



Interview Experience

Name: Ankush Das

Job Role: Post-Graduate Engineer Trainee(PGET)

The entire process took place from September 23, 2024, to October 4, 2024.

Online Assessment: 23 Sept 2024 - 25 Sept 2024

Interview: 3rd October 2024

Result: 4th October 2024

Eligibility criteria:

CGPA: 6+

Backlogs: No backlogs

MCA, MTech(CS) BTech (CS & IT) were the only departments eligible.

After the eligibility criteria were met, a total of 110 candidates were selected. Following the online assessment, 90 individuals were shortlisted for an interview. Ultimately, 37 students were selected.

(The majority of students selected were BTech students, and the number of students selected from MCA was lower compared to the previous batch. This is because there have been relatively fewer placements in the CS & IT department till that time.)

Online Assessment: (23 Sept 2024 - 25 Sept 2024)

On September 23, 2024, we received the Hirevue link for the online assessment. The assessment is scheduled to commence within a 48-hour time-frame, during which we are required to complete the test.

Essentially, there are a few pre-recorded video questions to which you must provide your response. These responses will be evaluated later by an interviewer or assessor. Preparation time of 1-2 minutes is given and a 3-minute recording window is provided for the answers.

The video interview is approximately 90 minutes in duration. You have the flexibility to conduct the interview from any location at any time, utilizing any mobile device or computer.

This assessment includes two section:

1. Two coding rounds:

- Process a sorted array of positive integers. The objective was to discern and display all unique number pairs from this array that, when combined, equate to a pre-specified number (X). These pairs had to be presented in an ascending sequence.

Approach:

The problem involves processing a sorted array of positive integers to identify unique pairs that sum up to a given target XXX. Here's how I approached it:

1. Understanding the Problem:

- We have a sorted array, which is beneficial because it allows us to efficiently find pairs without needing to check every possible combination.
- Our goal is to find unique pairs of numbers that add up to X and present them in ascending order.

2. Using a Two-Pointer Technique:

- Given that the array is sorted, one effective approach is to use a two-pointer technique.
- We initialize one pointer at the beginning of the array and another at the end.
- The idea is to evaluate the sum of the elements at these two pointers:
 - If the sum equals X, we have found a valid pair.
 - If the sum is less than X, we move the left pointer one step to the right to increase the sum.
 - If the sum is greater than X, we move the right pointer one step to the left to decrease the sum.
- This continues until the two pointers meet.

3. Avoiding Duplicates:

- To ensure that we only collect unique pairs, we can use a set to store pairs in a consistent format.
- Alternatively, since the array is sorted, once we find a valid pair, we can skip over any duplicates by moving the pointers past any repeated values.

4. Complexity Analysis:

- The time complexity of this approach is $O(n)$ because each element is processed at most twice (once by each pointer).
- $O(n)O(n)$
- The space complexity is $O(1)$ for the two-pointer method, excluding the space used for storing results, which can be considered constant space.
- $O(1)O(1)$

5. Implementation:

- In the implementation, I read the array and the target sum from input.
- Then, I applied the two-pointer technique to find and print all unique pairs that sum to X.

Code:

```
#include <bits/stdc++.h>
using namespace std;

void findUniquePairs(const vector<int>& arr, int X) {
    unordered_set<int> seen; // To track seen numbers
    unordered_set<int> pairs; // To avoid duplicate pairs

    for (int num : arr) {
        int complement = X - num;
        // Check if the complement exists in the seen set
        if (seen.find(complement) != seen.end()) {
            // Create a pair in ascending order
            int smaller = min(num, complement);
            int larger = max(num, complement);
            pairs.insert(smaller * 10000 + larger); // Store pairs as a single number
        }

        // Add the current number to the seen set
        seen.insert(num);
    }

    // Output the unique pairs
    cout << "Unique pairs that sum to " << X << ":" << endl;
    for (const auto& p : pairs) {
        int smaller = p / 10000;
        int larger = p % 10000;
        cout << "(" << smaller << ", " << larger << ")" << endl;
    }
}

int main() {
    vector<int> arr;
    int n, X;

    cout << "Enter the number of elements in the sorted array: ";
    cin >> n;
    cout << "Enter the elements of the sorted array:" << endl;
    for (int i = 0; i < n; ++i) {
        int num;
        cin >> num;
        arr.push_back(num);
    }

    cout << "Enter the target sum (X): ";
    cin >> X;

    findUniquePairs(arr, X);

    return 0;
}
```

- Identify the intersection of two sorted lists of integers, each integer separated by a semicolon
- Approach:**

1. Problem Understanding:

- We need to identify common elements (the intersection) between two sorted lists of integers, where each integer is separated by a semicolon.

