b'

# Amazon Interview Experience | Set 350 (For SDE I)

- Difficulty Level :\nHard
- Last Updated :\n09 Jul, 2019

**Round 1 ( Coding Round + MCQs ):**
Platform : Hackerearth

1. Maximum 0 flipping subarray
2. Find a tour that visits all stations

   The test cases were very weak and any brute force solution got accepted.

20 MCQs based on OS, DBMS. The questions were very similar to those present on Geeksquiz.

**Round 2 ( Telephonic Interview 1, duration \xe2\x80\x93 1 hour) :**

1. Find smallest range containing elements from k lists/
2. Delete operation on a BST.

**Round 3 ( Telephonic Interview 2, duration \xe2\x80\x93 35 mins ) :**

1. Adding two polynomials using linked list
2. Inorder successor in Binary search tree

After this round I was invited for onsite interviews at Amazon Bangalore Office.

**Round 4 : ( F2F , duration \xe2\x80\x93 45 mins )**

1. Relative Sorting
   This question can be easily solved using a custom compare function combined with merge sort, I figured this out later on . I had proposed a different solution based on 2 hashmaps. Time complexity was O( n log n ). My approach was a little complex and lengthy, so the interviewer did not ask me to code this.
2. Reverse a linked list.

He asked me to dry run my code and checked all corner cases.

**Round 5 : ( F2F, duration \xe2\x80\x93 1 hour, 2 interviewers )**

1. Design LRU Cache.
   I designed it using unordered_map and doubly linked list. I coded my own DLL class. After that they framed different test cases and asked me to explain how my code was handling those cases.
   They asked some questions about how hashmap is implemented.
2. You are given a string and 2 operators ( & , | ). Return the total number of ways in which you can evaluate to true using the given string and operator set.

   ```
   \r\nExample  : Input : TF\r\n                  Output : 1\r\n                  Input : TFF\r\n                  Outp
   ```

   I solved it using parenthesization and memoization. Time Complexity \xe2\x80\x93 O ( n3 ).

   I used a 3-D array to memoize.

   dp[i][j][0] : Number of ways to form 0 using substring( i, j )
   dp[i][j][1] : Number of ways to form 1 using substring( i, j )

   Interviewer stressed to use only a single 2-D array so I had to include the following precomputation to calculate the total number of ways to parenthesize a string of length n, lets say F( n ).

   F( n ) = 1 + summation ( F( i ) * F ( n \xe2\x80\x93 i ) ) i varies from 1 to n-1 .

**Round 6 ( F2F, duration \xe2\x80\x93 45 mins )**
Interviewer warned me to inform him in case I had already encountered any of these questions.

1. Minimum steps to reach a destination. I told him that I had already solved this question using BFS. He asked me to explain the approach, but did not ask to code.
2. Minimum number of jumps to reach the end of an array

   He was expecting an O ( n ) solution. I solved it using 2 pointer approach. He asked me to dry run my code and handle cases when array values are negative or zero.

3. Construct a Maximum Sum Linked List out of two Sorted Linked Lists having some Common nodes

I told him that this is a classical problem and I have solved it earlier. Told him the approach and handled all corner cases.

**Round 7 ( F2F, 2 interviewers, duration \xe2\x80\x93 30 mins ):**

Detailed discussion about my project. Some random questions about the technologies I had used.

1. Remove all nodes which don\xe2\x80\x99t lie in any path with sum>= k

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 Amazon** !

My Personal Notes\n*arrow_drop_up*

Add your personal notes her

Save

'