Infinity Champions Phase 4_Week 1 PS 2

Question1:

Problem Statement: You are a software engineer at a fintech company responsible for maintaining the backend of a high-traffic digital wallet platform. One of your tasks is to manage unique transaction IDs generated in sequence. These transaction IDs are stored as arrays of digits due to their potentially large size, which can exceed standard integer limits in some systems.

Each time a new transaction is initiated, the system must increment the last transaction ID by one and return the new ID in the same digit-array format. For example, a transaction ID represented as [1, 2, 9] should become [1, 3, 0].

Your task is to implement a program that handles this incrementation accurately, including handling carry-overs (e.g., [9, 9, 9] becoming [1, 0, 0, 0]). This program will be used in the core transaction processing system, so it must be efficient and reliable.

Constraints:

1 <= digits.length <= 100

0 <= digits[i] <= 9

digits does not contain any leading 0's.

Example 1:

Input: digits = [1,2,3]

Output: [1,2,4]

Explanation: The array represents the integer 123.

Incrementing by one gives 123 + 1 = 124.

Thus, the result should be [1,2,4].

Example 2:

Input: digits = [4,3,2,1]

Output: [4,3,2,2]

Explanation: The array represents the integer 4321.

Incrementing by one gives 4321 + 1 = 4322.

Thus, the result should be [4,3,2,2].

Example 3:

Input: digits = [9]

Output: [1,0]

Explanation: The array represents the integer 9.

Incrementing by one gives 9 + 1 = 10.

Thus, the result should be [1,0].

Input0:

4 3 2 5

Output0:

4 3 2 6

Input1:

1 2

Output1:

1 3

Input2:

1 9 8

Output2:

1 9 9

Input3:

1 9 8 6

Output3:

1 9 8 7

Input4:

9 9

Output4:

1 0 0

Input5:

9 9 2 4

Output5:

9 9 2 5

Question2:

Problem Statement: You are working as a software developer for an e-commerce platform that manages millions of products. Each product is assigned a unique ID, and these IDs are stored in a sorted array for quick access. When a new product is added to the catalog, the system needs to determine the correct position to insert the product ID into the sorted array to maintain order. If the product ID already exists, the system should return its current index; otherwise, it should return the index where it should be inserted.

Given that the array can contain millions of entries, performance is critical. You are tasked with implementing an efficient algorithm that operates in O(log n) time to find the correct index for the product ID.

Your solution will be a key part of the product management module, ensuring fast and consistent updates to the catalog while maintaining optimal search and insert performance.

Constraints:

1 <= nums.length <= 10^4

-10^4 <= nums[i] <= 10^4

nums contains distinct values sorted in ascending order.

-10^4 <= target <= 10^4

Example 1:

Input: nums = [1,3,5,6], target = 5

Output: 2

Example 2:

Input: nums = [1,3,5,6], target = 2

Output: 1

Example 3:

Input: nums = [1,3,5,6], target = 7

Output: 4

Input0:

1 3 5 6 7

2

Output0:

1

Input1:

1 3 5 6 7

8

Output1:

5

Input2:

1 2 3 5 6 7

4

Output2:

3

Input3:

1 2 3 5 6 7

6

Output3:

4

Input4:

1 2 3 4 5 6 7

2

Output4:

1

Input5:

1 2 3

4

Output5:

3

Question 3:

Problem Statement: You are a developer working for a messaging app that analyzes chat patterns to improve spam detection and user behavior analytics. Your team has observed that users often exhibit repetitive typing behavior when expressing emotions or urgency—such as typing “aaa”, “hhh”, or “zzz”—which are strings made of a single repeated character.

To help identify such patterns, you are tasked with analyzing user messages represented as lowercase alphabetic strings. Specifically, you need to detect the longest special substring—a substring made up of only one repeated character—that appears at least three times in the message.

Write a function that takes a message string as input and returns the length of the longest such special substring. If no special substring occurs at least three times, return -1.

This function will support spam detection algorithms and provide insights into how users express emphasis or emotion in repeated character sequences.

Constraints:
3 <= s.length <= 5 * 10^5
s consists of only lowercase English letters.

Example 1:

Input: s = "aaaa"
Output: 2
Explanation: The longest special substring which occurs thrice is "aa": substrings "aaaa", "aaaa", and "aaaa".
It can be shown that the maximum length achievable is 2.
Example 2:

Input: s = "abcdef"
Output: -1
Explanation: There exists no special substring which occurs at least thrice. Hence return -1.
Example 3:

Input: s = "abcaba"

Output: 1

Explanation: The longest special substring which occurs thrice is "a": substrings "abcaba", "abcaba", and "abcaba".

It can be shown that the maximum length achievable is 1.

Input0:

aaaabcdef

Output0:

2

Input1:

abaabab

Output1:

1

Input2:

prepster

Output2:

-1

Input3:

preppp

Output3:

1

Question 4:

Problem Statement: You are a software developer working with an EdTech company that builds intelligent test generators for teachers. One feature being developed is an algorithm to automatically adjust the answer key of multiple-choice tests to make patterns less predictable and challenge students' test-taking habits.

