b'

Amazon Interview Experience | Set 414 (For SDET-1)

- Difficulty Level :\nHard
- Last Updated :\n12 Jul, 2019

I applied through an employee referral for SDET-1 position. I was interviewed at Amazon Chennai(SP Infocity).I faced 5 face to face rounds.

On October 11th, I faced the first round which was mainly related to problem solving and coding.

Round 1:Problem Solving and Coding Round(1 hr)

The interviewer just started with a formal question of introduce yourself.Then she asked me about my final year project which I clearly explained and sketched it on the board.

Then she asked me two coding questions.

1. Simulate an android pattern lock.
2. Generate all words that can be formed with the given pattern. The keypad specification is the mobile keyboard.I will explain it clearly

If the given input is 2 3 6.

2 corresponds to abc. 3 corresponds to def. 6 corresponds to mno.

Suppose if we have to generate all combinations of 2,3,6 the output should be a set of strings

adm,adn,ad0,bdm,bdn,bd0,\xe2\x80\xa6..

I gave a recursive and an iterative solution to print the combinations.

Then the interviewer modified the question slightly.She asked me to validate the generated strings with a dictionary.

I gave two approaches for maintaining the dictionary. First I suggested a hash table approach which takes O(1) average time complexity for insert delete and search operations.

Then since the generated strings have common prefixes I suggested a trie approach for the dictionary.

The interviewer was pleased with my solution and asked me to code the solution completely.

After a few days I had 2 face to face rounds.

Round 2:Testing and Automation round(1 hr)

The interviewer just started out by asking whether I had previous knowledge about testing and testing environments to which I replied no.Then we discussed about the types of testing and the different phases and the testing procedures that are associated with it. Then she asked me two simple test case generation questions.

1. Generate all possible test cases for an add function which takes two strings as input and returns an integer which is the sum\xc2\xa0 of the two numbers that are given as string input.

I gave around 10 test cases for the question as it was pretty much straight forward. And I coded a few by using JUnit testing which is really simple.

2. Given a open file module in an editor what are the possible scenarios in which the open file module might give errors.

I gave a list of cases which we clearly discussed about the ins and outs of each one.

This round was quite easy since I prepared the testing part the previous day.

Round 3:Data Structures and Algorithms Round(1hr 15 mins)

Two guys interviewed me this round. We just started out with a formal introduction.Then they asked me two questions.

1. Knight walk problem.

Given a n*n chessboard and a knight which is placed at any one corner, Generate all possible paths following which the knight can cover all the squares.

I started with a approach using BFS. But there was a flaw in my approach which was that the knight would not escape from a cell if all the possible points the knight can go from that given point was explored previously.

Then I gave a solution using Backtracking.

Start with a corner.For each and every possible point the knight can reach make recursive calls to those using a simple for loop.Once the count reaches n*n(Example in a 8*8 chessboard,if the count is 64),Print out the traversed path that can be easily maintained in an array. This solves the flaw in my previous approach as we are able to go back to the other possibilty if a dead path is reached.

The interviewer was happy with my approach and asked me to code the solution. I coded it and then we moved on to the next question.

2. Given a linked list find the intersection point of two linked lists which converge at a common point.

I gave a brute force O(n^2) approach at first then optimised it to O(n) time and O(n) Space approach.Then I finally gave a O(n) time O(1) space solution by finding the absolute difference of the lengths of the two lists.Then they asked me to code the solution which was pretty much straight forward.

After this I had the final two rounds on October 31st.

Round 4:Hiring Manager Round(1hr)

The interviewer asked me to brief about myself.We then settled down on my project for the first few minutes and then the interviewer asked about the interviews that I had attended earlier and how the process went through. Then he started asking about testing and the use of testing corresponding to some given scenarios. After a few minutes of discussion he asked two questions regarding data structures.

