



Interviewing at Google? Here's 6 Things You Absolutely Need To Do



Anthony D. Mays | [Follow](#)

Software Engineer at Google



1,056



72



334

Having interviewed, coached, and sat on hiring committees with many candidates during my three year tenure at Google, I've learned a lot about what works and what *doesn't*. I conquered the interview process after failing once before and, even then, I thought it could've gone either way. With tons of stuff to study, it's hard to figure out how to prepare. Even really smart people can fail if they don't have a good plan for what to do when they get into the room with an actual software engineer and a whiteboard.

So, I'm going to share six tips I believe you absolutely need to nail your interview. After you check these out, watch this [sample interview video](#) conducted by Google engineers.

your comprehension. Remember, your interviewer is there to assist you, so repeating the question aloud will only serve to make their job (and yours) easier.

This will also buy you some time to think about what you've been asked and develop good questions or approaches. Plus, hearing yourself restate the problem might help you think through it more clearly.

2. Check assumptions

Many candidates start writing code almost immediately after hearing the question. This is a big mistake. Most coding questions have some level of ambiguity built in. Have at least two or three questions ready so that you can confirm you have all the necessary detail to solve the problem. You can also write out confirmed assumptions on the whiteboard (do it in small print to save space). Questions like "does input fit in memory?" or "can I assume input is always valid?" are good candidates if you can't think of any at first.

3. Use real examples

You've got a whiteboard—use it! Draw out an example array, a binary tree, a linked list, etc. Give it real data and write out the expected output of a working solution. You should practice coming up with good examples as part of your study regime (you do have one, right?). Using a visual example also gives you opportunity to think up more questions and your interviewer a chance to correct your assumptions. Don't forget to keep thinking out loud as you work through it.

If your interviewer provides examples, use those since they probably exist for a reason. This is also how interviewers will point you towards problems with your design or implementation.

4. Brainstorm solutions and their time/space complexity

Stop and think about various approaches. If you've put in time studying algorithms and data structures, this is where it really starts to pay off ([Cracking the Coding Interview](#), anyone?). Think about trade offs using [Big-O analysis](#) and *think out loud*. Don't stop with the first solution that comes to mind. Always ask yourself what's the best you can do. Trust me, we *always* will!

Tips: Don't forget the [space-time tradeoff](#) principle. Wanna go fast? Use more space. Also, hash tables people! And learn how to [think graphs](#)—including trees. If all else fails, [recursion, backtracking, and memoization](#) are useful tools on particularly difficult problems.

5. Write working code (no pseudo-code please!)

You might normally use pseudo-code to design your code, but you don't have time for that in a 45-minute interview. Choose your strongest language and turn your thoughts into working code as quickly as possible. This should be the easiest part of the interview assuming you've put time into practicing without an IDE (seriously, don't use an IDE to practice for the interview unless you're compiling code you've already written out). Make sure to practice using a whiteboard or just use pen and paper.

any logical bugs. Use the whiteboard to keep track of variable state as you iterate. Also, use the example(s) you created earlier and confirm that you get the expected output. Testing is crucial in software engineering, so never make the grave mistake of leaving this step out.

Practice makes perfect

It took nearly an hour and a half to finish a coding question when I first started using these six steps. After a month and a half of daily practice, I could solve most questions in about half an hour. *Nothing* takes the place of putting in the time to prepare for the interview until you've developed the muscle memory to execute these six steps perfectly. Is it worth it? **I'd say so.**

Anthony D. Mays is a Software Engineer at Google and a former foster youth from Compton. Read more about his story and career tips at anthonydmays.com.



Anthony D. Mays

Software Engineer at Google

Follow

72 comments

[Sign in](#) to leave your comment

Ajay Paul

SDE II, McAfee

1y

attending an interview is one thing, how does one get an interview call from google ?

Like Reply | 3 Likes 1 Reply



Pedro Ernesto Alonso

PHD Student at Luleå University of Technology

1y

I can tell you that a recruiter contacted me in LI, he liked what I have been doing, swift code.