A teacher wants to design a test with n true/false questions, where each answer is either 'T' (true) or 'F' (false). To increase difficulty and possibly mislead students relying on guessing patterns, the teacher intends to maximize the number of consecutive identical answers—that is, as many consecutive 'T's or 'F's as possible—so that the test appears non-random and psychologically challenging.

To achieve this, the system is allowed to modify up to k answers in the original key (changing a 'T' to 'F' or vice versa).

Your task is to implement a function that takes in the original answer key as a string (answerKey) and an integer k, and returns the maximum number of consecutive identical answers (either 'T' or 'F') that can be obtained by modifying at most k answers.

This feature will be used in automated test balancing to evaluate its impact on student performance under varying psychological conditions.

Constraints:

n == answerKey.length
1 <= n <= 5 * 10^4
answerKey[i] is either 'T' or 'F'
1 <= k <= n

Example 1:

Input: answerKey = "TTFF", k = 2
Output: 4
Explanation: We can replace both the 'F's with 'T's to make answerKey = "TTTT".

There are four consecutive 'T's.

Example 2:

Input: answerKey = "TFFT", k = 1

Output: 3

Explanation: We can replace the first 'T' with an 'F' to make answerKey = "FFFT".

Alternatively, we can replace the second 'T' with an 'F' to make answerKey = "TFFF".

In both cases, there are three consecutive 'F's.

Example 3:

Input: answerKey = "TTFTTFTT", k = 1

Output: 5

Explanation: We can replace the first 'F' to make answerKey = "TTTTTFTT"

Alternatively, we can replace the second 'F' to make answerKey = "TTFTTTTT".

In both cases, there are five consecutive 'T's.

Input0:

TTFT

1

Ouptut0:

4

Input1:

TTFTFF

3

Output1:

6

Input2:

TTFTFF

2

Output2:

5

Input3:

```
TTFTFFTTF
3
```

Output3:

```
8
```

Question 5:

Problem Statement: You are a data analyst at a fitness technology company that tracks users' daily calorie intake through wearable devices. Each day, the device records the net calories consumed, which can be positive (calories gained) or negative (calories burned).

Your team is building a feature to detect when users meet their weekly calorie goals. Specifically, you need to determine the shortest period of consecutive days where a user's total calorie intake is at least a given threshold k. This allows the system to identify effective diet or workout streaks and provide real-time feedback to users.

Given an integer array nums where nums[i] represents the net calorie intake on the i-th day, and an integer k as the calorie threshold, write a function to return the length of the shortest non-empty subarray whose sum is at least k. If no such subarray exists, return -1.

This feature supports personalized goal tracking and motivation prompts.

Constraints:

1 <= nums.length <= 10^5
-10^5 <= nums[i] <= 10^5
1 <= k <= 10^9

Example 1:

Input: nums = [1], k = 1
Output: 1
Example 2:

Input: nums = [1,2], k = 4
Output: -1
Example 3:

Input: nums = [2,-1,2], k = 3
Output: 3

Input0:

1 2

3

Output0:

2

Input1:

1 2 1 4

3

Output1:

1

Input2:

1 2 1 4

6

Output2:

3

Input3:

1 2 1 4

5

Output3:

2

Question 6:

Problem Statement: You are a software engineer at a stock trading platform that analyzes real-time price data to help investors make informed decisions. One feature your team is developing identifies when a stock is likely to face short-term competitive pricing pressure.

You are given an array nums, where nums[i] represents the price of a stock at a particular time during the trading day. For each time point i, you want to determine how many future time points have a lower stock price than nums[i]. This information helps detect possible downward trends or volatility patterns in pricing behavior.

Write a function that returns an array counts, where counts[i] represents the number of prices that are smaller than nums[i] and occur after index i in the array.

This analysis supports trend prediction algorithms and helps traders make smarter short-term decisions by highlighting when a stock price is relatively high compared to its upcoming values.

Constraints:
1 <= nums.length <= 10^5
-10^4 <= nums[i] <= 10^4

Example 1:

Input: nums = [5,2,6,1]
Output: [2,1,1,0]
Explanation:
To the right of 5 there are 2 smaller elements (2 and 1).
To the right of 2 there is only 1 smaller element (1).
To the right of 6 there is 1 smaller element (1).
To the right of 1 there is 0 smaller element.
Example 2:

Input: nums = [-1]
Output: [0]
Example 3:

Input: nums = [-1,-1]

Output: [0,0]

Input0:

5 3 8 1 6

Output0:

2 1 2 0 0

Input1:

5 3 8

Output1:

1 0 0

Input2:

5 3 2 1

Output2:

3 2 1 0

Input3:

5 3 2 1 2

Output3:

4 3 1 0 0