2. Input Parsing:

- First, we'll read the input strings for the two lists.
- We'll parse these strings to convert them into vectors of integers. Since the integers are separated by semicolons, we can use a string manipulation technique to split the strings at the delimiter.

3. Using Two Pointers:

- Given that both lists are sorted, we can utilize a two-pointer technique to find the intersection efficiently.
- We initialize two pointers, one for each list, starting at the beginning.
- We will compare the current elements of both lists:
 - If the element in the first list is smaller, we move the pointer in the first list forward.
 - If the element in the second list is smaller, we move the pointer in the second list forward.
 - If both elements are equal, we have found a common element. We add it to our result list and move both pointers forward.

4. Termination Condition:

- We continue this process until we reach the end of either list. The sorted nature of the lists ensures that we don't need to backtrack, making this approach efficient.

5. Output the Result:

- Finally, we will format the output by joining the common elements with semicolons and print the result.

Time Complexity:

- The time complexity of this approach is $O(n + m)$, where n and m are the lengths of the two lists. This is efficient since we are essentially traversing both lists once.

Space Complexity:

- The space complexity is $O(k)$, where k is the number of common elements, since we need to store them in the result vector.

Code:

```
#include <bits/stdc++.h>

using namespace std;

vector<int> parseInput(const string& input) {
    vector<int> numbers;
    stringstream ss(input);
    string token;

    while (getline(ss, token, ';')) {
        numbers.push_back(stoi(token)); // Convert each token to an integer
    }

    return numbers;
}

vector<int> findIntersection(const vector<int>& list1, const vector<int>& list2) {
    vector<int> intersection;
    size_t i = 0, j = 0;

    while (i < list1.size() && j < list2.size()) {
        if (list1[i] < list2[j]) {
            i++;
        } else if (list1[i] > list2[j]) {
            j++;
        } else { // list1[i] == list2[j]
            intersection.push_back(list1[i]);
            i++;
            j++;
        }
    }
}

return intersection;
}

int main() {
    string input1, input2;

    cout << "Enter the first sorted list of integers (semicolon separated): ";
    getline(cin, input1);
    cout << "Enter the second sorted list of integers (semicolon separated): ";
    getline(cin, input2);

    // Parse the input into integer vectors
    vector<int> list1 = parseInput(input1);
    vector<int> list2 = parseInput(input2);
```

```

// Find the intersection
vector<int> intersection = findIntersection(list1, list2);

// Print the intersection
cout << "Intersection of the two lists: ";
for (size_t i = 0; i < intersection.size(); ++i) {
    cout << intersection[i];
    if (i < intersection.size() - 1) {
        cout << ";" // Separate with semicolon
    }
}
cout << endl;

return 0;
}

```

2. HR question round (which roughly takes 15 minutes)

Ensure that you are in a well-lit and undisturbed environment. Additionally, verify the charging status of your device and the network connection.
 Disable all browser extensions, turn off all notifications on the device and uninstall OBS Studio (Open Broadcaster Software). And restart your device.

- Describe how you solved the second coding challenge, walk us through your logic and describe alternatives that you consider but did not use.
- Tell us about the education, training, and work experience that have helped prepare you for this position.
- Is there anything else you would like to tell us about how your background and experience matches the qualifications for these positions.

Round 1: (3rd October)

We anticipated that they might request us to execute the project on our laptop. However, we were not required to bring any laptop or device.

1. Find the fastest 3 horses Puzzle

There are 25 horses among which you need to find out the fastest 3 horses. You can conduct a race among at most 5 to find out their relative speed. At no point, you can find out the actual speed of the horse in a race. Find out the minimum no. of races which are required to get the top 3 horses.

Solution: <https://www.geeksforgeeks.org/puzzle-9-find-the-fastest-3-horses/>

2. What is the difference between delete and truncate.

3. I was asked about the definition of palindrome and was even asked to code it using pen and paper. This is very common in many interviews.

4. The Interviewer asked me to find depth and breadth of a binary tree, which the interviewer had drawn on the paper, Also asked me the code for the same.

5. Asked me regarding my MCA CET rank and preparation, this was a plus point for me, as my rank was very good?

6. What is state management? And how did you managed state of your project? What is redux and how do you implemented it in your project?

7. She asked me if I implemented authentication and authorization in my project, and if yes, how so?

8. How does MongoDB's sharding mechanism work, and what are the implications on performance, scalability, and data distribution?

