

CVENT INTERVIEW EXPERIENCE (INTERNSHIP+FTE)

5th-10th AUG 2022

Dhirender Kumar

MSC 2021 - 2023

OA ROUND 1: ONLINE MCQ TEST

First round was an MCQ round consisting of questions based on C, C++, JAVA, DBMS and aptitude questions.

Level: Easy to Medium

Total no of questions: 30

Time duration: 60 mins

63 students were shortlisted for next round

Brush up the basics before OA.

OA ROUND 2: ONLINE CODING TEST

This round had only one programming question. Two arrays were given as input which represents the height of players. Array 1 of n size represents the height of players of team 1 and array 2 of n size represents the height of players of team 2. We were required to build a team from both the arrays of size n in such a way that it maximizes the total height of players. Players should be chosen in alternate order i.e. once a player is chosen from array1, next player will be selected from array2. Also the index of player chosen should be greater than the index of current player i.e. if the current player is chosen from the i^{th} index of array1 then the next player will be selected from index $i+1$ or later of array2.

Question link:

<https://www.geeksforgeeks.org/maximum-sum-combination-from-two-arrays/>

26 students were shortlisted for interview rounds.

ROUND 3 : TECHNICAL ROUND 1

My interview was taken by Meenakshi Dahiya(Engineer Manager). She first introduced herself and then asked me to introduce myself while looking at my CV. After introduction we jumped to DSA and she gave me a question:

Problem 1: given an input array , output an array such that it consists of count of how many times you can move an element to the right. You can move an element to the right only if the element on the right is greater.

e.g. I/P: 5, 9, 2, 1, 7

O/P: 1, 0, 0, 1, 0

I hadn't done this problem earlier so I started with the brute force solution with nested for loops. She then asked me the time and space complexity of this approach. I told her:

T.C. $O(n^2)$

S.C. $O(n)$

Then she asked me to optimize it, i tried to think about it for 30 seconds or so but nothing was coming to my mind, so I told her that I'm not able to optimize it. She told me to take some time and think about it. After 2minute or so, I roughly mapped the problem to finding a smaller element to the right. Then I told her that this is what I'm planning to do, to which she asked me to write the time and space complexities along with the data structure that I'm planning to use.

T.C. $O(n)$

S.C. $O(n)$

Data Structures: Stack and pair(array)

Then after a minute or so, I told her my approach with the help of an example and she was satisfied with it. She then asked me to write the code for the same. After I was done writing code, we jumped to the theory questions.

What is your favourite subject ?

I said Operating Systems to which she asked me why OS.

I answered "We use Operating systems on a daily basis and I like to understand how things are working under the hood".

After this we jumped to the OOPS question.

What is abstraction?

I started with the definition and then how abstraction is achieved in java(using interfaces and abstract classes) and basic structure of interfaces and abstract classes(method implementations). Then I said "In Java 8 and later, we can define default methods in interfaces also"

If you can define methods in interfaces also, then what's the need of abstract classes?

I started with "Yes, there's an overlap in situations when we use interfaces and abstract classes." and then stated the difference between both(public, static and final variables in interfaces). She was satisfied with the answers.

Inheritance, when to use and its advantages.

I started the answer with the definition and then tried to explain it with the help of examples, but she was looking for some specific keywords in the answer and told me that these are the examples of inheritance. Then I stated the situations when we want to use inheritance and she was satisfied with the answer.

Polymorphism, compile time and runtime polymorphism.

I started with the definition and then explained both types of polymorphisms in detail, also touching on variable and method hiding.

She asked me if I have any questions for her, I asked about the feedback. She told me that they don't give feedback.

Then I asked for the tech stack they use. She told me that everyone here is a full stack developer. She also said that as a fresher it should not matter to you, you should be open to everything.

It's okay to be nervous during interviews.

Try to be as interactive as possible.

Even if you get a new problem, try to map it to some other problem that you know how to solve.

ROUND 4 : TECHNICAL ROUND 2

This round was taken by ANAND NIGAM (Director Of Engineering at Cvent).

Before starting the interview he asked me how my first round went and if I need anything. I told him that i was not able to give my 100% as I was nervous.

We then started with the question.

There's an orchid having N apple trees. Each tree has some apples on it. 2 people A and B went to orchid to pick apples s.t. A has to pick apples from X trees and B from Y trees. The constraints here are:

A and B have to pick from continuous trees

Trees from where A and B picks should not overlap.

We've to return the maximum number of apples that they can pick.

<https://leetcode.com/problems/maximum-sum-of-two-non-overlapping-subarrays/>

I was asked to write pseudo code for this problem.

I read the question once and asked him a question to confirm that there are always positive number of apples on each trees(because you never know).

He asked me what I think, I answered that in reality apples can't be negative. He replied "you've answered your question".

Then I started with generating a test case and explaining the brute force approach with the help of test case. He was satisfied with the approach and then asked me to write the pseudo code for the same. While writing the pseudo code, I asked him a question about an edge case(what if $X+Y < N$). He again asked me what I think should happen. After few seconds I replied that in this case the sum of apples on all the trees should be returned. He agreed to it and I continued writing the pseudo code. After I was done with the pseudo code he asked me the time and space complexity of my solution.

T.C. $O(n^2)$

S.C. $O(n)$

I was calculating and storing the running sum of subarrays of length X and Y. He asked me for the time complexity of this operation. I replied with $O(n)$. He then asked me what if instead of calculating and storing it earlier, we calculate it again and again, what will be the time complexity. I answered $O(n^2 \cdot (x+y))$. He then asked me that in my approach I'm using arrays of size N to store the sum of subarrays, due to which there's wastage of storage. I answered that to resolve this, we can use array of smaller size(equal to number of possible subarrays of length X and Y) and change how indexing is done. He was satisfied with the answer. He then asked me if I have any question for him. I asked what are the type of problems/challenges they face on a daily basis.

7 students were offered intern+FTE