

b'

Amazon Interview Experience SDE-1 | Feb 2020 (Exp 1.5 yr)

- Difficulty Level : [Easy](#)
- Last Updated : 13 Apr, 2020

Hey guys !

I am sharing you my interview experience with Amazon for SDE-1 role in Feb 2020.

Round 1:

Well first round was an online assessment test on AMCAT. There were two coding questions need to be completed in 90 Min.

1. Given a 2-D matrix of 0's and 1's, where 1 represents an infected person and 0 represents an uninfected person. After each second an infected person infects his 4 uninfected neighbors(L, R, U, D). Need to calculate time such that all becomes infected.
Approach : Apply BFS
Similar to : [Rotten Oranges](#)
2. Given a 2-D matrix of 0's and 1's, where 1 represents a building component and 0 represents an empty area. We need to calculate total number of connected building components.
Approach : Apply DFS
Similar to : [Number of Islands](#)

Next three rounds were onsite at Gurgaon location.

Round 2:

This round started with brief introduction about me and my work at previous company followed by two coding questions :

1. Given a string say ABAABCD. Calculate minimum number of letters to be removed such that remaining letters can form a palindrome string.
Answer for ABAABCD is : 2
Explanation : Remove C and D, remaining string is : ABAAB which can form a palindrome(BAAAB)
Approach : Simply count the number of odd characters. Since you can keep one character of odd count hence answer will be odd character - 1.
Used HashMap for storing characters and their count.
if(odd_characters==0) return 0;
return odd_characters-1;
2. Given two Linkedlists in sorted increasing order. Merge them in decreasing order. You have to merge in place, you can't create new linkedlist.
Approach : Simply apply merge-sort concept and append characters at front of merged list instead of end.
Merge in increasing order : [Merge Two Sorted LinkedList](#)

Round 3:

This round also started with brief introduction about me and my work at previous company followed by two coding questions :

1. Given an sorted array of 0's and 1's in non-decreasing order. Find the sum of array in O(log n)

Approach : Apply Binary Search to find the position of first 1 and return n-position+1.

2. Given an array of Integers, find and replace next smaller element of each element in the given array in $O(n)$.

Approach : Use Stack.

Steps :

Insert element starting from rear.

If top of the stack is greater than current element keep removing until smaller element is found or stack becomes empty.

if Stack \neq Empty answer for current element = -1

else answer for current element = stack.peek();

Insert current element to stack and repeat above 3 lines. Further he asked me some OS concepts and some previous work related questions.

Round 4:

This round also started with brief introduction about me and my work at previous company followed by two coding questions. He also asked me about my current project, asked deeply about the approach I used to solve the problems I faced.

1. Given a number n which represent total stairs. Find in how many ways you can reach the nth stair with 1 or 2 steps at a time.

Approach : DP

Since to reach 0th, number of ways = 1;

to reach 1st, number of ways = 1 (0→1);

to reach 2nd, number of ways = 2 (0→1→2 | 0→2)

to reach 3rd, number of ways = 3

Hence we can see, to reach nth stair, number of ways = ways to reach (n-1)th + ways to reach (n-2)th.

Link : [Nth Stair](#) Furthermore he asked me expand this problem with k-steps (k steps at max which can be taken) :

It's simple again just need to add previous k values to get ways to reach at particular stair.

2. Given a binary search tree(BST), find top view of given BST. Approach : Maintain both horizontal distance and as well as level for each node.

Create a `TreeMap<Integer, Pair>` which stores horizontal distance and a pair class object.

(Note : `TreeMap` is already sorted based on key i.e. horizontal distance)

Apply Inorder traversal as per following : `void topView(Node root, int hd, int level)`

```
{
    if(root==null)
        return;
    if(map.containsKey(hd))
    {
        if(map.get(hd).level<level)
            map.put(hd, new Pair(root.data, level));
    }
    else
    {
        map.add(hd, new Pair(root.data, level));
    }
    topView(root.left, hd-1, level+1);
    topView(root.right, hd+1, level+1);
} Simply print Pair.value for each key in TreeMap.class Pair{
    int value;
    int level;
    Pair(int a, int b)
```

```
{  
value=a;  
level=b;  
}  
}
```

Round 5:

This was final round of an hour which was held on Chime Video Call :

1. \xc2\xa0Asked me hardest thing which I have done in last 1.5 year and my previous work related questions.
2. A number is represented by a LinkedList, where each node represent a digit of number. You are given two such number, find sum of two numbers. You need to return head of LinkedList representing sum of two given numbers.
Approach : First I answered reversing both and then adding each digit and forwarding the carry. And finally reversing the final LinkedList.
Second I gave him another approach using Stack.
Finally I gave recursive approach and he asked me to solve using the same.
Link : \xc2\xa0\xc2\xa0[Sum Of Two LinkedList](#)

Finally I got a call from HR that I got selected for SDE-1 role.

Kindly keep below points in mind which I \xe2\x80\x99ve used during my interview :

- A. Always answer if you know else simply say \xe2\x80\x9cI don\xe2\x80\x99t know Sir/Ma\xe2\x80\x99am\xe2\x80\x9d.
- B. For coding questions, start with brute force approach to optimized one.
- C. While writing code make sure to cover all the edge cases by yourself only. You may dry-run your code before finally saying you\xe2\x80\x99re done.
- D. Should be able to calculate Time and Space complexity.

\xc2\xa0

Practice more for all Data Structure and Algorithm concepts on GeeksForGeeks, you will find all the concepts at this place.

Artical submitted by : Ripes h Yadav

My Personal Notes\narrow_drop_up

Add your personal notes here

Save