Assessment 8

Question # 01 : Matching Character Coderbyte

Problem Statement

You are given a string consisting of only lowercase alphabetic characters. Your task is to determine the **largest number of unique characters that exist between any pair of matching letters** in the string.

- A **pair of matching letters** means the same character appearing twice in the string (e.g., 'a' ... 'a').
- For each such pair, count how many **unique characters** are present between them.
- Your program should return the **maximum** such count among all pairs.
- If the string does not contain any pair of matching letters, return 0.

Examples

1. **Input:** "ahyjakh"
   Explanation:
   - Between the pair of 'a': unique chars = {h, y, j} → 3
   - Between the pair of 'h': unique chars = {y, j, a, k} → 4
     Output: 4
2. **Input:** "ghececgkaem"
   Explanation:
   - The farthest 'e' pair has unique chars {c, g, k, a, m} → 5
     Output: 5
3. **Input:** "mmmerme"
   Output: 3
4. **Input:** "abccdefghi"
   Output: 0 (no matching pairs that enclose characters)

Question # 02 : Histogram Area

Problem Statement

You are given an array of non-negative integers where each element represents the **height of a bar** in a histogram. Each bar has a **width of 1 unit**.

Your task is to determine the **largest rectangular area** that can be formed under the histogram by choosing one or more adjacent bars.

- The rectangle must be continuous (cannot skip bars).
- The area is calculated as:

Area=Height of chosen bar(s)×Width (number of bars)

- You must return the **maximum possible area** for any rectangle under the histogram.
- The array will always contain at least 1 element.

Examples

1. **Input:** [2, 1, 3, 4, 1]
    - Best rectangle is using bars of heights [3, 4].
    - Width = 2, Height = 3 → Area = 6.
      Output: 6
2. **Input:** [6, 3, 1, 4, 12, 4]
   Explanation:
    - Best rectangle is the single bar of height 12.
    - Area = 12 × 1 = 12.
      Output: 12
3. **Input:** [5, 6, 7, 4, 1]
   Explanation:
    - Best rectangle is using bars [5, 6, 7].
    - Width = 3, Height = 5 → Area = 15, but
    - Actually, the max is with [6, 7, 4] using height 4 → Area = 16.
      Output: 16

Question 3 : Min Window substring

Problem Statement: Minimum Window Substring (Subsequence version)

You are given two strings **s1** and **s2**.
Your task is to find the **smallest substring** in **s1** such that **s2 appears as a subsequence** within that substring.

Important Details

- The characters of **s2** must appear in the **same order** in the chosen substring of **s1** (but not necessarily consecutively).
- If there are multiple substrings with the **same minimum length**, return the one that appears **first (leftmost)** in **s1**.
- If no such substring exists, return an **empty string** ("").
- Both strings contain only lowercase English letters.

Examples

1. **Input:**
   s1 = "geeksforgeeks"
   s2 = "eksrg"

Output:
"eksforg"

Explanation:

- "eksforg" contains eksrg as a subsequence.
- It is the smallest and leftmost such substring.

---

2. **Input:**
   s1 = "abcdebdde"
   s2 = "bde"

Output:
"bcde"

Explanation:

- Both "bcde" and "bdde" are valid.
- "bcde" occurs first, so we return that.

---

3. **Input:**
   s1 = "ad"
   s2 = "b"

Output:
""

Explanation:

- There is no substring of s1 where s2 appears as a subsequence.