**ARUNABH PRAKASH (MSC 2022-24)**

**CIENA INTERVIEW EXPERIENCE (C/C++ DEVELOPER)**

Company came for two profiles: C++ developer and Java developer.

There were total 3 rounds: OA and 2 technical interview rounds.

**ONLINE ASSESSMENT**

I was asked 3 coding questions, other set contained 45 MCQ questions.

Q1. Given a string containing only uppercase characters, return how many

times you can remove letters (in any order) from the string and make the

string “BANANA” from it.

Example:

Input: NAABXXAN Output: 1

Input: NAANAAXNABABXNNBZ Output: 2

Input: AAXAYZB Output: 0

Q2. Given a string S return alphabetically smallest string that can be

obtained by removing exactly one letter from S.

Example:

Input : S=”hot” Output: “ho”( because it is alphabetically smaller than “ht”

and “ot”).

Q3. Given string S count the number of different letters that appear in both

uppercase and lowercase where all lowercase occurrences of the given letter

appear before any uppercase occurrences.

Example:

Input: S=”aaAbcCABBc” Output:2 ( a and b are lowercase letters coming

before uppercase)

Input: S=”xyzXYZabcABC” Output:6

Input: S=”ABCab” Output:0

**Optimizing Time Complexity and Space Complexity was not part of selection criteria, even brute force submissions were accepted.**

**TECHNICAL INTERVIEW 1 (Lasted 1 hour):**

Interviewer first introduced himself (don’t remember the name but he was senior manager in Ciena) then asked me to introduce myself.

*I told my name, course am currently pursuing, about my UG degree, my cs hobbies (Web development, AI and ML) and non-cs hobbies (polity, economy, emerging tech., fintech and share market).*

Then, sir asked me to explain my best project.

*I explained my AI project: Financial Advisor Chatbot (reason being: novel approach, practical use and some of the latest libraries used). While explaining, I focused on motivation to do the project, concept, implementation and its application.*

Then, sir asked me some questions related to it and problems I faced.

*Tip: always do SWOT (strength, weakness, opportunities and threat) analysis of your projects, interviewers are more interested in these than actual code.* ***Reason being: if you really did your project, you might have got stuck somewhere else it is more likely that you copied running code from somewhere online.***

*In first round, apart from project related discussions, I was only asked 2 DSA questions and some HR questions in between.*

DSA questions:

Q1: Given a word stored in singly LinkedList, check whether it is palindrome or not. (Both TC and SC was already given by the sir i.e. O(n) and O(1) respectively, so he was not looking for any brute force solution).

*Initially, I couldn’t think of anything so I started with Brute force even though TC and SC were already given. Then finally I caught on to it, my solution was to find the middle of LinkedList and reverse linked list before middle and then start from middle and compare letters on right and left side respectively.*

***Tip: always ask as many questions as you can and apply all approaches, I started by asking if I can use 2 pointer approach, then asked if I can modify this to double linked list and so on. They are more interested in what you can think than whether you can answer with exact correct approach they have in mind. That’s why people who answer right off the bat because they have done the question will just be given new question until they reach a point where they have exhausted all the questions or until the interviewee is stuck.***

*After this, sir asked me to write the whole code to see if I can code my answer.*

*There were some HR questions in between such as how many members are there in my family, where does my brother work etc. then he moved on to next question.*

Q2. Given a singly LinkedList of numbers, find number at “x” position from the end. TC and SC were again given in the beginning itself, i.e., O(n) with strictly 1 iteration and O(1) respectively.

*I answered with O(n) but with 2 iterations. Then, I used 2 pointer approach to find middle and and reverse LL before middle and if middle position is less than end -x then go back and look for it else go forward towards end side.*

Then sir asked me to do it without modifying.

*I couldn’t think of anything so I asked for hint.*

He said use two pointers such that when one reaches end other should be at correct position.

*I caught on, and then wrote whole code.*