b'

Amazon Interview Experience for SDE-1 (1 year Experienced)

- Difficulty Level :\n[Easy](#)
- Last Updated :\n24 Nov, 2020

I got this opportunity during the COVID time and hence all the interview rounds were conducted on Amazon Chime (video call) and I had to write the code on LiveCode (a shared IDE) which was visible to both me and the interviewer.

First-round (Online Test)

- There is a matrix containing 0,1 and 9. You need to start from the top left and reach the cell containing \xe2\x80\x989\xe2\x80\x99. There is only one such cell containing 9. \xc2\xa0You can move on to the cell containing \xe2\x80\x980\xe2\x80\x99. A cell containing \xe2\x80\x981\xe2\x80\x99 is an obstacle and you have one jump which you can use to cross the obstacle. We need to find the minimum number of steps required to reach the cell containing \xe2\x80\x989\xe2\x80\x99 from the top left of the matrix. (I applied BFS and it worked for me)
- There is a list of Amazon orders, with each of them being represented by a string. Each of the strings was of the format \xe2\x80\x98XXX YYY ZZZ\xe2\x80\x99. We need to sort these based on a few priority conditions.
  Solution: I used the sort function of C++ STL and wrote a custom comparator function)

Second Round:

- Given an input string and q number of operations that need to be performed on it. There are three types of operations.
  1 \xe2\x80\x93 reverse the string
  2 1 a \xe2\x80\x93 insert the input character (here \xe2\x80\x98a\xe2\x80\x99) in the front of the current string
  2 2 a \xe2\x80\x93 insert the input character (here \xe2\x80\x98a\xe2\x80\x99) in the back of the current string

  Solution: Firstly I gave the approach using a doubly-linked list, but he wanted to use the same string variable to perform operations instead of making a DLL. As the insertion in the front on the string is a costly operation, so we can keep the front insertions in a vector and when all the operations are completed we can then make a string out of those characters and add a prefix to the current string in which all the reverse operations and insertions at the back has been performed.

  Also we need to maintain a \xe2\x80\x98flag\xe2\x80\x99 which is toggled based on the reverse operation (as soon as there is an operation of type \xe2\x80\x981\xe2\x80\x99 (reverse), toggle the value of flag)

  * Case 1: flag == false
  \xc2\xa0 \xc2\xa0 \xc2\xa0\xe2\x80\x93 front operation: needs to be added to the front_vector
  \xc2\xa0 \xc2\xa0 \xc2\xa0\xe2\x80\x93 back insertions: insert the character at the back of the original string (concatenation)

  * Case 2: flag == true
  \xc2\xa0 \xc2\xa0 \xc2\xa0\xe2\x80\x93 front operation: insert the character at the back of the original string (concatenation)
  \xc2\xa0 \xc2\xa0 \xc2\xa0\xe2\x80\x93 back insertions: needs to be added to the front_vector

Also, before generating the final string, you need to check if the flag is true means the string is reversed, so based on that you need to generate the output string (I missed this case, when the interviewer pointed out I incorporated this).

Dry Run Example:

abc -> input string(lets say \xe2\x80\x98s\xe2\x80\x99)
5 -> number of operations \xc2\xa0 \xc2\xa0

1 \xc2\xa0 \xc2\xa0 -> \xc2\xa0 \xc2\xa0s = abc (front_vector {}, flag= true) \xc2\xa0
2 2 a \xc2\xa0 \xc2\xa0 -> \xc2\xa0 \xc2\xa0s = abc (front_vector {a}, flag= true)
2 1 b \xc2\xa0 \xc2\xa0 -> \xc2\xa0 \xc2\xa0s = abcb (front_vector {a}, flag= true)
1 \xc2\xa0 \xc2\xa0 -> \xc2\xa0 \xc2\xa0s = abcb (front_vector {a}, flag= false)
2 1 x \xc2\xa0 \xc2\xa0 -> \xc2\xa0 \xc2\xa0s = abcb (front_vector {x,a}, flag= false)

Front insertion string: xa
Output (Final string): xaabcb
\xc2\xa0

- https://www.geeksforgeeks.org/find-the-farthest-smaller-number-in-the-right-side/

This round went for around 1.25 hours and I solved both the questions.

Third Round:

- Merge two sorted linked lists (https://www.geeksforgeeks.org/merge-two-sorted-linked-lists/)
- Trapping rainwater problem (https://www.geeksforgeeks.org/trapping-rain-water/)

This round went for around 1 hour and I solved both the questions.

Fourth Round (Hiring Manager):

- Started with an introduction and then moved on to a detailed discussion about the project with my current employer which included
  - What was my role?
  - What components did I work upon?
  - How were the components tested?
  - What was the size of the user base?
  - How were the bugs reported and handled
- Behavioral Questions:
  - Tell me a situation where you worked on a tight deadline
  - Tell me a situation where you took a decision
  - Tell me a situation where you worked for a client requirement and delivered it within the deadline.

This round went for around 1.25 hours and for the behavioral questions you need to follow the STAR (S \xe2\x80\x93 situation, T \xe2\x80\x93 task, A \xe2\x80\x93 action, R \xe2\x80\x93 result) method.

Fifth Round (Senior Dev Manager):

- Started with an introduction, a brief discussion about the project with the current employer
- Anagram (https://www.geeksforgeeks.org/check-whether-two-strings-are-anagram-of-each-other/). \xc2\xa0I solved the question quickly but there were few exceptional \xc2\xa0cases which can be handled and this will reduce the complexity for them. As the interviewer pointed out the cases, I incorporated the changes in the code, and he was satisfied.
- Behavioral Questions:

- Tell me a situation where you worked on a tight deadline
- Tell me a situation where you went out of your comfort zone to learn and delivered something.

This round went for around 45 minutes.

Verdict: \xc2\xa0 Selected \xf0\x9f\x99\x82

Tips:

- For coding interviews, you will first need to explain your approach verbally and then if the interviewer is satisfied with the runtime complexity, he will ask you to write the code with proper syntax. You will not have to take input, just take the input as function parameters and write the complete logic in the function). Practice on explaining your approach.
- For behavioral questions, find the set of questions https://leetcode.com/discuss/interview-question/437631/Amazon-Leadership-Principle-(Behavioral)-Questions. Here is a nice simple blog explaining the process of behavioral interviews at amazon (https://interviewgenie.com/blog-1/interviewing-at-amazon-behavioral-interview-questions)
  - Make a list of tasks that you did
  - For every question try to fit those tasks into the situation
  - For every question be ready with at least two tasks.

My Personal Notes\n*arrow_drop_up*

Add your personal notes here

Save

'