**AMAZON OA QUESTIONS**

**Q1.)** [**https://leetcode.com/discuss/post/6533984/amazon-oa-are-you-ready-2025-by-anonymou-p8zl/**](https://leetcode.com/discuss/post/6533984/amazon-oa-are-you-ready-2025-by-anonymou-p8zl/)

**Q2.)** Your team at Amazon has recently adopted a new password validation scheme for internal user accounts.

A password consists of lowercase English letters only and is valid only if it contains at least one vowel and at least one consonant. Vowels are the characters 'a', 'e', 'i', 'o', 'u'. The rest are consonants.

The strength of a password is defined as the maximum number of contiguous subsegments the entire password can be divided into such that each subsegment is a valid password.

Given a password string, find its strength. Return 0 if the password itself is not valid.

Note: A subsegment of a string is a segment composed of contiguous characters of the original string, taken in the same order.

Example

password = "hackerrank"

This password can be divided into 3 subsegments: "hack", "er", and "rank". Each segment contains at least one vowel and one consonant.

It can be shown that the string cannot be divided into more than 3 valid subsegments. The strength of the password is 3.

Function Description

Complete the function findPasswordStrength in the editor below.

findPasswordStrength has the following parameter(s):

string password: the given password

Returns int: the strength of the password

Constraints

1≤∣password∣≤10 (the length of password)

password consists of lowercase English letters only.

**META SCREENING**

**Q1.)** <https://leetcode.com/problems/binary-tree-right-side-view/description/>

Explained a BFS-based approach, which the interviewer was fine with and asked me to write the code.

After that, as a follow-up, asked me if this can be done using DFS. Asked me to write code for that as well.

**Q2.**) Add two numbers (in string format, could be very big numbers), contains decimal as well (For e.g. 99.898 + 7.6). No (-) or (+) characters in the input, only digits and decimal.

**GOOGLE OA QUESTION**

We are given a two-player combinatorial game involving 12 cards laid out on a table. The cards come in 4 different colours, with exactly 3 cards for each colour. Initially, each card is placed separately, resulting in 12 stacks (each of size 1).

Game Rules

1. Move Definition:

A move consists of selecting two stacks and merging them (i.e., stacking one on top of the other) if and only if:

• The top cards of both stacks are of the same colour.

• The two stacks have the same number of cards.

2. Losing Condition:

• A player loses if, on their turn, no legal move exists (i.e., there are no two stacks satisfying both conditions).

Question: can player 1 win

**Amazon Interview | SDE 2 | March 2025**

Recently, I had the opportunity to interview with Amazon for an SDE-2 position. Initially, I was contacted for an SDE-3 role, but after an initial screening with the Hiring Manager, the HR team proceeded with SDE-2.

Background: Senior Software Engineer  
Experience: 5.5 years

Note: Every round started with a quick introduction.

**Phone Screen**  
Interviewer: Senior SDE  
Questions: Two Leetcode medium-hard problems (Top 50 Amazon-tagged)  
Self-feedback: Went really well

HR reached out the very next day, requesting slots for four onsite rounds to be completed within 2-3 days.

**Onsite Rounds**

**Round 1**  
Interviewer: SDE-2

Questions:  
One hard DSA problem (Top 100 Amazon-tagged)  
A follow-up question on the same problem, only approach no code needed  
Two Leadership Principles (LP) questions  
Self-feedback: Went really well

Round 2  
Interviewer: Senior SDE

Questions:  
One medium DSA problem (not found on Leetcode) – could be solved using Greedy  
One standard LLD problem (Amazon asks this frequently; easily found in resources)  
No LP questions in this round  
Self-feedback: DSA -> Went really well, LLD -> Went well from my perspective

Round 3  
Interviewer: SDM

Questions:  
One hard HLD problem (not straightforward, had never heard of it before)  
Two LP questions at the end  
Self-feedback: Managed to answer all follow-up questions. I feel I did well, but the final decision likely depends on the SDM

Round 4  
Interviewer: SDE-2

Questions:  
One medium DSA problem (famous design problem on Leetcode)  
Two LP questions at the end  
Self-feedback: Went really well

**Amazon 0A – Feb 2025**

Full OA was 2 coding questions + some GDS questions + some behavioural/culture fit questions. (MCQ)

Amazon L5, Feb 2025

Don't really remember Q1, but it was a pretty straightforward question

Here's Q2:  
Given two strings s and t, return the number of subsequences of s that are lexicographically strictly greater than t, modulo 1e9 + 7.

Example  
s = "aba"  
t = "ab"  
=> expected return: 3

Subsequence of s | strictly greater than t?

-------------------------------------------------

a | no

ab | no

aa | no

aba | yes

b | yes

ba | yes

a | no

I came up with a DFS+Memo approach, that keeps track of the index of s and the length of the current subsequence I am pathing, so typically kinda a 2D top down DP, but only passed like 11 or 12 test cases out of 15 (TLE) :(

**AMAZON PHONE SCREENING INTERVIEW FOR GRAD SDE ROLE**

[Amazon](https://leetcode.com/discuss/topic/amazon/)[Technical Interview](https://leetcode.com/discuss/topic/technical-interview/)[Career](https://leetcode.com/discuss/topic/career/)[interview question](https://leetcode.com/discuss/topic/interview-question/)[Interview](https://leetcode.com/discuss/topic/interview/)

**Interview Experience**

**Duration:** 30–40 minutes  
**Format:** Technical Interview (Focused on DSA and Concepts)

**1. Introduction**

* The interviewer introduced himself.
* He asked me to introduce myself.
* Immediately transitioned to coding without much small talk.

**2. Coding Question 1 — One Missing Number**

**Problem:**  
Find the missing number from a continuous vector of numbers.

**My Response:**

* Asked about edge cases.
* Wrote a brute-force approach and explained:
  + **Time Complexity and Space Complexity**
* Mentioned that it could be optimized.
* Interviewer stopped me from writing optimized code and asked for verbal explanation.

**Optimized Approach (Verbal):**

* Explained the optimized approach clearly.
* Discussed:
  + **Time Complexity and Space Complexity**
* Interviewer was satisfied with the explanation.

**3. Follow-up — Two Missing Numbers**

**Problem Update:**  
Now two numbers were missing from a continuous array.

**Discussion:**

* He asked whether the brute-force approach would still work.
* I confirmed that it would.
* He then asked me to write the **most optimized approach** for this case.

**My Response:**

* Wrote the optimized solution.
* Discussed:
  + **Time Complexity and Space Complexity along with code structure**
* Had a short discussion on trade-offs.
* Interviewer was satisfied.

**4. Conceptual Question — Hash Map Lookup**

**Question:**  
What is the time complexity to search an element in a hash map?

**My Answer:**

* For map in C++ → O(log n) (uses Red-Black Tree).
* For unordered\_map → O(1) average (uses hashing).

**Follow-up:**

* Explained the difference between map and unordered\_map.
* Interviewer seemed happy with the answer.

**5. Wrap-up**

* Interview ended due to time constraints.
* I asked him:

“When was the last time you implemented or used a high-level data structure?”

**His Response:**

* Described a recent real-world scenario involving maps.

**Summary**

* Focused mainly on:
  + Coding skills
  + Optimizing solutions
  + Analyzing time & space complexities
* Interviewer appreciated verbal clarity and quick thinking.