

Amazon Interview Experience | Set 302 (On-Campus)

- Difficulty Level : \n[Medium](#)
- Last Updated : \n08 Jul, 2019

0th Round: (Online): 20 MCQs + 2 Coding Questions:

MCQs Topics: OS Page fault, Waiting time (RR Scheduling), Paging, Semaphores, etc.

DS : Hashing (simple chaining based numerical)

Aptitude : 1 Probability question, Puzzle: 1 question, C: 2 questions, etc.

There were 4 rounds of interviews(all technical). Each round was for 45-50 minutes.

1st Round:

In this round the interviewer asked me the below two questions.

1.a) Given a number find the next greater number with the same set of digits.([Find next greater number with same set of digits](#))

I had to write the complete code covering all the edge cases.

Since I had practiced this question, it didn't take much time to code for it.

1.b) Given a Binary Tree, convert this to a tree where each node contains the sum of the left and right sub trees in the original tree.([Convert a given tree to its Sum Tree](#))

For this question, first I discussed the approach, confirmed the edge cases and then wrote the code using recursion.

2nd Round:

This round started with questions from Operating Systems.

The interviewer asked me questions on Process and Threads, Process Scheduling, Mutex and Semaphores and Memory Management.

Some of the questions that were asked:

- Difference between process and thread.
- Advantages and disadvantages of using threads.
- Difference b/w Scheduler and Dispatcher.
- Name all the process scheduling algos and explain the Round Robin Process Scheduling.
- At max how many threads can be active at a time in a machine?
- What is race condition and how do we avoid it?
- Difference between Semaphores and Mutex?
- What is secondary memory and why do we use it?
- What is thrashing?

Then he asked me questions on Computer Networks.

- Explain OSI reference model and write all the layers present in this model along with the protocols used at each layer.
- Difference between TCP and UDP.

Finally he gave me a problem and told me to first discuss the approach with him and then code the same covering all the edge cases.

The problem was as follows:

[You are given an infinite stream of numbers, at any instant of time you have to print the earliest non-repeating number.](#)

eg: 2 3 4 2 5 6 output 3

2 3 4 2 3 7 8 output 4

I discussed my approach with him, he told me to further increase the efficiency and gave me hints. With his hints, I came up with the final solution (using Doubly Linked List and HashMap) and wrote the code.

He evaluated it and spotted 3 errors.

3rd Round:

In this round the interviewer asked me questions on the Internship project.

He told me to briefly explain the complete project and asked various questions on the same.

Then he gave me a coding question and asked me to first explain the approach and then code.

The question was,

You are given an infinite stream of numbers and an integer k, at any instant of time you have to print the top k elements scanned till that point.

I explained the naive approach of using an array of size k and then inserting the elements at proper position thus making sure

that the array always remained sorted. Complexity $O(k)$

He was not satisfied with this solution and told me to further improve it i.e. aim for $O(\log(k))$.

I did some brainstorm for a while and then gave the final solution to him using a min heap of size k.

He also asked me to write down complete code for min heap.

The next question was on Caches.

He asked me to tell different page replacement policies.

Then he asked me, How would you implement LRU Cache?

I told him that it can be implemented using Priority Queue.

Then he cross-questioned me on Priority Queues, its internal implementation and complexities.

He didn't ask to code for it.

4th Round:

The interviewer gave me 45 minutes to solve the below coding problem.

You are given an infinite stream of numbers, at any instant of time you have to print the median of the elements scanned till that time.

Median of an array here refers to the middle element of the array after sorting.

First I told him that I would maintain a sorted array, thus order of insertion would be $O(N)$ and order of query would be $O(1)$.

He told me to further optimize it.

I thought of using heaps and somehow try to achieve $O(\log(N))$ for both insertion and query, but then he told me that it can be further optimized.

He told me to aim for $O(\log(N))$ and $O(1)$ for insertion and query respectively.

Suddenly I got the idea that I could make use of a minheap and a maxheap to solve the problem.

I discussed my approach with him. He was impressed.

Finally I coded the problem.

At last he asked me, Given an inorder traversal of a BST, can you build the tree? I told him yes. Then he asked me, How many such trees are possible?

I knew answer to this question, as it was taught in the class.

I told him about Catalan Number.

Verdict: 3 selected out of 33 interviewed.

If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

[All Practice Problems for Amazon !](#)

My Personal Notes\ *narrow_drop_up*

Add your personal notes here

Save