

Amazon Interview Experience for SDE-1 (Full Time-Referral) 2020

- Difficulty Level : [Expert](#)
- Last Updated : 28 Sep, 2020

Hi geeks! wish to share my interview experience with you all.

During this pandemic time, Amazon interviews were all virtual through Chime (Video Conferencing & Online Meetings App)

Round 1: (Online Assessment Round)

[Q1\) Maximized partitions of a string such that each character of the string appears in one substring](#)

[Q2\) Count of substrings of length K with exactly K distinct characters](#)

I have linked a similar problem in geeksforgeeks.

It was nearly 3 weeks gap before I got a call from Amazon regarding the interview process after this Online round.

So be patient and keep preparing.

Round 2: (Technical Interview Round-1)

It is a coding round consisting of two coding questions. I was asked to discuss my approach and analyse the time complexity before getting into the actual coding.

[Q1\) Find the smallest window in a string containing all characters of another string](#)

I was able to come up with $O(S + T)$ solution where S and T are lengths of the strings

[Q2\) Remove minimum number of parentheses to make the input string valid](#)

Round 3: (Technical Interview Round-2)

[Q1\) Find k closest elements to a given value given array is not sorted](#)

I thought of a binary search solution after sorting but was asked to use an apt data structure.

Later I explained my solution using Heap.

[Q2\) Given a input string 2\(a3\(bc\)\) output should be abcbcbcabcbcbc](#)

Solution was using stack

Round 4: (Hiring manager Round)

Lots of questions on my project with follow up questions as well.

I was asked to design classes for a deck of card with functionalities like shuffle and hand-over.

Discussed about Serializing and deserializing the binary tree and many more

Round 5: \xc2\xa0(Bar Raiser Round)

Q1) Bubble shooter : Given a matrix of colors

[[\xe2\x80\x98r\xe2\x80\x99, \xe2\x80\x99g\xe2\x80\x99, \xe2\x80\x99b\xe2\x80\x99],
[\xe2\x80\x98o\xe2\x80\x99, \xe2\x80\x99g\xe2\x80\x99, \xe2\x80\x99g\xe2\x80\x99]] and a target array [\xe2\x80\x98g\xe2\x80\x99, \xe2\x80\x99r\xe2\x80\x99]

take each bubble from target array and try to blast the same color bubble in the matrix from the bottom view.

if able to hit a bubble, then that bubble in turn hits the neighbor bubble (4-direction) of same color.

Bursting a bubble makes the spot transparent and could allow to view the interior bubble from bottom view.

If one target color couldn't hit the matrix then the game ends \xe2\x93 should return the count of the burst bubbles .

I was asked for an optimized solution \xe2\x93 mainly for checking the bottom view

I used a hash to keep track of active rows for each column. whenever a bubble is burst active row is the row below it.

so bottom view for each column is matrix[active_row[col]][col]

Q2) Immediate next larger element in an array.

I was asked for O(N) solution \xe2\x93 Used monotonous stack for my approach

There was a follow up \xe2\x93 to find Immediate left larger element without extra traversal or extra space

I was able to answer that too

After these there were plenty of behavioral and project related questions. Some of them are \xc2\xa0

1) what is the challenging project that you worked on? \xe2\x93 asked to explain in detail about the implementation, schema discussion etc.

2) Total number of the projects and how big they were.

3) Any customer tickets that you worked on.

4) Asked about [KMP algorithm](#) and [Segment tree](#).

5) Explain about a time when you had a tight deadline.

6) Explain about a time when you came up with any ideas or automation.

Result: SELECTED

Before Interview :-\xc2\xa0

\xc2\xa0I would like to share about my preparation process. Choose any programming language of

your like and keep solving plenty of problems in geeksforgeeks or similar sites to get the idea of problem solving. While solving problems my pattern is to cover the data structures one by one, understanding it's best/average/worst case time complexity. When you are using a STL in your code make sure you are clear about it's time complexity. I used to spend some time in the discussion board and solution panel after submitting my approach to understand various thought process of other geeks and learnt a lot.

During Interview :-

The Interviewer expects the most optimized solution with proper usage of suitable data structures and algorithms for every question with full implementation. Interviewers will help you whenever you are stuck just keep thinking loud. Before jumping into solution, understand the question properly and avoid any assumptions regarding the problem statement get them all clear by asking the interviewer as many question as possible. Think of all possible edge/corner cases. Discuss your solution with the interviewer and explain the cases that you are handling. It would be nice to make a rough run through your approach with an example input.

All The Best !

My Personal Notes

Add your personal notes here

Save