

Microsoft Interview Experience | On-campus for internship

- Last Updated : 22 Aug, 2019

So, recently, Microsoft had come to our campus for internships and out of around 150 people, 10 were selected. This is my experience of the whole interview process.

Round 1 (online coding round):

The coding round was hosted on mettl and there were three questions which we had to solve in 90 mins. The questions are selected from a huge database, so, everyone was allotted a different set. My questions were

1) Given a directory path in linux, find how far you are from the main directory after the complete path is traversed.

<https://www.geeksforgeeks.org/simplify-directory-path-unix-like/>

2) Find the longest palindromic substring of given string.

<https://www.geeksforgeeks.org/longest-palindrome-substring-set-1/>

3) There are n post offices in a row. You are given the distance of each post office from your home. A post office can send a post to another post office if the distance between the two places is less than or equal to x . You are given q queries. Each query contains indices of two post offices. You have to find whether the post offices can deliver post to each other or not. (Expected time complexity $O(n+q)$)

For example

Input : $n=3$, $post[n] = \{2, 3, 5\}$, $x=2$, $q=2$

Query 1: $\{1, 2\}$. Output: Yes (post office 1 and 2 can deliver post because the distance between them is less than 2).

Query 2: $\{1, 3\}$. Output: Yes (Even though 1 can't deliver directly to 3, 1 can deliver mail to 2 and 2 can deliver to 3. So, 1 can deliver post indirectly to 3).

The questions weren't tough but the online compiler was a bit different. For example, instead of strings and arrays as input in the function, we were given pointers as input. In the third question, the queries input was given as a double pointer. Some students were unfamiliar with dealing with pointers and so, weren't able to solve the problems completely.

After this round, 75 people were shortlisted for the group fly round.

Round 2 (Group fly round):

There were 2 questions which we had to solve in 60 minutes. We had to write the code on a paper and submit it.

Question 1:

Find the longest continuous subarray having at most k distinct elements.

<https://www.geeksforgeeks.org/longest-subarray-not-k-distinct-elements/>

Question 2:

Assume an array of n elements. We define a new sort wherein we divide the array into two equal parts and check whether either of the parts is sorted. We continue the process till we get a sorted subarray. Find the maximum length of the sorted subarray.

Example

Input: $n = 6$, $\text{arr}[] = \{7, 10, 9, 8, 11, 3\}$

Output: 2 {7, 10} or {8, 11}

Explanation: As the array is not sorted, we divide the array into {7, 10, 9} and {8, 11, 3}. We again divide the first array into {7, 10} and {9}. Thus, the maximum sorted subarray is {7, 10} whose length is 2.

After this round, 18 people were selected for interviews.

Round 3 (1st interview round):

Each interview round lasted for about 30-40 minutes. The interviewer first asked me about myself, then asked about my projects. Then, he asked me 2 questions. For both the questions, he asked me the time complexity and the space complexity and after he was satisfied with the approach, he asked me to write the code on paper.

1) Given an array, find the length of the longest sine subarray. (sine subarray means the elements should be in an increasing-decreasing-increasing or vice-versa manner).

My approach was to set a flag for increasing/decreasing and then, check whether the sine condition is satisfied or not. If it is, then increase the count and change the flag, else, set the count to 0 after updating maxCount. Time complexity $O(n)$, Space complexity $O(1)$

2) Find the number of connected components of a graph. (Standard BFS/DFS approach)

<https://www.geeksforgeeks.org/connected-components-in-an-undirected-graph/>

Round 4 (2nd interview round):

1) Given an array, select k elements from the array having the maximum sum. We discussed many approaches for this question having different time complexities. Some approaches were sorting ($O(n \log n)$), max heap ($O(n \log n)$), dp ($O(nk)$), recursion ($O(2^n)$), finding k maximum elements ($O(nk)$).

2) Given a pyramid of numbers, find the smallest path value to go from the root to the last level.

For example-

1

2 2 2 2 2

5 5 5 5 4 5 5 5 6

Smallest path $\rightarrow 7$ (1-2-4)

We discussed a dp approach for this.

$\text{dp}[i][j] = \max(\text{dp}[i-1][j], \text{dp}[i-1][j-1]) + \text{arr}[i][j]$ where i is the level index and j is the element index of

that level. (Remember to write the base cases and corner cases for all dp questions).

Round 5 (3rd interview round):

1) Given an array of integers, select two numbers having the least difference. I, first, gave him a sorting solution where you sort the array and find the least difference between adjacent elements. Then, he asked me to do it without sorting. I first gave him a brute force solution of $O(n^2)$ but when he asked me to further optimize it, i was unable to do so.

2) A shopkeeper sells items of Rs. 5. There is an infinite line of customers who want to buy the items. The customers have notes of either Rs. 5, Rs. 10 or Rs. 20. We have to find the conditions under which the shopkeeper will be unable to sell more items and find the profit he makes under each condition.

For example- customers[] = {5, 10, 10, 20}. For this case, after the 2nd customer, the shopkeeper will be unable to return change of Rs. 5 to the 3rd customer as he only has Rs. 10 note with him. So, the business stops and the profit is Rs. 10.

After this question, he asked me to generalize the problem to include Rs. 1, Rs. 2, Rs. 50, Rs. 100 etc notes too. He asked me how this system will be implemented in real life where there are so many types of denominations and so many types of products having various prices.

After this, he also asked me some basic OOP questions, like what are the 4 pillars of OOP and describe each of them.

Round 6 (HR round);

This round was the easiest of all and it lasted only about 10 mins for me. The interviewer asked me the same basic questions and why i wanted to join Microsoft. Then, he asked me two basic questions of DSA:

1) Given a sentence containing many words, reverse the words in the string.

<https://www.geeksforgeeks.org/reverse-words-in-a-given-string/>

2) Find the LCA of any two nodes in a BST.

<https://www.geeksforgeeks.org/lowest-common-ancestor-in-a-binary-search-tree/>

After each interview round, the interviewer would ask me whether i had any questions for him. I asked them which skills i needed to develop before coming if i get an internship.

The overall experience was a little tiring because the interviews which started at 9 in the morning went on till 9 in the night. But, it was a great experience as the interviewers were very friendly and gave me enough time to think about the solution. The important thing to remember is that even if you don't know the complete solution, you have to keep communicating to the interviewer because he generally helps you get on the right path and guides you towards the solution.

The results came in the night and i was one of the 10 selected people. Out of the 10 people, there were also some girls who had given the Codess program organized by Microsoft due to which they got direct entry to the interviews without giving the coding test or the group fly round.

I would suggest to practice from geeksforgeeks and interviewbit (only after covering the basics from geeksforgeeks) because they have a large set of questions which cover almost all the subtopics from all the topics. Also, before any interview, go through the internship experiences that people put

up here so that you get an idea as to what type of questions are asked in the interview by the company.

Also, even though they didn't ask me much from those topics, I would recommend revising the concepts of OOP and DBMS because a lot of companies ask questions from those topics in the interviews.

Thank you!

My Personal Notes

Add your personal notes here

Save