Himanshu (MSC 2020-2022)

Round 1:

The interviewer started by discussing about the college, how is M.Sc. going in online mode and some other ice-breaking questions. The interviewer shared the google docs link and directly jumped into DSA questions.

First Question: (Easy) Link

I started with the naive recursive approach then he told me to write code for that. After that he told me to optimize that so I used the dynamic programming approach, we discussed the time and space complexity and he was satisfied with my solution.

Second Question: (Easy) Link

I told him the naive approach to sort normally that would take O(n log(n)) time then he asked me to optimize it, for that I used two pointer approach. He asked me to share my screen and write the code in whatever IDE/text-editor which I was comfortable with. I wrote the code and he asked me to run the code for different test cases just to ensure there are no edge cases that weren't handled.
Then he asked me few core subject related question:

- What is abstraction in OOPs? Write pseudo-code for that.
- What is encapsulation and data hiding? Write pseudo-code for that.
- What is deadlock? Write pseudo-code for that. Follow up question - How can we resolve deadlock?

Third Question: (Medium/Hard) Link

I told him the naive approach and thought of some patterns, he gave me a few hints but still there were some overflow conditions, since the time was already exceeded I didn't write code for this question.
He gave me some feedback (mostly positive) and asked if I had any questions for him, I asked about what team and projects he works on.

Second Round

The second round started with a brief introduction of myself and he asked me to explain any one of my projects.
The interviewer then jumped onto DSA questions:

First Question: (Easy/Medium) Link

I told him the approach and he asked me to code it in any IDE/text-editor of my choice. He was satisfied with my solution and we moved on the next question.

Second Question (Easy/Medium) Link

I told him the naive approach first that was to linearly traverse the array, then he asked me to optimize it. I optimized it by doing jumps of (2 * idx) rather than linearly traversing, after finding a one I have the left and right boundary so I did a simple binary search on that to find the first occurrence of one in that range.
The interviewer was happy with this solution and didn't asked me to code it.

Third Question (Hard)

This question was related to DBMS, the problem statement was something like this:
"Given two different services running in parallel, we have a token that should be given to the first user that accesses it from either of the service."
I discussed something related to mutexes but he said that approach won't work on multiple services, I told him that we can have a boolean value in a database to see if the token is already acquired or not. He said there will be a race condition if both of the service access the boolean value at the same time. I thought a lot about this problem but I couldn't come up with a solution. He told me that we can use the ACID properties of databases in order to solve this problem. Since select doesn't follow the ACID properties we can use update for making sure that we acquire a lock first.

Fourth Question (Medium/Hard)

It was a pattern matching question. The problem statement was:
"Given a book, find a sentence in the book."

I started with the brute-force approach, then he told me to optimize it, I told him about the KMP algorithm but he wasn't satisfied with my solution. He said that the time was almost over so he told me that I was expecting a `Trie` based solution.
He didn't gave me any chance to ask a question he simply said goodbye and left the call.

Third Round

So the interviewer gave me his introduction and asked me to introduce myself, what my favorite subject was and he asked me about my open-source contributions. He went in-depth into that open-source project and asked me how it was working at a high level (make sure you know about any the project if you are mentioning any open-source projects).
Then he started with my favorite subject which was Operating Systems, he said that let's say we don't have any operating system and we are trying to implement one from scratch, what components do we start from. I told him about starting with the boot-loader and he had a lot of follow up questions regarding that, there are some that I remember:

- Where is the boot-loader program stored?
- Where is the BIOS memory?
- What does the ROM stores?
- What happens if the boot-loader of your system if removed?

After that he asked me to think of the next important component, I told him file system. He asked me some questions regarding that:

- What type of file system do we want to have hierarchical or flat?
- What is an inode?
- How do you make searching of a file quick?

From that he jumped onto some RDBMS concepts like:

- What is normalization?
- Tell me about 2NF, 3NF and 4NF.

- Is there a situation when we want to do de-normalization of a database?

Then he asked me to optimize the search operation in a file, I told him about tries but he asked me for a better solution which I couldn't provide.
The time was up by then, he asked me if I had any questions for him, I asked him about his current work and how do they maintain code quality for large code-bases.
I asked for a feedback at the end and he gave me a very positive feedback.
Overall I liked this round the most, it was like a rapid fire round he moved from one question to another really quickly.

HR Round

She introduced herself and asked me to introduce myself. Then she asked a few questions:

- Tell me about some of your projects.
- Tell me about your family.
- Why InfoEge?

Then she asked me if I had any questions for her, I asked about her work in the company.

Some Tips

- Try to think out loud and clear the question in the beginning.
- If you are not able to get to a solution try to explain the interviewer what is blocking you.
- If you have mentioned something on your resume make sure that you know the ins and outs of that.
- Ask for a feedback at the end of the interview.