

Google-Bangalore Interview Experience

- Difficulty Level : [Expert](#)
- Last Updated : 21 Jun, 2021

Disclaimer: I could not mention the exact questions asked to me due to an agreement, however I tried to describe the topics as best as I can.

Phone Test:

On late December I was contacted by a Google Recruiter from Singapore in LinkedIn. She shared her calendar and asked me to mark a 30 minute window for a quick chat over my professional experience and interests. The call started with me describing the same and then pivoted to some rapid fire questions regarding basic concepts like heaps, trees, their algorithmic complexities in various cases, how one outperforms the other and so on. It concluded by her giving me a math question and solving it as quickly as possible, but at every step I was expected to explain my approach. Finally she said I passed the phone test and soon scheduled a 45-min video interview on Google Hangouts.

Google Hangouts Test:

The interviewer was from Google US. It started with a friendly introduction and then the real problem followed. It was a DP problem on 1-D array, not very hard once you see the solution. But at the heat of the moment (with less time) I explained a seemingly obvious greedy approach, which he refuted soon with a test case. Fortunately I got the DP algorithm in time and could code it (with few implementation flaws that I later realized). He asked about the space and time complexity, and then asked me to reduce the space complexity. I did it. It ended with some discussion over Project LCZero, and then about AlphaZero (by Google Deepmind) that recently won against one of the strongest computer chess engine Stockfish.

A few days later got to know from the recruiter that I have been selected for the onsite interviews at Google Bangalore.

Onsite:

Round 1:

It was the worst round for me among all of them. The question was basically DP on trees, and although generally such questions belong to the hard category but coincidentally I solved a similar problem few years back during college (I even mentioned about this to the recruiter after he asked if I had seen this before). So I solved the basic version very quickly. But then he added a simple constraint, and it suddenly was a black out for me, as I scratched my head for long but could not find any possible way to include this constraint in my existing solution. I proposed to him that this edge case can be handled explicitly, but that really destroys the beauty of the solution. So he insisted me to find a concrete way of solving all cases, and I took up the challenge, and soon enough ran out of time and also could not find the concrete solution. Only 4-5 minutes were remaining for the interview to end, I hurried to code my existing solution (and handled the corner case explicitly), but even while I was coding I understood it went very wrong. Discussed the complexity of the solution and it ended.

Round 2(postponed):

Interviewer was at a tour, so this round was postponed. A sigh of relief from me, after having such a disastrous beginning I needed a break.

Round 3:

It was a behavioral round a.k.a the Googleness round. It had nothing to do about tech but it is about how do we react to different situations in the professional world. I actually enjoyed this round very much, because most of the questions were such that I had already faced such a situation in my previous company, so I responded quickly with what exactly I did in that situation, and in some cases what things I could have done better that I realized lately. I completely forgot about the mishap in the first round after this round. I was quite happy with the way I dealt with the questions and also, the lunch interview was to follow

Round 4:

It is not really an interview. A googler came and took me with him to have lunch, and wow, so many dishes, chicken items (my favourite) all around. I ate quite a lot, and in the process had a good chat with him regarding what he worked on, where is he from, and some basic stuff about the culture at Google. We all know the culture is definitely good, but hearing some details from an actual googler in person was handy.

Round 5:

It began right after lunch. This round I believe went the best for me. The question started with multiple string matching (with some secret constraints, ofcourse), and then a lot of discussions happened regarding topics like KMP algorithm, Z algorithm, Rabin Karp hashing mechanism, and finally concluded to a modified Trie operation. Regarding coding, I did not have to build the trie myself, I was asked to assume I have the trie from a library, but need to write custom operations on top of the functions available from the Trie. Then we discussed about the space and time complexities and how can we further optimize them.

Round 6:

It involved building a game. It felt very simple, I was quite surprised initially. But as I kept on asking questions and the interviewer stopped me at several points and revealed more constraints about the game which kept on increasing the complexity. I proposed a solution with a BST first, later changed it to an algorithm that does not require the ordering and hence could be solved just by hash maps. The final code logic was quite simple, and so after my coding finished we had quite some time to spend on other topics. He then gave me a design problem which basically boils down to computing shortest path between well-defined vertices, gave some approaches, hopefully he was satisfied.

A few days later my recruiter informed me that I have done good at the previous on-site rounds and she would forward the packet to the Hiring Committee regardless of how the upcoming round would be. I was overjoyed and started preparing for the upcoming round anyway.

Round 2(postponed earlier):

I actually wanted a face-to-face interview, comeon a F2F conversation can beat a video chat any day. However due to some reasons it happened through video call. I got the question completely wrong and kept on suggesting approaches for 10 minutes until I realized (my recruiter helped me to understand it quickly, or else the whole interview would have been over by a miscommunication), then quickly I jumped back to the original problem and gave a quadratic solution. He gave me some time to think if this can be optimized, and suddenly I remembered Kadane's algorithm,

using which I could overcome the bottleneck (which was a small part) of the whole solution and it became a linear solution. Very little time was left by then so I hurried to code, and like it happens, missed some implementation details in the coding part.

A few days later I was informed by my recruiter that I have an upcoming meeting with a team manager from Google Ads. I was super excited as things really seemed to go well. I had a small 15 min meeting over Google Hangouts with him, described him my interests and experience at the previous company, he described me about the role, the tech stack and it seemed interesting. A match happened and I received a statement of support from him soon for being at that team.

All's well that ends well. Well it did not in my case, I could not make the cut in the final step the Hiring Committee.

I asked for the reviews and my recruiter explained to me that at few instances my coding was not upto the mark and I could have simplified the logic in some situations.

Result Rejected.

All the coding rounds were done either at Google Docs (during hangouts) or at their personal interviewing application (at onsite, quite similar to docs) .

Overall, although the whole process took around 3 months, it was quite interesting. Preparing over the stuff that you once did at college, and then giving those interviews, meeting the Googlers along the way, getting to the onsite Google India office for the interviews (twice!, in case of me) had been a wonderful journey.

My advice to fellow Google enthusiasts is that, balance the time well between the thinking part and the coding part during interviews, your coding really makes a very important impact later, you might have explained the algorithm nicely during interview but later (say at HC stage) all that is left is your code. And that 'code should talk well'.

My Personal Notes

Add your personal notes here

Save

,