DSA Assessment 3

Instructions:

Tool used: VSCode or Intellij Idea with proper environment setup

Topic Covered: Sliding Window, Two Pointer, Sorting, Binary Search, & Prefix Sum

Test date: 31st Jan, 2025

Test mode: Offline (C-107, C-108)

Reporting time: 9:20 AM - 9:40 AM

Test time: 10:00 AM - 1:00 PM

Submission time: 1:00 PM - 1:15 PM

Solution discussion time: 2:30 PM - 4:30 PM

Offline Assessment Guidelines:

- **Internet Usage:**
    - Ensure that the internet is closed during the assessment. The assessment will be conducted offline using an IDE.
    - Use of any online resources is prohibited.
- **Mobile Phones:**
    - Switch off your mobile phone.
    - Place your mobile phone inside your bag, and keep your bag outside the assessment venue.
- **Items to Carry:**
    - Bring your **laptop** and **charger**.
    - Carry a **pen**, **water bottle**, and **A4 sheets** for dry run submission.

Marking scheme:

- **Clean & Concise Code:** Proper naming conventions, modular code, reusable functions.
- **Code correctness:** Does the solution solve the problem as expected?
- **Efficiency:** time and space complexity, optimal solution.
- **Edge Cases:** handling of boundary conditions and exceptional inputs.
- **Time and Space Complexity:** Document the complexity analysis of your solution in the code using comments.
- **Dry run:** Please perform a dry run of the program and submit the results as a hard copy. Ensure that your dry run includes step-by-step tracing of the input, intermediate computations, and the final output.

Submission Guidelines for Weekly Test:

- **Join the Google Classroom:**
  - Link: Google Classroom
  - Class Code: **msomyj2**
- **Prepare Your Google Document:**
  - Create a Google Document with your **name** and include your **GUID** (Global Unique Identifier).
  - Copy your code from **VS Code** and paste it into the Google Document.
- **Download as PDF:**
  - Once completed, download the Google Document as a **PDF file**.
- **Submit the PDF:**
  - Upload the PDF to this assignment link: Weekly Test Submission
- **Submit the hard copy:**
  - After completing the test, print your work and submit a **hard copy** for final review.

Problem 1: The Fruit Basket Problem

Maria's Fruit Market is a popular destination for fresh fruits. Maria takes great care in selecting the freshest fruits from local farmers. She has a unique way of arranging fruits in baskets to attract customers. Each basket contains a variety of fruits, and Maria ensures that the weights of the fruits in each basket do not differ significantly.

One day, Maria received a new shipment of fruits. She weighed each fruit and stored the weights in an array. However, she noticed that some fruits had unusual weights, which would affect the overall appearance of her baskets.

To solve this problem, Maria decided to replace some fruits with new ones. She can replace up to K fruits. Help Maria find the minimum possible difference between the heaviest and lightest fruit in the basket after replacing K fruits.

Constraints

1 ≤ N ≤ 10^5

0 ≤ K ≤ N

-10^9 ≤ weights[i] ≤ 10^9

Input Format

The first line contains two space-separated integers, N and K.

The second line contains N space-separated integers representing the array weights[].

Output Format

Print the minimum possible difference between the heaviest and lightest fruit.

Example 1:

Input:

6 3

-1 3 -1 8 5 4

Output: 2

Explanation:

Replace batches with moisture content -1, -1, and 8 with new batches of moisture content 3, 4, and 5. The resulting array is [3, 3, 4, 5, 5, 4]. The minimum difference is 5 - 3 = 2.

Problem 2: The Poet's Dilemma

A famous poet, known for her beautiful and melodious verses, is working on a new piece. She wants to create a poem that not only rhymes but also has a specific structure. The poem must contain all five English vowels ('a', 'e', 'i', 'o', 'u') at least once and in alphabetical order.

The poet has written a long string of vowels, but she needs help finding the longest beautiful substring that meets her requirements. A beautiful substring is one that contains all five vowels in alphabetical order.

Help the poet find the length of the longest beautiful substring in her poem.

