b'

Amazon Interview Experience for SDE

- Difficulty Level :\nMedium
- Last Updated :\n20 Aug, 2021

Round 1(Online Assessment): This round usually consists of two coding questions, for which you have to write the code as well as give a proper explanation in a separate window. The time for this round is approx 1 hr\xc2\xa0 **\xc2\xa0**

Questions asked were:

1. Sort a list of orders of prime and non-prime orders (https://leetcode.com/discuss/interview-question/1261316/amazon-oa-sde-1-new-grad-2021-batch-india). I did this by simply passing my custom comparator to the standard sort function. It is easy but it has some edge cases, that took a bit of my time.
2. Optimize Memory usage (https://leetcode.com/discuss/interview-question/373202)

After this, you will have to do some behavioural MCQ type questions.

In all the upcoming rounds I was asked both behavioural and coding questions therefore I will try to include both of them.

Round 2: This round started with my introduction and current project. After this interviewer inquired about a time when I have to do some unexpected complex work and how I handled the situation.

Coding questions:\xc2\xa0

1. Given a binary tree calculate the sum of all the left leaves:\xc2\xa0
2. Implement LRU Cache. The interviewer purposely left this question vague by just quoting \xe2\x80\x9cImplement LRU cache\xe2\x80\x9d, therefore I asked many clarifying questions and slowly came to an optimal solution using a doubly-linked list and hash maps.

Round 3: This round also started with my introduction and current work.\xc2\xa0

Coding questions:

1. Evaluate an arithmetic expression that is given as a string, the expression will only contain numbers and the four operations +, -, *, /

   ```
   Input: "3*2+5"\r\nOutput: 11
   ```

   I decided to solve this question using two stacks, and storing all arithmetic operators and numbers in different stacks with rules:

   \xe2\x80\x9cno low precedence operator can be pushed on a high precedence operator\xe2\x80\x9d

   \xe2\x80\x9cWhen an operator is popped out then two numbers are also popped out from the other stack\xe2\x80\x9d

   so my algorithm was basically this:

   - initiate two stacks as **operatorStack** and **operandStack**
   - scan through the expression
   - if a number is found push it into operandStack
   - if an operator is found and the top of operatorStack has low precedence than this current operator, push in into operatorStack
   - if an operator is found and the top of operatorStack has high precedence than this current operator, then keep on popping out operators from the operatorStack until the top has low or equal precedence than the current operator. With each operator popped, you have to pop two numbers from the operandStack and apply the popped operator on these two numbers and push the result back onto the operandStack, something like this:

     ```
     op = operatorStack.pop()\r\nnumber2 = operandStack.pop()\r\nnumber1 = operandStack.pop()\r\noperandStack.push(nu
     ```

   notice here the second number is popped first.

   - after the scan of expression, pop the remaining operators from operatorStack with the same above rules
   - Your final answer will be the top of the operandStack.

   There is one more simple approach, convert the infix expression to postfix and evaluate it simply with one stack.

2. Calculate the minimum number of jumps required to reach the end of an array

   I gave two approaches for this question, one O(n^2) and the other O(n)
   Since there was very little time left (around 15mins), I had some difficulty in explaining the O(n) approach, but the interviewer was convinced and asked me to code that out.

After this some computer science-related questions were asked like:

1. What type of databases you have used
2. What is the difference between SQL and NoSql Databases and for which type of query both are optimized?
3. Difference between thread and process
4. Why is communication faster in the thread as compared to process (because thread share same memory space)
5. What are locks and semaphores

Round 4: In this round for approx 30 mins the interview asked behavioural questions and he really wanted detailed and deep answers for all of them. I am quoting the ones which I can remember:

- A complex feature you have worked upon, what challenges you faced, how did you handle each of those
- A time when you faced close and stringent deadlines, how did you meet them
- How did you learn the technology you were working on, what steps did you take?
- How did you improve on yourself? etc
- In the next 30 mins he asked this coding question: Find a pair of numbers from an array whose sum is closest to zero. For this, I used sorting and two pointers approach

Round 5: Again in this round approx 30 mins were consumed by the behavioural questions, and similar to the previous one, he also wanted detailed and deep answers

Coding questions:

1. Intersection of two linked lists: I gave two approaches for this: first one: using two for loops and the second one: using hashmap
2. Symmetic Binary Tree I solved it by doing a parallel dfs for both the child of the root and comparing the node at each level.

Some Tips:\xc2\xa0

- Don\xe2\x80\x99t try to take too much time in explaining the first question if it was easy because usually the interview also wants to ask a second question and hence you will be left with very little time to explain and code the second question.
- Be honest in all the behavioural questions because they ask many counter questions on your answer, and if you have made it up, it will surely fall apart at some time.\xc2\xa0

My Personal Notes\n*arrow_drop_up*

Add your personal notes her

Save

'