9. I was lastly asked a simple SQL query of easy level difficulty.

Overall this round lasted for 20-30 minutes. I was very confident after this round, as I had answered all the questions and my explanation of Redux had impressed the interviewer.

Round 2: (3rd October)

The Interviewer was a very senior person in the bank and was very friendly with me. He asked me if I lived in Mumbai. I said no, but I used to live in Pune. I currently live in Mumbai, and coincidentally, he lives in the same place as me in Pune. So that was a good start.

- 1. He started asking me questions regarding my resume, What does your coursework include? What is Data mining and what the techniques used for data mining?**
- 2. What are the certifications mentioned and what did you learn in it?**
- 3. He asked me regarding my project? What problem does it solves, why did you choose this particular project?**
- 4. He asked me the database structure of my project?**
I explained him the schema, tables, views, and how does it solves for redundancy.
- 5. He gave me a complicated SQL Query(I honestly don't remember the actual query, but it was of medium difficulty. And the interviewer gave me hints and guided me throughout)**
- 6. He asked me questions regarding blockchain and crypto, some real life examples(I answered bitcoin and Ethereum) He asked me what is the main purpose of Ethereum.**
- 7. Then, he asked me about the implementation of blockchain technology within the bank and the associated security concerns. Also asked me a real life problem.**
- 8. For the Design question, he asked me to implement or design how would you calculate the angle between the hour and the minute hand. Also asked me to code it.**
The question seems very easy, but it not that straightforward and this required cross questioning with the interviewer to understand the edge cases and the constraints.
- 9. Finally, he inquired if I had any questions for him.**
Please do not skip this question, as it demonstrates your interest and engagement in the process. I inquired about the technologies he is involved in at the bank.

Post Interview:

After the second round, I was asked to stay for group discussion. I had not anticipated this third round. As the second round was highly scrutinizing, and I had very less confidence after this round. This round took nearly an hour, and I was the last person to be interviewed.

I was waiting for almost 2 hours after the interview, for the GD. As one group was already engaged in the GD.

In addition to myself, there was one more candidate for the GD. We were both quite nervous, as GD sessions involving only two individuals are very uncommon.

Furthermore, we were the final two candidates, and after a two-hour wait, we had started to lose hope.

After the final group's departure from the GD room, we were informed that we can leave, and the GD session would not proceed. Upon hearing this news, I was deeply disappointed, as I had anticipated that this would be final opportunity for us to get selected.

I was scared, as I had been diligently preparing for Bank of America for a long period. Losing this opportunity would have severely limited my chances of securing placements.

By the time we exited the TPO office, it was nearly 8:00 PM. We had missed lunch, and I was in no mood for dinner. I was simply lying on my bed and continuously checking my WhatsApp and mail. This continued until 2:00 PM, despite my exhaustion. Finally I was asleep, and the next morning at 5:45 AM, I heard a notification sound and checked, It was message from Swapnil Sir, that had list of selected students. Total 37 students were selected, 1 - MTech, 27 - BTech, 9 - MCA.

As I was scrolling the list, I was unable to locate any names from the MCA department, as the list was sorted alphabetically by department. Finally, I noticed my name with my blurred vision. I was overjoyed, as I had not been confident after the previous round. I checked again and there it was, my fruition after two years of hard work and dedication in this college.

Tips:

- Make sure that you are in well cleaned, ironed, formal attire. As the first impression does matter to the person who will be hiring a candidate for the bank.
- Make sure to you have your breakfast, as you most likely will be skipping lunch and will be waiting till the dusk. Additionally you can carry candy or bars, which will help you to fuel during the process. (Remember that every bit of effort you make will be matter for your selection)
- Thoroughly revise your resume, as it will be scrutinized for an extended period of two hours. Be authentic in all aspects of your resume, as any discrepancies may have severe repercussions.
- If you have any notable achievements, please include them in your resume. However, ensure that they do not occupy an excessive amount of attention on your resume. This gives you a little edge as it shows that you are quite an interesting person.
- Refrain from including extraneous information in your resume that is unrelated to the job role or the bank's requirements.
- Previously, Bank of America rarely asked design or architecture questions. However, for our batch, they asked design questions to all participants. Therefore, it is important to note that this interview experience may not be the same as yours. Each year, technologies and industries evolve to higher paradigms, which may result in a change in the level of scrutiny. Therefore, it is crucial to prepare thoroughly above the standard.
- Make sure that you have a comprehensive understanding of your project, as you may encounter numerous inquiries regarding it. The most frequently asked question is, what problem does your project address?

🍀 BEST OF LUCK 🍀