1. Given an array of integers print the pairs that add upto a given sum.

I gave a O(nlogn) approach at first to find the pairs and print them. The approach was to sort the array and keep two indexes(at each end) left and right and find whether pairs that add up to the given sum and print them until the left and right pointers cross.

Then he asked me if I can optimize it further. I gave a O(n) approach using hashing.I maintained a hashset of integers. Put the first number in the hashset, start from the second element and for each number in the array check whether (Given sum \xe2\x80\x93 Current element) is present in the hashset. If so print the pair.

2. Given a binary tree print the bottom view of it.

After a few minutes of thinking I came up with an approach by using the horizontal distance of the nodes from the root. We have to keep note of the last node at a particular horizontal distance while traversing using level order traversal. I used a Hashtable to store the last visited node at each horizontal distance.

This approach uses O(n) time and O(n) space.

He then asked a puzzle.

In a room of 30 people,find the unique number of handshakes given a condition that each one in the room should have shaken hands with everyone.

I just used a simple logic. In a room of 2 people,there is only one unique hand shake(First person has 1 unique handshake whereas the second one has none). If we consider three people,n = 3 (A,B,C)

A has two unique handshakes B,C.(n-1 handshakes)

B has 1 unique handshake C.(n-2 handshakes)

C doesn\xe2\x80\x99t have any unique handshake.

total number of handshakes = n-1 + n-2 = 2n-3\xc2\xa0 = 3.(which is the sum of first two natural\xc2\xa0 numbers)

Similarly for n persons the unique handshakes would be the sum of first n-1 natural numbers.

For n = 30, Total number of unique handshakes is 29*30/2\xc2\xa0 = 29*15 = 435.

That\xe2\x80\x99s a wrap. He was happy with the way I approached the solution.

He asked if I had any questions for him. I asked about the culture and working environment at amazon.

Round 5:Bar Raiser Round(1hr)

Two members entered the room and introduced themselves and we just started out with my introduction and my project.

They asked me how many rounds I had before the current one. They just asked a few questions about testing at first.Then they asked two coding questions.

1. Given a binary tree print the Kth largest element in it.

I just instantly started with a brute force solution by finding the inorder traversal of the tree.Then sort the array and find the kth largest in the sorted array which is the (n-k)th element from the start index of the array,where n is the total number of nodes in the tree.This uses O(nlogn) time and O(n) space.

They asked me to optimize it further.

I gave a O(nlogn) time and O(1) space solution by converting the binary tree into doubly linked list and then finding the kth largest in a doubly linked list which can be done in O(nlogn) time.

But there is a better solution for this which I couldn\xe2\x80\x99t think of during the interview.Just construct a minheap of k elements,Insert elements into the heap by traversing the tree using any traversal.

If the heap root element is smaller than the current node in the tree remove the root element from heap and insert the\xc2\xa0 new element to the heap. At last after the complete traversal of the tree the kth largest element will be at the root of the heap.This uses O(nlogk) time and O(k) space.

But they were happy with my approach as they said me to optimise it either by space or time. I gave a constant space approach so they were satisfied.

2.Given a directed graph, find whether there exists a cycle or not.If it does exist print it.

I gave a solution using BFS. I put a node in the queue and then added all possible nodes that can be reached from the current node. While we traverse the nodes we just mark the node as visited in the hashmap in order to avoid indefinite looping between nodes. They were satisfied with my approach.

Finally they asked how HashMap is implemented and how it works(especially how hashcode and index are computed). I designed my own hash function and explained the way get and put functions work in a hashmap. They were happy with my explanation and then they asked how would you modify your hash function in case of collision and discussed in brief about collision handling techniques.

They asked if I had any questions for them. I just asked the same question I asked the previous round.

P.S: For each question we have to write the code clearly. Don\xe2\x80\x99t just dive straight into the coding part, explain your approach clearly. Be confident and they don\xe2\x80\x99t want the optimized solution for the question,they see how the candidate approaches the question.