

Amazon Interview | Set 109 (On-Campus)

- Last Updated : 29 Jul, 2019

Amazon visited my Campus a few days back. I did not clear the final round of Amazon but I got selected in D.E. Shaw the very next day after rejection from Amazon. Here is my interview experience with Amazon.

Online Round (conducted on HackerRank):

20 MCQ questions and 2 Coding Questions (no C++ questions, C input-output questions(very hard) and aptitude questions).

Coding Questions :

1. [Merge overlapping intervals](#)(Very well known repeating problem).
2. [Given a singly linked list you have to subtract the value of first node from the last node and so on until you reach the middle node.](#)

Eg: Input: 5 -> 4 -> 3 -> 2 -> 1

Output: 4 -> 2 -> 3 -> 2 -> 1

Advice: Many people say that making 1 code is enough to clear the 1st round while this is not true, I made both the codes and many of my friends who made only one code were rejected. Everyone who made both the codes was selected. There were some students who cleared this round by making only one code but don't try your luck here by making only one code is my suggestion.

Round 1(Bar Raiser):

Complete discussion on my project.

Suppose you have a file with billion entries and you have to sort the data of a file according to a column and can put only that column in memory and after putting that your memory gets full.

e.g. your file contains

name age address Pincode sex

name age address Pincode sex

And you can have to sort the file according to Pincode then you can only put Pincode in the memory.

I approached it firstly by using merge sort but this requires the entire file to be put into memory. Then, I suggested using min heap but he said that the heapify process of billion numbers will take a lot of time. He said you are getting close, then I suggested that we can use a balanced binary search tree to store the Pincodes along with indexes and then traverse the BST in inorder fashion and swap the indices of the first Pincode of file with the index of the inorder element. He was happy after hearing this and asked me to write the code and to be careful with edge cases.

I did it.

Round 2 (CS Fundamentals)

He wanted to check my knowledge of DBMS, OS, and Computer Networks. Asked everything related to these subject including all the layers of networks and from process-threads to deadlocks to memory management in OS. Asked a SQL query to find 3rd maximum marks of the student from a database. I gave him 2 solutions.

Asked me my favorite sorting algorithm. My favorite is insertion sort but I told him merge sort because I knew a lot about merge sort so I wanted to drive the interview towards merge sort and as expected he asked many questions on merge sort and I gave him all the answers.

He asked me to build a tree from given preorder and postorder traversal of the tree, I said it is impossible to build from only these 2 traversals, you have to give me in order to build a unique tree.

Round 3 (CS Fundamentals + Coding)

Deep discussion about project.

Asked me to choose my favorite subject other than any related to coding or algorithms. I said Computer Networks. He asked why it is your favorite? Why not OS or DBMS?

He asked me What happens when you type www.amazon.in in your browser. I was expecting this question :P. I knew this, told him everything.

See the links below for solution

[What really happens when you navigate to a URL](#)

[What exactly happens when you browse a website in your browser?](#)

[What happens when you type a URL in browser](#)

He asked me to solve a coding problem now.

Suppose I am given a sorted array of unknown size, then how will I search an element in this array efficiently.

I said I can increase i by the power of 2 and check whether the given $arr[i] > element$ to search whenever I find such element then I can apply binary search between $i/2$ and i . He said this approach will work but what happens if while increasing i , you cross the array length. Suppose an array is of size 6, then your i will become 8 and $arr[8]$ will give garbage value so your program may go in an infinite loop. I said I can increase i in the try block and whenever this case

occur then I can catch the exception in catch block where I will decrease i by 1.
He smiled and asked me to write the code by taking care of edge cases.

Round 4 (Coding)

Deep discussion about the project which was on security. The interviewer asked me to design the database of a chat application similar to WhatsApp. I did it and said what if some intruder access the database, then I said I will apply md5 encryption on phone numbers, then he asked me how md5 works. Why only md5? He asked me what are Man in the middle attacks and asked me to remove the man in middle attacks from the message transfers. He was impressed.

\xf0\x9f\x99\x82

He then asked me what is cache, why are cache fast and how it differs from RAM and hard disk architecturally. This one was a bouncer, I hated computer architecture, admitted I am a beginner in computer architecture. He laughed and said thank god, there is one thing in which you are a beginner. I laughed too \xf0\x9f\x98\x80

He moved further towards coding.

1. Suppose the structure of a binary tree is

```
\r\n    struct node {    int val;    \r\n        struct node *left,*right,*random;\r\n    }node;
```

[Where random pointer points to any random node of the binary tree and can even point to NULL, then how will you clone this tree.](#)

This was the culprit question who led to my rejection.

I gave a lot of thought to it but wasn't able to solve it directly. I said we can use hash, then he said what if there are duplicates in the tree then your hash will fail, then I said we can convert the binary tree to doubly link list in place then clone the doubly link list, he said ok but how will you generate the original tree from doubly link list, I never said it was balanced. I gave a lot of thought here, but yeah couldn't solve it.

I got the solution by trying it after the interview, you can clone the left and right pointers easily by traversing the tree and to clone the random pointers, you can apply this: newroot->random=oldroot->random; oldroot->random=newroot; Then traverse new tree again and put newroot->random=newroot->random->random.

2. [Find the row with maximum number of 1s](#)

I solved it efficiently.

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.

[All Practice Problems for Amazon !](#)

My Personal Notes

narrow_drop_up
Add your personal notes here

Save