

b'

## Microsoft Interview Experience (For Internship)

- Difficulty Level :[Hard](#)
- Last Updated :[01 Oct, 2018](#)

The complete process consisted of an online coding round, a group fly round and three technical interviews.

**Online Coding Round:** There were 3 \xe2\x80\x9cfunction problems\xe2\x80\x9d i.e. we only had to complete a function, carrying 2, 3, and 5 marks respectively and the total time allotted was 75 minutes. It was conducted on the Cocubes platform and though I did not face any problem with libraries in Java, some of my batch mates using C++ did. On submission, the results of some trivial test cases were shown.

**2 marks:** Odd numbered cars can travel on an odd date and even numbered cars on even dates. Other cars would be fined 250 each. Given the numbers of the cars(an array of integer) traveling on a date(an integer), calculate the total fine collected.

Examples:

```
\r\nInput : cars = [1, 2, 3], date = 15\r\nOutput : 250(for car 2)\r\nInput: cars = [3, 4, 5], date = 16\r\nOutput : 50
```

**3 marks:** [Next smallest palindrome](#)

**5 marks:** [Sum of cousins in a binary tree](#)

**Group fly round:** There were 7 short answer type question on C and DSA and a coding problem [Detect and remove loop in a linked list](#)

**Interviews:** These were 3 personal technical rounds.

**First round:** It was a [linked list question](#). I had to code the solution on a paper and dry run it while explaining it to the interviewer.

**Second round:** I had done a course on Machine Learning so I was asked some questions on gradient descent and classification problems. I was asked two problems. The first one was to find the square root of a number accurate to 10 decimal places. I gave a binary search solution which I had to code.

The second question was to sort a list of numbers if the k least significant digits are not important.

Examples:

```
\r\nInput : list = [674, 689, 349, 327, 989], k = 2\r\nOutput : [349, 327, 674, 689, 989] is valid and so is [327, 349,
```

My first approach was radix sort but eventually came up with a solution based on bucket sort. The idea is to map every element to \xe2\x80\x9cbuckets\xe2\x80\x99 by dividing it by  $10^k$  e.g. both 674 and 689 will get mapped to the bucket 6, 349 and 327 to \xe2\x80\x93\xe2\x80\x93 and so on and then sort the buckets. I did not have to write the code.

**Third round:** We had a discussion on my project on key frame extraction. He asked me questions on the algorithms and how I could improve it.

The question asked was open ended and the solution was reached through discussion.

There is a numeric keypad. The key 1 maps to \xe2\x80\x93\xe2\x80\x93, \xe2\x80\x93\xe2\x80\x93 and \xe2\x80\x93\xe2\x80\x93, 2 to \xe2\x80\x93\xe2\x80\x93, \xe2\x80\x93\xe2\x80\x93, \xe2\x80\x93\xe2\x80\x93 and so on. So the sequence of key press 12 maps to all possible combinations i.e. \xe2\x80\x93\xe2\x80\x93, \xe2\x80\x93\xe2\x80\x93, \xe2\x80\x93\xe2\x80\x93, \xe2\x80\x93\xe2\x80\x93 and so on. There is also a list of strings and given a sequence of key press display the strings having a mapping as prefix. This is similar to how a search works in the dialer of a phone.

```
\r\nlist = [abc, adg]\r\n\r\nsequence = 1\r\noutput = [abc, adg]\r\n\r\nsequence = 12\r\noutput = [adg]\r\n\r\nsequence
```

Memory is not a constraint but time is critical.

My initial approach was to store the list of strings in a prefix tree followed by BFS according to the sequence of key press. But the complexity is exponential to the number of key presses in the sequence so I had to find a different solution which was based on hashing.

This was not so much of a question as it was a discussion of various approaches and the most important aspect was to convey the thought and the rationale behind it.

I had to code up a part of the solution.

Overall, the interviewers were helpful and it was encouraged to asked questions.

Keep hustlin\xe2\x80\x93,  
orpidus

My Personal Notes[\narrow\\_drop\\_up](#)

Add your personal notes her

Save

,