Like Reply | 1 Like



Tal Peretz

Data Scientist at Simplex

2y

Or Cohen-Ofir

Like Reply | 1 Like

laconis Software Engineer at Snowflake Computing

"Choose your strongest language..." When I interviewed at Google's Boston office, I began to code a solution to one of the interview questions using JavaScript. The interviewer stopped me and asked if I could use a different language b/c he wasn't familiar with JavaScript. I tried my best with Python, but I wasn't as confident as I would have been using JavaScript. Is this common for Google interviews?


Like Reply | 1 Like 2 Replies

 **Eric Leschinski**
Software Developer. I just received my MS in CS specialization ML from Georgia Tech.
Open to relocation US-Northeast

2y

The interviewer was trying to Throw you off balance. Much like two soldiers in combat trying to harm one other, the google interview at is one where each tries to intellectually subdue the other by preventing your opponent from using their greatest weapon. You need to have excellent skills in resisting people playing the dark triad. The correct answer was to nod sympathetically, like a mother to a loving child and say "Oh, I'm sorry that you don't understand computer-datum-coding, maybe you could reduce the complexity of the problem so you will be capable of following along?".

Like Reply

 **Anthony D. Mays**
Software Engineer at Google

2y

Not that I'm aware, though miscommunication does sometime occur. It's been my experience that Google does its best to pair up the right interviewers for each candidate that is familiar with their programming language or technical expertise.

Like Reply | 1 Like

 **Tim H.**
UX/UI Front End Designer at IMAX. NYr in LA

2y


You don't have time for pseudo code in a 45 minute interview of complexity that you are asked to do on the spot that would be given probably a full sprint to do... (walks out door, knocking over your pencils while making eye contact)

Like Reply | 1 Like 2 Replies


D.  Software Engineer at Google

Mays To be clear, there's nothing wrong with using pseudo code during the interview if you can still crank out good code in good time. My only point is that you can save time by writing good code upfront without pseudo code (which is what I tried to do in my interviews). There would, of course, be a problem with acting rudely to your interviewer.


Like Reply

 **Amanda Wong, CPSR** 2y
Senior Technical Recruiter with CBS Interactive - We are hiring! Please check us out cbsinteractive.com/careers
Wong, CPSR Sorry about that!

Like Reply

 **Sreejith Rajasekharan** 2y
Team Leader at Oracle India Pvt. Ltd
Rajasekharan Its an amazing experience to just go through their interview process....one thing i found is they are not looking for an easy answer or a bookish answer to even a simple questions..they have really experts in panel who would put you in tough spot with simplest questions...no doubt they are leaders in everything...

Like Reply | 3 Likes

 **Erroll -EL- Warner** 2y
Business Retail Sales
EL-Warner Did you had to study for an interview? I thought your qualification and a normal interactive recruiting process will follow. Some CEOs are so confused about their own organization process of recruiting that they are now blaming their Human Resources and recruitment staff. A recent research stated 63% of CEOs have lost confidence in their Human Resources and recruiting personnel. The problem seem to be those CEOs and not their Human Resources staffers. Many of those CEOs sit in their offices disconnected and lonely. Their office loneliness seem to be causing them to do weird things.

Like Reply | 1 Like 1 Reply

 **Anthony D. Mays** 2y


Mays

Google Technical Interview <https://www.linkedin.com/pulse/my-guide-preparing-google-technical-interview-anthony-mays>

Like Reply

Show more comments.


More from Anthony D. Mays 6 articles



Hacking the 'Hood: How Tech Can Solve Its Pipeline Problem

[Hacking the 'Hood: How Tech Can Solve Its...](#)

November 13, 2017



How To Craft A Solid Resume That Grabs The Attention Of Top Tech Companies

[How To Craft A Solid Resume That Grabs The...](#)

July 13, 2017



Google Helped The Father I Never Thought I Would Be

[Google Helped The Father I](#)

May 6, 2017