

Amazon interview Experience | Set 416 (On Campus for Internship)

- Difficulty Level : [Medium](#)
- Last Updated : 12 Jul, 2019

Coding Round

There were 2 coding questions (10 marks each) and 20 MCQs from various core concepts (OS, DS, Algorithms, DB, C language etc).

- **Question 1 :** <https://www.geeksforgeeks.org/given-a-string-print-all-possible-palindromic-partition/> Given a string, find all possible palindromic partitions of given string.
- **Question 2 :** [You are given with a large paragraph and N words. You have to find a min length subparagraph of the paragraph which contain all those N words in any order. \(Case Insensitive\)](#)

22 students were shortlisted for the interviews.

Interview Round 1

The interviewer asked to introduce myself and then he asked to explain my [projects](#), one of them in detail. Then he gave me 2 coding questions

Question 1 : Given a sorted array of integers (might contain duplicates), you should write a function which returns the first index of an element.

E.g. `arr = [1,2,2,2,3,3,3,4]` `val=3` ; `ans = 4`

Answer : It is a basic [binary search](#) except for the condition that you will check not only if the current element is mid, but also if its previous element is less than it (if a previous element exists) and only then return mid.

Else recur on the left part till the above condition is met.

Question 2 : Given a binary tree check if its a binary search tree or not? [Check for BST](#)

It is very important to write neat code on paper. The interviewer dry runs a few examples on the code.

Interview Round 2

The interviewer asked me about my previous interview and its questions.

Later he gave me 2 coding questions.

Question 1 : Given an array of integers with the property that `arr[j] ≤ arr[j-1]` is either 1, 0, -1 and a search value, provide an efficient search mechanism.

Answer : The naive approach is to perform linear search which is $O(n)$. Then I thought along lines of binary search but, as such we cannot divide and recur because of the array's property.

After some time, I came up with a method which starts with the first element and compares each element with the search value and increment its location by absolute difference of the current value and search key.

This is a worst case $O(n)$ algorithm, but does far better than linear search in generic case.

Question 2 : [Given an array of integers \(duplicates allowed\) return if it is a set of contiguous integers or not?](#)

Input: 5,2,3,6,4,4,6,6 Output: Yes (as it is from set of [2,3,4,5,6])

Answer : First, I gave him a hashing based approach. Initially, make a hash of all the elements and then pick the first element and increment in 1's till you find a value not present in the hash. Similarly decrement in 1's from the first value and get the count. If the count equals hash size it is true else it is false.
It is $O(n)$ time complexity and $O(n)$ space-complexity.
He asked me to improve the space complexity to $O(1)$.

I tried a lot but didn't get to solution. Then, he simplified the problem by giving hints and making it a [duplication detection problem in a array](#) without using any additional space provided the numbers are in the range $[0, \text{length_of_array} - 1]$.

This can be done by accessing element at index j as $\text{arr}[j] \% n$ and as make $\text{arr}[\text{arr}[i]] = \text{arr}[\text{arr}[i]] + n * \text{arr}[i]$. So if at any stage $\text{arr}[j] / n$ is > 1 implies we have visited it earlier. He was satisfied with this solution.

DS Questions

1. Give use cases when you will prefer Hash table instead of BST and vice-versa?
2. What happens in case of duplicate insertions in Hash table and BST. What is its effect on the complexity of search?

I answered these and he was happy with the answers.

The interviewers are very comforting and give clues to direct us to solution.

My Personal Notes

Add your personal notes here

Save