

Online Round

This round was conducted on CoCubes. There were 3 coding problems and 70 minutes in total. The problems that I got were as follows.

Problem 1 (3 points)

You are given an integer N , and you have to compute Y such that:

$$Y = \sum_{i=1}^N \sum_{j=1}^i j$$

The statement was not so direct, but it was very easy to see. Also, it was guaranteed that the result would always fit in an integer (32 bit signed integer).

So, I made a linear time solution with constant space using the formula (for each integer $i \in [1, N]$):

$$\sum_{j=1}^i j = \frac{i \times (i + 1)}{2}$$

Problem 2 (3 points)

Given an $N \times M$ matrix filled with 0s and 1s only, if an element is 0, set it's entire row and column to 0s. Do it **in-place**.

<https://leetcode.com/problems/set-matrix-zeroes/>

Problem 3 (5 points)

Given two non-empty binary trees s and t , check whether tree t has exactly the same structure and node values with a subtree of s . A subtree of s is a tree that consists a node in s and all of this node's descendants. The tree s could also be considered as a subtree of itself.

<https://leetcode.com/problems/subtree-of-another-tree/>

Advice:

- 1) Don't rely on sample tests, and pretests present on any platform. Do rigorous case analysis and testing on your own.
- 2) This platform didn't have an option to run custom tests, so always keep in mind the edge cases, places where your code can go wrong.
- 3) Keep calm, it'll always help.

Technical Round 1

This round was conducted on Google Meet Video Call, and it was a 45 minute round. It went very quick because I was supposed to solve 4 coding problems (not much of code, approaches and code segments mostly). No introduction, nothing, he just started off with the problems. But I still have to say he was friendly.

Problem 1

Given n ropes of different lengths, we need to connect these ropes into one rope. We can connect only 2 ropes at a time. The cost required to connect 2 ropes is equal to the sum of their lengths. The length of this connected rope is also equal to the sum of their lengths. This process is repeated until n ropes are connected into a single rope. Find the minimum possible cost required to connect all ropes.

<https://www.geeksforgeeks.org/minimize-the-sum-calculated-by-repeatedly-removing-any-two-elements-and-inserting-their-sum-to-the-array/>

Problem 2

Given two strings S and T , return if they are equal when both are typed into empty text editors. $\#$ means a backspace character.

Note that after backspacing an empty text, the text will continue empty.

<https://leetcode.com/problems/backspace-string-compare/>

I solved it using a two-pointers approach in linear time (one pass) and constant space (the most efficient way (for both space and time)). Then the interviewer asked me if I could solve it with some other technique, because most of the candidates used the same technique.

I told him that we can, at the expense of using linear extra space, maintain two stacks, one for S and one for T . Then we process both the strings. Whenever we encounter a $\#$ character, we pop elements from stack (if it is non empty), otherwise we push characters onto the stack.

And then at the end, we compare both the stacks.

He was satisfied that I came up with another approach for the same problem.

Problem 3

Construct an efficient data structure which supports insert, search and delete (a delete operation has to search first) operations in $O(1)$ time.

I told him that a Hashmap supports all of these operations and with the same efficiency. But he wanted me to do something different. Then I came up with a [trie](#) solution, but here he was concerned with the space required for a trie vertex, because he said that the data can be anything, so the character set for a trie vertex can be big (let's say up to 256). Then I came up with an ad-hoc data structure using a Hashmap and a Vector, where I could find the index of an element

from the Hashmap, and keep the data in the Vector. Also on deletions, I could move elements to the end, which could be used again during insertions to save space. Also if we didn't have free elements present in the Vector, then we could always insert new elements at the end in constant time. He was satisfied with my solution.

Something very close:

<https://www.geeksforgeeks.org/design-a-data-structure-that-supports-insert-delete-search-and-getrandom-in-constant-time/>

Problem 4

Given an array of digits (0 to 9). Find the largest number that can be made from some or all digits of the array, and is divisible by 3.

This was trivial, some basic observations, some case work and basic knowledge about modular arithmetic.

A similar problem:

<https://www.geeksforgeeks.org/find-largest-multiple-3-array-digits-set-2-time-o1-space/>

Also, at the end he asked me if I had worked with some Object Oriented language. I told him about C++ and Java. He didn't ask me anything about them, and the round ended.

Advice:

- 1) They help you figure out things, you just have to be open about your approaches. Don't be blank, just think out loud.
- 2) Try to move with solutions elegantly, don't just throw anything, think twice and figure out cases.

Technical Round 2

Another one on Google Meet. This time the interviewer asked me to introduce myself, then she asked about the department in a very friendly way.

Then I was asked to tell about all Object Oriented Programming concepts I knew along with an example for each, and if I had worked with any language which supported these concepts, I told about C++ and Java.

Problem 1

She asked me to share my screen and open LeetCode, she gave me a problem link, I opened it and she first checked the Submissions tab, I hadn't attempted it so we proceeded with it (had I attempted it, she would have given another one).

