Gmail

Steven Athouel <sathouel@gmail.com>

Interview Preparation Material

1 message

Lital Elkayam Efron <litalee@google.com> 25 décembre 2017 à 11:36
À : Steven Athouel <sathouel@gmail.com>

Hi Steven,

Thanks again for your time today, it was great speaking with you.

As agreed, please find below the preparation material for your upcoming technical interview. Please review everything extensively and train as much as you can!

One of our coordinators will be emailing you shortly (from an @google.com email address) with the date & time of your phone interview. Please simply confirm that the time works for you & that your contact information is correct.

Please also let me know if you know of anyone else who would be interested in a role at Google (colleagues, contributors, peers), as we often find that the best know the best and welcome any recommendations!

****

INTERVIEW SET UP:
This will be a technical interview that will last ~45 minutes. Google takes an academic approach to the interviewing process. This means that we are interested in your thought process & your approach to problem solving, as well as your coding abilities. You may be asked questions that relate to technical knowledge, algorithms, coding, performance, how to test solutions, & perhaps your interest in Google products. During the interview, please feel free to ask the interviewer if you are not clear with any of the questions. Also, feel free to talk through a problem when needed.

EQUIPMENT NEEDED:
You'll need a computer with internet access for the interviews to use Google Docs -- a web-based word processor which lets you share & collaborate your work online. Once you click the link, you'll be able to use the Google Doc along with the interviewer. Please note that you do not need a Gmail account to use these documents - just click on the link provided & you will be able to edit the document.

We also recommend using a headset or hands-free device during the interview, to allow for easier conversation while coding.

GOOGLE TECHNICAL INTERVIEW TIPS:
Please be prepared for the engineers to ask you questions in the following areas:
- Google products (i.e. what you use, your favorite one, etc.)
- Anything on your resume
- Coding ability
- Algorithm Design/Analysis
- System Design

HELPFUL LINKS:
Interviewing at Google
Google Products
Five Essential Phone Screen Questions
TopCoder Tutorials
The Official Google Blog: “Baby Steps to a New Job”
“How to Get Hired”

We also highly recommend that you get comfortable answering problems similar to those available on www.TopCoder.com.

AREAS TO PREPARE, DIRECTLY FROM OUR ENGINEERS:

1) Algorithm Complexity: You need to know Big-O. If you struggle with basic big-O complexity analysis, then you are almost guaranteed not to get hired. For more information on Algorithms you can visit this link.

2) Coding: You will be expected to write some code in at least some of your interviews. You should know at least one programming language really well (preferably be C++, C, Java or Python, though C# is OK too, since it's pretty similar to Java). You will be expected to know a fair amount of detail about your favorite programming language. Strongly recommended for information on Coding: *Programming Interviews Exposed; Secrets to Landing Your Next Job* by John Monagan & Noah Suojanen.

3) Sorting: Know how to sort. Don't do bubble-sort. You should know the details of at least one n*log(n) sorting algorithm, preferably two (e.g., quicksort & merge sort). Merge sort can be highly useful in situations where quicksort is impractical, so take a look at it.

4) Hashtables: Arguably the single most important data structure known to mankind. You absolutely should know how they work. Be able to implement one using only arrays in your favorite language, in about the space of one interview.

5) Trees: Know about trees; basic tree construction, traversal & manipulation algorithms. Familiarize yourself with binary trees, n-ary trees, & trie-trees. Be familiar with at least one type of balanced binary tree, whether it's a red/black tree, a splay tree or an AVL tree, & know how it's implemented. Understand tree traversal algorithms: BFS & DFS, & know the difference between inorder, postorder, & preorder.

6) Graphs: Graphs are really important at Google. There are 3 basic ways to represent a graph in memory (objects & pointers, matrix, & adjacency list); familiarize yourself with each representation & its pros & cons. You should know the basic graph traversal algorithms: breadth-first search & depth-first search. Know their computational complexity, their tradeoffs, & how to implement them in real code. If you get a chance, try to study up on fancier algorithms, such as Dijkstra & A*.

7) Other Data Structures: You should study up on as many other data structures & algorithms as possible. You should especially know about the most famous classes of NP-complete problems, such as traveling salesman & the knapsack problem, & be able to recognize them when an interviewer asks you them in disguise. Find out what NP-complete means.

8) Mathematics: Some interviewers ask basic discrete math questions. This is more prevalent at Google than at other companies because we are surrounded by counting problems, probability problems, & other Discrete Math 101 situations. Spend some time before the interview refreshing your memory on (or teaching yourself) the essentials of combinatorics & probability. You should be familiar with n-choose-k problems & their ilk – the more the better.

9) Operating Systems: Know about processes, threads & concurrency issues. Know about locks & mutexes & semaphores & monitors & how they work. Know about deadlock & livelock & how to avoid them. Know what resources a processes needs, & a thread needs, & how context switching works, & how it's initiated by the operating system & underlying hardware. Know a little about scheduling. The world is rapidly moving towards multi-core, so know the fundamentals of "modern" concurrency constructs.

10) Research Publications: Check out Google's publications here & paper on Google's Hybrid Approach to Research.

Here are a couple of extra videos which are definitely worth watching before your interview:

1) Example of a Coding Interview
2) How to best prepare for a technical interview: Preparation

Good Luck and let me know if you have any questions!

Best,
Lital