BuildHer Interview Questions

Easy 1

Given a string s consisting of words and spaces, return *the length of the* ***last*** *word in the string.*
A **word** is a maximal
substring consisting of non-space characters only.
Example 1:
Input: s = "Hello World"
Output: 5
Explanation: The last word is "World" with length 5.
Example 2:
Input: s = " fly me to the moon "
Output: 4
Explanation: The last word is "moon" with length 4.
Example 3:
Input: s = "luffy is still joyboy"
Output: 6
Explanation: The last word is "joyboy" with length 6.
Constraints:

- 1 <= s.length <= 104
- s consists of only English letters and spaces ' '.
- There will be at least one word in s.

Algorithm and Logic:

Step 1: Get the length of the input string.

Step 2: Initialize a variable to keep track of the current length of the word.

Step 3: Start from the end of the string.

Step 4: Iterate through the string from the end.

1. If the current character is a space, check if **curr_len** is not 0. If true return **curr_len** because a word has been found. If false, return **curr_len** to 0.

2. If the current character is not a space, increment **curr_len**.

3. Move to the previous character in the string.

Step 5: If the loop completes and no word has been found yet, return the last found **curr_len.**

Step 6: Take input from the user, call the function with the input, print the result.

Medium 2

Given an integer array of size n, find all elements that appear more than ⌊ n/3 ⌋ times.
Example 1:
Input: nums = [3,2,3]
Output: [3]
Example 2:
Input: nums = [1]
Output: [1]
Example 3:
Input: nums = [1,2]
Output: [1,2]
Constraints:

- 1 <= nums.length <= 5 * 104
- -109 <= nums[i] <= 109

Algorithm and Logic:

Step 1: Use counter to count the occurrences of each element in the list.

Step 2: Calculate the threshold, which is the minimum count for an element to be considered a majority element.

Step 3: Initialize the empty list to store the majority elements.

Step 4: Iterate through the counts dictionary.

Check if the count is greater than the threshold. If true, add the element to the result list.

Step 5: Return the list of majority elements.

Step 6: Take input from the user, convert it into a list of integers, call the function with the input list, and print the result.

Hard 2

You are given a string s. You can convert s to a palindrome by adding characters in front of it. Return *the shortest palindrome you can find by performing this transformation*.

Example 1:

Input: s = "aacecaaa"

Output: "aaacecaaa"

Example 2:

Input: s = "abcd"

Output: "dcbabcd"

Constraints:

- 0 <= s.length <= 5 * 104
- s consists of lowercase English letters only.

Algorithm and Logic:

Step 1: Get the length of the input string.

Step 2: Define the helper function to check if a substring is a palindrome.

Step 3: Find the longest palindrome prefix.

Step 4: Build the shortest palindrome by adding the remaining characters in reverse order and concatenate them with the original string.

Step 5: Take input from the user, call the function with the input string and print the result.