b'

Google Bangalore Interview Experience

- Difficulty Level :\nMedium
- Last Updated :\n18 Feb, 2020

Initial Phone Screen:

The recruiter asked me about my background and discussed the interview procedure. She also asked me some fundamental one-word answer DS/Algo questions like time complexity of quick-sort.

Technical Phone Interview 1:

It didn\xe2\x80\x99t go very well as I had trouble explaining my approach to the interviewer. It was a computational geometry question and I had never done questions of this type before. I had a hard time coming up with the naive solution because all the time I kept assuming the solution I was thinking would not work (again lack of confidence). My nervousness was getting bigger as the time kept passing. About 20 minutes later he asked me to code whatever I had yet told him. After I coded, he pointed out cases in which my algo would fail. Since I coded in a very structured and modular way, so I was able to quickly fix it but still, it was not completely correct.

I got my recruiter\xe2\x80\x99s call after a week saying it was a mixed response \xe2\x80\x93 she told me that I didn\xe2\x80\x99t think out loud but did very well on all the other parts like data structure and code structuring and they\xe2\x80\x99d like to conduct another phone interview.
Well after the call, I just tried compiling the code and discovered to my astonishment that it ran successfully and the answer was correct. I felt so stupid for not believing in myself all the time, that despite having the ability and enough practice, I was under-confident. Anyway, I was glad that I still had another chance.

Technical Phone Interview 2:

It was an expression evaluation sort of question with some conditions. I did it using stack and finished coding it in 20 minutes. I checked my code twice but missed a syntactical error and he had to point it out. He then asked me a follow-up and I did it in a couple of minutes as I made a function for core logic, so I had to adjust only that. It went superb and in the end, we were left with enough time for chit chat.

As my onsites approached, my LC problem count got\xc2\xa0**177 \xe2\x80\x93 59 easy, 102 medium, 16 hard**
(I started doing hard problems alongside medium ones after I got the onsite invitation as I had to level up my prep. Hard problems are exhausting and are usually a combination of 2-3 medium ones. So I focussed more on medium ones).

Onsite Interview:

Date: January, 2020

As I reached the office, I was very nervous. But I\xe2\x80\x99m happy that my recruiter calmed me and boosted me for the interview.?

Round 1:

The interviewer asked me a warm-up question first and I instantly told her the solution and time/space complexities. That made me feel extremely relaxed. The next question she asked was a graph problem. It could be solved by both dfs/bfs, I discussed why bfs would be a little better over dfs. I wrote down the algorithmic steps before coding and then coding in itself took very little time. (I\xe2\x80\x99ve written this approach in detail in the interview tips below)

Round 2:

The second interviewer asked me to segregate numbers within a range(0 to N) that meet some criteria. I told him the naive iterative approach to check all numbers and then discussed ways to optimize it. I solved it by backtracking to generate only the valid numbers.

Round 3 (Googlyness / Behavioural):

This round felt important in ways that it shows your determination and optimism in different situations. I liked that the questions were scenario-based and open-ended.

Round 4:

It didn\xe2\x80\x99t go as well as my other interview rounds. It was a tree question for which I came up with a recursive solution but wasn\xe2\x80\x99t sure of how to properly make use of the returning values. Time was running out, so he asked me to code whatever I had yet told him. I could not plan my code prior, so had little difficulty implementing it and was re-rewriting my code. By the use of some hints, I could barely finish my code on time. He wasn\xe2\x80\x99t able to ask any follow-ups.

Round 5:

It was an array question for which I came up with a dfs solution but got stuck. The interviewer nudged me towards bfs. I coded the bfs solution and in the follow up he asked me how I would test it.

Result:\xc2\xa0My onsites went fantastic, so they fast-forwarded my application. The HC approval came next week.

Conclusion:

I started my journey by canceling my interview due to my fear of rejection, but now I realise it was me rejecting myself that day! I highly advise everyone \xe2\x80\x93\xc2\xa0*Have faith in yourself and put yourself out there, you may be disappointed if you fail, but you are doomed if you don\xe2\x80\x99t try!*

Learning is an uphill journey and it requires a lot of consistent practice. I\xe2\x80\x99m on a full-time job and I saved my break-times to do Geeksforgeeks and\xc2\xa0 leetcode problems. Some days were hectic but I still at least did one problem. I never studied much on my weekends (I wanted to but was too lazy to get out of bed ever). So just stay consistent and soon you\xe2\x80\x99ll start enjoying the process, in fact, you will start finding excuses to do leetcode.

Resources:

- **Clear data structures \xe2\x80\x93**\xc2\xa0I used Geeksforgeeks and HackerEarth tutorials to understand the implementation of various data structures. It was quite explanatory yet brief.
- **Clear algos \xe2\x80\x93**\xc2\xa0Spare some time to read CLRS if you wish to be a real programmer. Since it\xe2\x80\x99s very vast, I used it when I needed in-depth knowledge of sorting and dynamic programming.
- **Cracking The Coding Interview (by Gayle Laakmann McDowell) \xe2\x80\x93**\xc2\xa0Very very helpful!! Once you have your data structures and algos clear, use this book to learn how to apply the knowledge practically while coding. Start with the first question and keep moving one at a time because difficulty level increases with each question (I felt so).

- **Dynamic Programming \xe2\x80\x93**\xc2\xa0For top-down solutions, dfs/backtracking with memorization works well.
- For bottom-up:\xc2\xa0https://www.byte-by-byte.com/dpbook/\xc2\xa0(\xe2\x80\xa6 though it\xe2\x80\x99s very tough to implement a bottom-up DP in interviews, and no one expects you to do it also)
- **Leetcode premium \xe2\x80\x93**\xc2\xa0It is absolutely worth it.
- **Mocks \xe2\x80\x93**\xc2\xa0I found both company-wise mocks and random mocks to be helpful in measuring prep. Mocks are good because the problems don\xe2\x80\x99t have the difficulty level tag, so sometimes I could solve hard problems by breaking it down which I would have otherwise ignored knowing it is hard.

Interview Tips:

1. Structure your code better \xe2\x80\x93 make functions and reuse.
2. Always use a whiteboard if given a choice between whiteboard and pen-paper!! It becomes very easy to explain your solution to the interviewer and it also opens your mind to think in many many directions (due to its vastness).
3. If you have difficulty coming up with the naive solution, take a general case example and solve for it. Never take special cases for building solutions, rather keep them as condition covering for later. At this stage, don\xe2\x80\x99t think about the time and space complexities, just solve it because some candidates fail to do even that, so something is better than nothing.
4. Discuss time and space trade-offs after each solution you tell, even if the interviewer doesn\xe2\x80\x99t ask you \xe2\x80\x93 it is expected to tell yourself.
5. Once the solution is discussed and finalised, coding should be as quick as possible. Since it\xe2\x80\x99s not that simple, we can break down the coding steps. (I write algorithmic steps in plain English each time before attempting to code) \xe2\x80\x93 like, make a hashmap of the given data \xe2\x80\x93 use queue data structure \xe2\x80\x93 what arguments to pass to a function \xe2\x80\x93 what to return \xe2\x80\x93 how to use the return value. This helps in two ways:
   - You can easily cover the special cases and edge cases because, at this time, you\xe2\x80\x99re neither thinking about the solution nor coding it, you are just planning how to code it.
   - If something is not right, the interviewer can immediately point it out. So it gets fixed before you start coding the wrong solution.

Lastly, big thanks to the GeeksforGeeks for all the inspiring posts and amazing solutions!! All the best to everyone trying out there \xf0\x9f\x99\x82

My Personal Notes\n*arrow_drop_up*

Add your personal notes her

Save

'