2. If you need a course on data structures and algorithms we recommend taking Google's Intro to Data Structures and Algorithms course on Udacity. It is focused on helping you get through a Google interview.

3. Get the latest edition of Cracking the Coding Interview by Gayle Laakmann Mcdowell

The book may look big, but it is written in a friendly way and most of the book is just code with solutions to problems. Here is a secret - you can skip them and just read the theory part. For problem-solving see recommendations in other sections.

4. Use LeetCode to practice coding problems.

   a. We recommend to choose technical topics in approximately this order:
      i. Basic
         1. Array
         2. String
         3. Hash Table
         4. Set
         5. Linked List
         6. Stack
         7. Queue

      ii. Intermediate
         1. Binary Search
         2. Recursion / backtracking
         3. Dynamic Programming

4. Tree
5. Graph
6. Depth-First Search (very popular with interviewers!)
7. Breadth-First Search (very popular with interviewers!)
8. Binary Search Tree
9. Heap (very popular with interviewers!)
10. Sort
11. Bit Manipulation
12. Math

b. Aim to 150-200 easy problems in total for all topics recommended above before moving on to medium ones.

c. Start with 5-10 problems a week and build a weekly or even daily habit before moving to a higher number of problems.

d. Please note some problems marked as “easy” are actually quite hard and some of them may take you hours if you are a beginner and that’s ok. Lots of successful engineers had the same challenge.

e. Try estimating the time complexity of your every solution.

f. After you submit a successful solution check some examples of the solutions of other people on the Discussions page.

g. We recommend solving the same problem in multiple ways rather than immediately moving on to the next problem.

h. Google will most often ask “medium” and “hard”, many other companies are usually a bit easier.

i. Practice drawing out graphs and trees on a piece of paper, whiteboard, and most important code editor. You will need this to explain your solutions during the interviews.

j. Aim to solve 500 before applying to Google. If you solve 300 you can already contact us at Outtalent tilek@outtalent.com sharing your LeetCode profile link and we can recommend the next steps for you.

5. Positions for newly graduate students most of the time don’t require a System Design interview as a part of the interview process, but more senior positions do. In that case, please use LeetCode System Design questions to practice for that too.

a. Video example of a system design question and solution: System Design: How to design Twitter? Interview question at Facebook, Google, Microsoft https://www.youtube.com/watch?v=KmAyPUv9gOY

b. Prep for the system design interview.

6. Optionally, in addition to LeetCode you can try using other websites:

a. InterviewCake

b. InterviewBit

c. HackerRank

7. Optionally, check out some video resources:
    a. How to: Prepare for a Google Engineering Interview
    b. Google interview example
    c. CS Dojo YouTube channel

8. Read LinkedIn resume improvement and update your resume accordingly.

9. Find the roles you want to target and see what requirements they have.

10. Read through What to do during a coding interview to see the most common problems.

11. See Soft Interview Questions for Engineers and follow recommendations there.

12. Do tens of mock interviews on a peer-to-peer interview website Pramp. Don't forget to keep increasing the level you choose for yourself there. The higher you set your level the stronger interviewers you will get.

13. If you are not a native English-speaker we highly recommend the following:

    a. Install Grammarly to see your mistakes in writing and learn to prevent them.
    b. Take an online English test to identify problematic areas.
    c. Practice your English where possible.
    d. We recommend you to consider taking additional English-language classes.
    e. If you are tired watch entertaining/relaxing videos in English with subtitles. :)

14. If you are a junior software engineer try to figure out your strengths and what you would enjoy working on the most by trying to work on a wide variety of technologies to understand what you enjoy doing the most and what you are good at, for example:
    a. Data science and machine learning
    b. Mobile app development iOS / Android / React.Native / Flutter / etc
    c. Consumer hardware
    d. Robotics
    e. AR
    f. VR
    g. Data visualization
    h. Computational biology
    i. Blockchain

15. If you are a student, considering trying to get a good result at competitive programming.

16. If you are a student, keep applying for internships from your junior year, both for Summer and Fall seasons.

17. Consider buying a small whiteboard for real video interviews and mock interviews. This will set you apart from others in case you will have a video interview rather than a phone interview.

Good luck!