

Microsoft Interview experience | Set 92 (1.8 Yrs Experienced for IDC)

- Difficulty Level :[Hard](#)
- Last Updated :14 May, 2017

My resume was forwarded to Microsoft through a consultancy firm. I was called for a day long interview process at Microsoft IDC, Bangalore.

Round 1 (1 hour+):

Question 1: <https://www.geeksforgeeks.org/median-of-stream-of-integers-running-integers/>

I first gave the insertion sort algorithm, he told me to improve upon it. I was stuck for some time so he gave me another problem: <https://www.geeksforgeeks.org/k-largestor-smallest-elements-in-an-array/>. I was able to solve this in a few minutes using the Min Heap method, so he then told me to think on similar lines for the original question he gave me. After around 20-30 minutes of brainstorming I was able to solve it using the Heaps method.

Question 2: <https://www.geeksforgeeks.org/count-possible-paths-top-left-bottom-right-nxm-matrix/>

I solved this question pretty quickly. He asked me if I had come across this question before; didn't lie, told him I already knew this.

This was followed by discussion on what happens in the Bing Team at Microsoft.

Round 2 (1 hour+):

Question 1: Given a large text file, find an efficient algorithm to find the least distance(measured in number of words) between any two given words.

Solved this using hashing; he then extended the problem saying that this program would be executed a large number of times (to find distances between many pairs of words), so I need to improve my algorithm such that the time complexity is reduced after the first iteration of problem solving. Solved this using hashing and then using binary search to find the nearest index of the second word to the index of the first word in the hash.

Question 2: He first gave a background on how the autocorrect feature in phone keypads work, explaining that the probability of the autocorrected word would be greater than the typed word. He told me to find an algorithm to find the most probable word of length n that can occur given that there are m possible characters. The probability that a word can occur is given as: $P(\text{1st letter}) * P(\text{2nd letter given that 1st has been pressed already}) * P(\text{3rd letter given that 2nd has been pressed already}) * \dots$ and so on.. All values of individual terms in this equation is already given.

Think of the problem as a $m \times n$ grid where each element in a column is connected to every element in the next column by some weight. Solving this problem by brute force would be $O(m^n)$. By storing the solutions of subproblems the complexity would reduce to $O(m*n)$ which is the solution the interviewer wanted.

Round 3 (1 hr): Hiring manager round: Generic question but was a bit unexpected. Started off by asking about my current jobs and why I wanted to change, then the conversation went towards why I had chosen a non-CS course in my undergraduate, the conversation then took off on a completely different path about what computers can do and what computer science cannot do until it reached all the way up to The Turing machine. To be honest I was most nervous after this round.

If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

[All Practice Problems for Microsoft !](#)

My Personal Notes\ *narrow_drop_up*

Add your personal notes here

Save