b'

Amazon Interview Experience (On-Campus for SDE-1) 2019

- Difficulty Level :\nHard
- Last Updated :\n25 May, 2021

Amazon conducted a pool placement at our college in the month of feb 2019. We were shared the link for a test hosted on HackerEarth which contained 20 MCQs (technical \xe2\x80\x93 OS, DB, TOC etc from GeeksQuiz) and 2 Programming problems which had sectional cutoff.\xc2\xa0

Prelims\xc2\xa0
\xc2\xa0

1. Tiling Problem
2. Given a String of the form ab2c3 where the string preceding the integer is repeated that many times, you are supposed to find the Kth character of the string. Eg: ab2c3 => ababc2 => ababcababc

Round 1:\xc2\xa0

The interviewer was friendly and he asked me to relax. He went through my resume and asked me about the apps that I had built at hackathons and how they worked. After that he gave me a simple problem.\xc2\xa0
\xc2\xa0

1. *Given an array of positive and negative integers, print x if +x and -x are present in the array.* I asked for some clarifications whether I should print all distinct *x*\xe2\x80\x98s or if I should print an x if a pair of *+x* and *-x* is encountered. The first approach I told was to use a map and I was keeping a flag for *+x* and *-x* if it\xe2\x80\x99s found once. Later he asked me to print all pairs, so I stored the frequencies of all the elements in the map and iterated through the negative elements and for each element *x* , I would print *x min(count[-x],count[+x])* times. He said he can\xe2\x80\x99t afford that much space and he wanted me to optimise space further. So I told him a 2 pointer approach where I sort the array once and then keep two pointers to the start and end. I would move the start pointer forward if the sum is less than 0 and I\xe2\x80\x99ll move the end pointer backward if the sum is greater than 0. He was fine with the solution and asked me to code it in a paper. I wrote the code and walked him through it.
2. *Design the logic for minimising cash flow in an app like \xe2\x80\x98Splitwise\xe2\x80\x99.* Here the interviewer told me about an app called splitwise which i had used once. In the application each user adds the amount he spends and how it\xe2\x80\x99s shared by other users of the app. The aim is to minimise the number of give and take operations. I initially thought of a very naive approach where I wanted to create classes for each person and expenditure and iterate through the expenditures of other people to find how much a person should give or take. When I took a closer look I got the idea of modelling it as a directed graph and adding directed edges for transactions. With the graph I though of taking the difference between the pair of edges between two people to reduce a give and take operation to a single give/take operation. There was a catch, if A has to give B Rs.10, B has to give C Rs.10, and A has to give C Rs.10, the minimum operation to do is to give Rs.20 from A to C. B is not involved here as he has to spend all he gets. So I said we could preprocess the graph with the numbers on the incoming and outgoing edges. If the total flow is 0, we could remove that node. He seemed convinced with the approach. He gave me a graph after all the preprocessing done and finally asked me how to minimise it. So I used a greedy method. I was settling the amount of the person who has to get the largest amount by giving the amount of the people who has to give lesser amounts and he said that\xe2\x80\x99ll work.

Round 2:\xc2\xa0

At the beginning of this round, the interviewer asked me about the data structures I knew. Linked lists, trees, graphs, arrays etc. was my answer. He asked me how well I knew Dynamic Programming. I said I wasn\xe2\x80\x99t strong in that and he said he\xe2\x80\x99d ask me a question on dynamic programming for sure.\xc2\xa0
\xc2\xa0

