

Amazon Interview Experience | On-Campus for SDE-1

- Difficulty Level : [Hard](#)
- Last Updated : 17 Aug, 2020

Recently, Amazon came to our campus for the role of Software Development Engineer. The hiring process consisted of 1 coding test and 4 interview rounds.

Round 1 (Coding Round): The coding round consisted of a code-debugging section, Aptitude questions, and 2 coding questions. The code-debugging section was simple where we had to find the fault in the code due to which wrong output was coming. Aptitude questions were general logical-reasoning questions. The coding questions were also easy. I only remember one question.

- [Find an element in a sorted 2-D matrix.](#)

After this round, 57 people were shortlisted for interviews. Due to the current situation of coronavirus, all the interviews were conducted online on the Amazon Chime platform.

Round 2 (Technical): I was asked three coding questions. I was asked to write the code on the LiveCode site that Amazon uses and discuss the time and space complexity.

- [Find the number of disconnected components in a graph.](#)

I first gave a DFS approach by using a visited array but the interviewer asked me to use $O(1)$ space. Then, I suggested changing the node value when we visit the node.

- [Populate the next pointer in a binary tree.](#)

I first gave a level order traversal method which required a queue. She then asked me to do it in $O(1)$ space. So, I gave a recursive solution in which we traverse each level before moving forward.

- Given a BST, find the number of arrays that will result in the same BST.

For eg Given a BST like this: One array is 2 3 4 5 7 6 10. Another array is 2 5 7 6 10 3 4. But this is not valid array 2 5 10 7 6 3 4. This is because if 10 comes before 7, then 10 can become the root and 7 become the left child which will result in different BST.

```

5 \r\n    /    \\r\n  3      7\r\n    /  \\    /  \\ \r\n  4 6  10

```

Round 3 (Technical): The interviewer first asked me some questions on the project that I had done. Then, he asked me two coding questions.

- Given an array where each element denotes the cost, we will take two elements, add them, then insert it back in the array. The cost of the operation is the sum of the cost of both elements. We have to minimize the cost.
Eg: given array as 2, 3, 4, 10. The minimum sum is $5 (2+3) + 9 (5+4) + 19(9+10) = 33$.

The solution that I gave was to insert all elements in a min-heap (implemented as a priority queue) and then pop two elements from the queue, add them and push it back in the queue. We will do this till the size of the queue becomes 1. He then asked me to write the complete implementation of a priority queue, that is, the push, pop, top, empty, and size functions.

- [Devise a stack data structure that performs the following operations in \$O\(1\)\$ time](#)

[push, pop, top, max element, min element.](#)

I first gave a solution by using three stacks one for the actual stack, one for min element, and one for the max element. He asked me to optimize the space complexity. So, I gave a solution where we encode the min element in the stack itself.

Round 4 (Technical): In this interview, I was asked questions from literally all topics. He started with networks and asked me all the steps that happened when I type www.google.com. Then, he went on to ask what happens in the OS when I type this and how does the OS respond to this high priority process. Basically, he was asking about the scheduling policy used for scheduling a high priority process. He then asked me what happens in the server and possible causes of bottlenecks in servers. Then, he moved on to OOP and asked me about the importance of inheritance and interfaces in OOP. He then asked me two coding questions.

- [Given a linked list, return a new linked list such that the order of nodes is 1->n-1->2->n-2 and so on.](#)

For this, I gave the solution to split the linked list in the middle, reverse the second half and traverse both the halves alternately.

- Given a hashmap, we need to print the hashmap in sorted order.

For this, I didn't know the concept of a linked hash map in Java. He gave me the concept of a linked hashmap as I was stuck. I gave a solution to add the key-value pairs in a vector, sort them, traverse the vector and populate the next pointers in the nodes of hashmap. Then, by just traversing the hashmap, we will get a sorted order of nodes.

Round 5 (Behavioral + Technical): In this round, the interviewer asked me many behavioral questions. Some of them were about the difficulties I faced in a project, how I solved them, and how I will handle disagreements in a group. Then, he asked me two coding questions.

- [Find the number of couplets and triplets in an array whose sum is 0. I gave him the two pointer approach for solving this problem.](#)
- [Given a string, I have to find the longest substring with no repeating characters.](#)

This is a basic sliding window problem where the window will only contain non-repeating characters. We will update the max length if the size of the window becomes greater than the max length.

In the last round, the interviewer was more interested in my approach than the code itself. Also, he asked a lot of behavioral questions in the last round. This can be considered as an HR round, although it was not explicitly mentioned anywhere.

Overall, the interviews were fascinating and whenever I got stuck somewhere, the interviewer would guide me in the right direction. In the end, 7 people were shortlisted for the job, and luckily, I was one of them.

I would suggest doing the amazon interview questions from GeeksforGeeks and interviewbit as they contain all the types of questions asked to date. The archive interview questions in GeeksforGeeks are very helpful as it contains all types of problems with varying difficulty. I would also suggest revising the concepts of Networks, OOP, OS as they might ask questions from those topics too.

My Personal Notes

Add your personal notes here

Save

,