Richard Kullu(Msc 2021-23)
Cvent Interview Experience (Intern + FTE) [1st week Aug 2022]

Round 1.1 (**Online assessment**)
30 question related to:
1. Find error, file descriptor, open and close files in C/C++ program
2. DBMS (queries and theoretical)
3. Quant Aptitude (simple CAT exam like question)
4. More CS fundamentals.

This was MCQ round in the wheebox website with camera and mic on.
More than 90 participated in this and approx. 60 got selected for the next round.

Round 1.2 (**Coding Round**)
Given two arrays each number is non-negative. This is how to imagine these two arrays are on top of each other. (There was a limit of only 50 compilations at max)

1	16	2	0	43
4	2	7	2	0

find the max sum value (by iterating the whole array) you could get if you could choose any value from array at index ‘i’ such that 1 <= i <= n (size of array) but can’t choose both values from the same index. Ex: Can’t pick 1 and 4 to make value 5, as they are of the same index ‘1’ And I can't choose a value from one array simultaneously. Ex: You can’t choose values 1 and 16 from array 1 to sum 17 as they are in the same array, same for array 2.

There are many ways to make sum with these constraint, Ex: 1 + 2 + 2 + 43 = 48
Or 4 + 16 + 2 + 2 + 43 = 67.

I first thought it was a greedy problem, and i could get optimal value from zig-zag pattern and coded a greedy solution in the first 5 min then, it didn’t pass every test case, then I realized they won’t give a simple greedy problem for a time limit of 40 min, as also I noticed there are many ways to choose num to make a sum, and for some test cases there can be only 1 optimal solution and some other other there can more than 1 optimal solution.
From this i was sure it was DP problem, I made a 2d-array with 2 rows and n columns, so my idea for code was if i pick value at index ‘i’ either from array 1 or 2, then i could use optimal sum value considering ‘i - 1’ elements, and keep on going till n. I took a huge amount of time (20 min) to solve this simple problem because of my mistake regarding initialization of the initial dp array ‘index’. But after testing for 3 testcase, every answer given by my code was correct.

Remember to not get frustrated if not getting a solution, and don’t make silly coding mistakes like using ‘j’ instead of ‘i’ as they look similar on screen, make sure to correctly initialize every. And visualize the whole code working and then code. Both rounds were held in the afternoon.

Round 2.1 (**On-site**) (**TLDR Incl.**)
This was also a coding round. My interviewer's name was Anand Nigam. He first asked a simple question and asked which languages I'm good at and also about projects (not too much about projects). He took a paper and started drawing a rod which we could cut and each cut was allotted a price which we could sell that rod for. He told me to find the maximum value I could sell this rod of size ‘n’.
He explained this problem very slowly making sure I understood everything about the problem. Remember to tell your observation about the problem out loud to the interviewer. First thing i realized was there are many ways to cut this rod, example N = 2, arr[] = 10, 5
Meaning 1 size rod price is 10, size 2 whole rod is just 5, i could either sell this rod for 10 + 10 of size 1 + 1, or price ‘5’ of size ‘2’, optimal is 20(1 + 1) and it could be applied to any size.
I told my approach that i could check for all possibilities of cutting like if cut rod of size 1, i could use optimal value of size ‘n - 1’, if i cut for size 2, i could use optimal value for size ‘n - 2’ and so till cut of size ‘n’ and optimal value of size ‘0’. And to get optimal value I could use a dynamic programming look-up table.
He told me don’t speak out technical jargon of dynamic programming explain it in concept and idea type, then i explained many examples of cutting, then he asked to write a math recurrence formula, then he also asked me to write code as a function however i want, and i had written a bottoms-up approach, he keeps on asking very minute details about the code as to why i’m using 2-loops, why i’m iterating till ‘i’, and keeps on going to minute details about my code.
Then he also asked me to write a top-down approach. I asked him if I should use a dp-lookup table or not. He said why I was using it, I told because of the better time complexity. Then I had written top-down code in a function, then he again asked for minute details about my code. He keeps on asking me about minute details of my code and correctness of my code. As this was a 1-to-1 onsite interview I was able to explain to him better, as in online interviews, there are possibilities of network issues and sometimes the interviewer never turns on camera.
Then he asked me if you want to ask any questions regarding the company.

TLDR: First explain approach, explain correctness of code, Every why and why not, as why 1 or 2 loops, why 2-d or 1-d dp table.

There were 8 people in on-site round 2, 4 people were selected for the next round.

Round 2.2 (**On-site** Same day after 20-30 min)
My interviewer name was Arvind, he asked me about my preferred languages and projects, he then asked my about react js dom, how is it different from browser dom, he then started with coding question he explain question in paper give an array of integers (+ve, -ve) find maximum subarray sum index’s.
He then asked me about, Given a number ‘N’ to find if it's a perfect square,
if yes return 1 else 0. But I can't use any library functions or multiplication or division.
I had asked if i could use for loop

In this one needs to notice the pattern as every perfect square number comes from pattern
1,4,9,16,25,36
+3,+5,+7,+9,+11 (each one has a difference of 2), can be coded with a for loop easily.

ADVICE FOR MY JUNIOR'S IN MSC COURSE

1. As you will have less time to prepare for placement, if you have never done any leetcode problems, start practicing leetcode problems **DAILY** from the day you read this statement. Practice the **concept** as memorizing code is futile. (300+ medium level would be fine)
2. Learn to code every data structure from scratch (Priority Queue, Binary Search Tree, Linked List etc) and also OS concepts like multi-threading.
3. Revise CS fundamentals (DBMS, OS, CN) almost daily.
4. Make good projects (at max 3-4) (i made it on react native and react js)
5. Learn to make a good resume. Whatever you have written on your resume, make sure you know everything about it.