Jatin Rohilla – MCA Batch 2017-20
Amazon Interview Experience
Date – 19 July 2019

I've tried to list down all the questions I faced. I've also tried to find them online while writing this to get you a link for more clear description.

Online test

Around 65-70 people sat for the test. 15 people were shortlisted.

It consisted of

- 28 MCQs based on core CS – trees, bfs, dfs, c++ outputs, stacks, queues.
  If you've prepared for placements, you'll get through them pretty easily.
- 2 Coding questions – everyone had different and random questions. Most questions were custom logic based (easy level) including some standard questions – LCS, LIS, topological sort.

Advice –

- the platform is shit. Get your basics right. don't depend on a lot of libraries (just to be safe)
- Code quickly. Coding 2 questions means you're in for the next round.
- It is advisable to have ample practice for these type of questions to build up speed and efficiency. These questions have hidden test cases. Make sure you're well versed with them.

Round 1

The interviewer was very friendly. He first introduced himself and then it started with the standard – tell me about yourself / introduce yourself type of question. This introduction is usually just an ice breaker, keep it short and crisp.

Then he proceeded and asked two coding questions –

- Given a sorted linked lists, delete all occurrences of duplicate elements. You have to write a function and return the new head.
  eg – 1->2->2->3 will result in 1->3
  Eg – 1,1,1,1,2,2,3,3,4,4 will result in null list
  https://leetcode.com/problems/remove-duplicates-from-sorted-list/

- Given courses with dependent courses, find the order in which they should be executed, if they can be executed, else return empty list. You should clarify in which format is the input, etc, etc. This is a standard topological sort problem
  https://leetcode.com/problems/course-schedule/

I first explained overall approach then coded them on paper.

Advice –

- Always clarify the question. You don't want to be solving a wrong question.
- Do not start coding unless the interviewer is satisfied with your approach.

Round 2

This round had only 1 question. The interviewer introduced himself. And he advised me to clearly understand the problem before proceeding.

I'll try to explain the problem.

There's a complete binary tree. Each nodes has two children. ALL leaves are on same level. The nodes are labelled as 1,2,3,…. And so on natural numbers, in level order traversal format. Each alternate tree level is black. Like- level 1 with root is black, level 2 is red, 3 Is black, 4 is red and so on.

The twist is, the tree is infinite, it can't be stored in memory. It's a virtual tree. There exists no real pointers. So you don't really have a node structure, or any pointers. Everything is imaginary.

Now the question is, given label names of two nodes (a and b), find the no of black nodes that exist in the path from a to b. (The path may or may not pass from root.)

Interviewer in the starting itself told that no need to panic. He was very friendly. I started small, slowly describing all the observation. And we slowly built the solution together. Took us around 30-40 minutes maybe. Then I coded it on paper.

Function to be made was - `int countBlackNodess( int a, int b )`

After this we had some spare time, so we talked a bit.
I asked him a few questions about the work at amazon, etc, etc.

Round 3

The interviewer was friendly. The questions were as follows.

- Implement BST iterator.
  https://leetcode.com/problems/binary-search-tree-iterator/
- Find zeroes to be flipped so that number of consecutive 1's is maximized
  https://www.geeksforgeeks.org/find-zeroes-to-be-flipped-so-that-number-of-consecutive-1s-is-maximized/

In both these questions, I first described my approach. Then coded it on paper, run several test cases on the code. Make sure you've tested your code before submitting it to the interviewer.

Then he asked one more question. It had some preceding questions, but the crux was this question.

- 0-1 BFS variant
  https://www.geeksforgeeks.org/shortest-path-weighted-graph-weight-edge-1-2/

We discussed this third one, then he simply asked me to code a normal BFS for a graph.

- BFS of a graph
  https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/

Then he asked some general tech questions -

- What is normalization? Can you tell me what are some normal forms.
- What database have you worked on.
- Do you know diff between sql and nosql. When should we use nosql

- What type of db would you use for a ecom platform
- Example of nosql databases? Example of nosql databases of amazon

Advices –

- Always think about your coding style during interview prep.
- A lot of questions especially graph ones look like they have long code, but if your coding style is crisp, clean and you use libraries as required, your code turns out to be very clean.
- Always strive for clean code. That's what impresses the interviewer apart from your solution.

Round 4 - Bar Raiser

This seemed like the toughest round. The interviewer's style was very different from others. He was a very experienced interviewer. He preferred to keep quiet. He gave me a lot of time to think on every question he put, very explicitly. Then once I was satisfied we discussed those problems.

- Tell me your best project from resume
- What did you do in this
- How is it different from any other similar project
- How did you assure code quality in your project
- How do you define code quality
- How did you test your code
- Did you write test cases for your code
- Write down test case for this code (he gave a small code – like fib, fact)

He took each point very seriously and raised a lot of genuine arguments. He's confident and experienced (6-7 years – manages 70-80 teams)

- Then he asked me if I liked music.
- Which app do I use for music.
- Lets design a db for youtube (saavn, soundcloud, amazon prime) kinda app.

So I wrote a basic schema. He then picked up on the details. And then we moved to a specific table.

- Why this table like this
- Do you think youtube uses it like this
- Do you think its scalable
- What can you do to make it scalable
- What should you do to make it scalable
- Take your time, think about more use cases, develop a generic db structure which can accommodate some of future changes.
- A little discussion about ideologies - "plan a lot of use cases and then make db schema" or "deliver and iterate".

Then he asked me a coding question.
https://www.geeksforgeeks.org/find-first-non-repeating-character-stream-characters/

I tried some data structures but could not confidently get it right, I think.
We just discussed the approach. No code.

Then some general questions -

- Why do you want to join Amazon? Why not other companies?
- Are you open to relocation?
- Do you have any question for me? Then I asked him a question about life at amazon.

This was a really heavy interview, some of it because of the questions, some of it because of his valid arguments, and more of it because of confidence on his approach.

Result

Selected for Full Time position.

Total selections – 4
- 2 internships and conditional job offer
- 2 internships and full time employee offer

Advice

- Be confident. Be Positive.
- Speak well. Speak clearly.
- Speak a lot, unless he asks you to keep quiet and think and answer
- Write clean code, use rough sheet to structure your code if necessary.
- Test your code, talk to the interviewer while writing code if possible.
- Keep attacking the problem from new angles. Never say, I give up.
- It is advisable to not tell him if you've seen the problem before. If he asks you explicitly, tell the truth. Use your own judgement, you should be natural.
- Be well prepared. Get familiar with standard questions. Under-prep will do you no good.
- Paper coding means you should be able to write code without an IDE. I always coded on my laptop during preparation. And I was super comfortable on paper the few times I tried.
  So it is really okay if you usually practice on laptop. Although, get familiar with paper as well. Identify where you code best.
- I recommend interview bit for practice and leetcode for interview prep.
- I recommend solving questions and submitting them on leetcode/ interviewbit. Then only you'll learn to avoid sloppy mistakes. Within weeks you'll be great at it.
- Never assume you can code a solution perfectly just because you know the approach. Always code it out whenever possible.
- It is usually advisable to start from brute force approach and then move to efficient, however, I did not spend a lot of time on brute force in any of my rounds. I simply moved around the problem for 3-4 minutes, stated the brute force solution, and then approached towards the optimal answer. It saves time. Although, it varies from interviewer to interviewer, and candidate to candidate. Do whatever suits your style.

All the best, and good luck.