1. *Given a generic tree, find the count of all special nodes. A node is a special node if there is a path from root to that node with all distinct elements. The input was not a pointer to a tree. He\xe2\x80\x99d give me an adjacency list and an array of values where the value of ith node in the adjacency list is the ith element in the values array. He asked me not a create a tree out of the given information and rather do it with the adjacency list itself.* I suggested to do a depth first search keeping a set which contains all elements upto a given node. Once I reach a particular node, I check if it\xe2\x80\x99s already in the set. If its already in the set, I return because that element has already been visited and is not a special node. Otherwise I increase the count of a global variable by 1 and push that element to the set. Then I go through the adjacency list of that element and call this function recursively. Once I return from the element after visiting its neighbours, I pop the element from the set . I told him the approached and he asked me to write the code for it. He was convinced with the approach and he liked the code.
2. *Given an integer array, find the longest subsequence with adjacent numbers having a digit in common.* Eg: 1 12 44 29 33 96 89 . The longest subsequence here is { 1 12 29 96 89} and the answer is 5. I initially tried a 2D DP solution where dp[i][j] indicates the length of longest sequence with ending at i containing j as a digit. It\xe2\x80\x99s a N X 10 DP matrix. Interviewer asked me why I needed a 2D DP solution and I struggled to convince him. I wrote the code for it. It wasn\xe2\x80\x99t completely correct. I was missing something. After thinking for a while I narrowed down to a solution containing only 10 elements dp[0], dp[1], dp[2].. dp[9] which is updated everytime I see a new number. I take a number, I go through all digits in the number , and find val = 1 + max(dp[d] for all digits d in the number) . Set this val to dp[d] for all digits in the number. He gave a hint to take the max.

Although I couldn\xe2\x80\x99t completely solve it without a hint, I was confident in the direction I was going and I was continuously interacting with the interviewer. I was selected to the next round.\xc2\xa0

Round 3:\xc2\xa0

The interviewer asked me if I was comfortable with the interview process so far and how the previous interviews were. I said it was good and he gave me the first problem to solve.\xc2\xa0
\xc2\xa0

1. *Given a binary tree, modify the tree satisfying the following* constraints:\xc2\xa0
    1. Value at root must be the sum of left child and right child (not subtrees).
    2. You can\xe2\x80\x99t reduce the value at any node. You can only increase it.
    3. Value of root node must be minimum. I drew a few trees and asked him the output for those examples. He asked me to say it myself and I did. I thought of doing a post order traversal as we need to visit root\xe2\x80\x99s left and right chlid before visiting root. In the post order traversal, we keep the sum of root\xe2\x80\x99s left and right child in a variable sum. We take the difference of this sum and root\xe2\x80\x99s data. If the sum is greater than root\xe2\x80\x99s data, we replace root with the sum. Otherwise we have to distribute the root\xe2\x80\x99s value to root\xe2\x80\x99s left and right child so that all the three conditions are satisfied. (We can\xe2\x80\x99t reduce the value at any node). He asked me to write the code for it and I did. After that he gave me another problem.
2. *Given an array of 0s 1s and 2s, sort this array in one iteration.*
3. How a web page is displayed when you enter a URL in the browser ?

I solved both the problems and wrote code for them. The interviewer asked me about my projects. He was focusing on one machine learning project that I had done and asked a lot of questions about it. I was selected to the next round.\xc2\xa0

Round 4:\xc2\xa0

The interviewer asked me some CS fundamentals in this round as well as some behavioural questions.\xc2\xa0
\xc2\xa0

1. *Difference between threads and processes.*
2. *Deadlocks and its prevention*
3. *Cost of polymorphism in OOPs*
4. *Implementation of virtual methods, dynamic binding, vtables etc.*
5. *Implementation differences between sets and maps.*
6. *Compressed Tries*
7. *Implement a Trie data structure and write functions to insert and search for a few words in it.* I wrote a class to implement a character Trie using a vector of nodes as children. He asked me to improve on space. So I used a hashmap to store the child nodes only if a child exist. I wrote the code for insertion and finding a word and walked him through.
8. *Check if two words are anagrams.* I implemented it first by sorting two strings and comparing them. He asked me to write a better approach. So I used a hashmap to do it.
9. Some behavioural questions like what would you do if you\xe2\x80\x99re recently joined and your boss is out of station, what would your friends tell about you (good and bad), why amazon etc.

This round was difficult as compared to the previous ones. After a discussion with the entire panel, they hired two from the drive. I was one among them.
\xc2\xa0

My Personal Notes\n*arrow_drop_up*

Add your personal notes her

Save

'