<https://leetcode.com/problems/first-missing-positive/>

Yes, it is a LeetCode Hard, I was very surprised with the way this interview was going ahead. And I was expected to implement the most efficient solution (linear time with constant space) and pass all the tests live on LeetCode. Also, I had to explain everything and dry run my code before submitting.

I would say she was not much of a help, she also did try to confuse me at various instances, but I kept calm and I was confident enough with my approach, I coded it elegantly and got my solution accepted in very less time.

Problem 2

This was a very direct problem:

<https://www.geeksforgeeks.org/nth-node-from-the-end-of-a-linked-list/>

I just gave the two-pointers approach straightaway and she was satisfied.

Problem 3

She asked me a puzzle, and I was able to come up with a solution very quickly (I had heard about something very similar).

Puzzle: <https://www.popularmechanics.com/science/math/a24208/riddle-of-the-week-6/>

Solution:

<https://www.popularmechanics.com/science/math/a24210/solution-to-riddle-of-the-week-6/>

Advice:

- 1) Have faith in yourself if you feel your approaches are correct, but yes, always think twice before doing so (if you're absolutely sure about it).
- 2) Try explaining a few cases every time, also try to comment code while you type, so that the interviewer can also see what you mean.

Technical Round 3

Again on Google Meet. It started with a friendly introduction, then he opened my resume and asked me to tell about what I liked. I told him about my inclination towards Mathematics and Competitive Programming (CP). Also my resume had a few of my Competitive Programming achievements so he asked me about my CP journey. He asked me mathematics and DSA problems only (maybe because of my CP experience). Nothing about my projects or from any Computer Science Fundamentals (CSF).

Problem 1

<https://leetcode.com/problems/max-points-on-a-line/>

He stated it very vaguely, I had to dig in a lot. Also he mentioned a few key details incorrectly (maybe he messed up in wrapping the original problem, or he was himself not so clear with it) which made the problem very different (and difficult to come up with a solution) from the original problem. He first mentioned that I have to minimize the error (sum of distances of all points from the line), which made it more of a regression problem. I had also asked him explicitly if I were to place maximum points on the line, he denied. I used calculus to solve this version, though he was impressed but it was not what he wanted to ask.

Then I just told him as a side note at the end that if we were to maximize the points lying on the line, then we could have hashed the slopes and intercepts of all possible pairs of points forming a line, and then count which line had the maximum points stored in a Hashmap. This solution works in quadratic time and linear extra space with some observations and optimizations. He was both impressed and satisfied at this point.

Problem 2

Now he just mentioned a direct problem.

<https://leetcode.com/problems/find-median-from-data-stream/>

He asked me if I had heard about it. I told him yes, I have. I also told him that this is one of the most interesting interview problems that I have heard of. He asked me the reason, I told him that it could be solved in many beautiful ways.

First I told him about the standard solution using two heaps (one min heap and one max heap). He was satisfied.

Then I told him that I know about some advanced data structures which I learnt about during my CP journey. He got very interested in knowing about them.

First advanced DS that I used to solve the same problem was a Fenwick Tree (Binary Indexed Tree). I told him that we can use binary search with a Fenwick Tree to support Order Statistics properties.

<https://www.geeksforgeeks.org/order-statistic-tree-using-fenwick-tree-bit/>

Then I told him that we have a prebuilt Order Statistics Tree in GNU's Policy Based Data Structures (PBDS) Library in C++ which supports Order Statistics operations in logarithmic time.

<https://www.geeksforgeeks.org/ordered-set-gnu-c-pbds/>

Now, at this point he was very much impressed and he also gave me a lot of compliments for my deep knowledge about such advanced data structures.

Now, this was supposed to be a 30 minute round, about 35 minutes had already passed. And at this point he shared another problem with me, and he said that it was a special problem.

Problem 3

(This is a very short and extracted version of that problem, originally it had a lot of description)
You have an infinite 2D plane containing a lot of points, you have to find a point which is not in the set of given points.

My solution:

We can use trie data structure, maintain all points in our trie by compressing the coordinates, and then finally do a custom inverse search in the trie which will give us a point not in the plane.

He said, *Good*. Now he said that imagine if we had a billion points and we couldn't store them in any data structure. Also, we have to find a point which is as far as possible from all of our points.

Now, I was taken aback here. Because this problem is very closely related to a research problem that I know about, and it is an NP problem. I asked him about it, and he said yes, this problem is one of Paytm's ongoing research problems that they're working on. Though it was very abstract, he said that they were working on something similar. Also, he mentioned that he wanted to test how a candidate with a very good DSA and CP knowledge performs on a real world problem.

So, I told him about centroid based clustering techniques and about maximizing the minimum radius (or distance) of a chosen point greedily. And a lot of such other techniques around clustering only.

He said he was very satisfied with my performance, and also appreciated me for a few points he noticed during the interview.

So, it was a 30 minute round which went for about an hour. And it did go very nice.

Advice:

- 1) The interviewer might not be prepared well with a problem, it's your responsibility to satisfy him. Ask more meaningful questions if you're not clear. Try to know about what he really wants to hear from you.
- 2) Widen your horizons, don't stick to limited approaches or techniques. You might come up with something better by doing so.