Input Format:

The input contains a single string word consisting of English vowels.

Output Format:

Print the length of the longest beautiful substring in word. If no such substring exists, print 0.

Example 1:

Input:

aeiaaioaaaaeiiiiouuuooaauuaeiu

Output:

13

Explanation:

The longest beautiful substring in word is "aaaaeiiiiouuu" of length 13.

Example 2:

Input:

aeeeiiiioooauuuaeiou

Output:

5

Explanation:

The longest beautiful substring in word is "aeiou" of length 5.

Example 3:

Input:

a

Output:

0

Explanation:

There is no beautiful substring, so return 0.

Constraints:

1 <= word.length <= 5 * 10^5

word consists of characters 'a', 'e', 'i', 'o', and 'u'.

Problem 3:

Given an integer array nums and an integer k, split nums into k non-empty subarrays such that the largest sum of any subarray is minimized.

Return the minimized largest sum of the split.

A subarray is a contiguous part of the array.

Constraints:

1 ≤ nums.length ≤ 1000

0 ≤ nums[i] ≤ 10^6

1 ≤ k ≤ min(50, nums.length)

Input Format:

The first line contains two space-separated integers, n and k, where n is the number of elements in the nums array and k is the number of splits.

The second line contains n space-separated integers representing the nums array.

Output Format:

Print the minimized largest sum of the split.

Example 1:

Input:

5 2

7 2 5 10 8

Output:

18

Explanation:

The best way to divide the playlist is into [7, 2, 5] and [10, 8], where the maximum duration among the two segments is only 18.

Example 2:

Input:

5 2

1 2 3 4 5

Output:

9

Explanation:

The best way to divide the playlist is into [1, 2, 3] and [4, 5], where the maximum duration among the two segments is only 9.

Problem 4: The Digital Code Problem

Dr. Maria is a cryptologist working on a top-secret project. She's been assigned to analyze a digital code consisting of a string of digits from 0 to 9. The code is used to transmit sensitive information, and it's crucial to identify the longest sequence of digits that meets specific security requirements.

The security protocol dictates that the code should be semi-repetitive, meaning that there should be at most one adjacent pair of the same digit. For example, the sequences "0010", "002020", and "0123" are semi-repetitive, while "00101022" and "1101234883" are not.

Help Dr. Maria find the length of the longest semi-repetitive substring in the given digital code.

Input Format:

The input contains a single string s consisting of digits from 0 to 9.

Output Format:

Print the length of the longest semi-repetitive substring in s.

Example 1:

Input:

52233

Output:

4

Explanation:

The longest semi-repetitive substring is "5223".

Example 2:

Input:

5494

Output:

4

Explanation:

s is a semi-repetitive string.

Example 3:

Input:

1111111

Output:

2

Explanation:

The longest semi-repetitive substring is "11".

Constraints:

1 ≤ s.length ≤ 50

'0' ≤ s[i] ≤ '9'

Problem 5: The Investment Portfolio Problem

Alex is a financial analyst who manages investment portfolios for clients. She's analyzing a portfolio consisting of n assets, each with a value represented by the array nums. The client wants to invest in a contiguous block of k assets, and Alex needs to determine which block will yield the maximum average return.

Help Alex find the maximum average value of a contiguous subarray of length k in the portfolio.

Input Format:

The first line contains two space-separated integers, n and k, where n is the number of assets in the portfolio and k is the number of assets to invest in.

The second line contains n space-separated integers representing the values of the assets in the portfolio.

Output Format:

Print the maximum average value of a contiguous subarray of length k in the portfolio, rounded to five decimal places.

Example 1:

Input:

6 4

1,12,-5,-6,50,3

Output:

12.75000

Explanation:

Maximum average is (12 - 5 - 6 + 50) / 4 = 51 / 4 = 12.75

Example 2:

Input:

1 1

5

Output:

5.00000

Constraints:

n == nums.length

1 ≤ k ≤ n ≤ 10^5

-10^4 ≤ nums[i] ≤ 10^4

--

Thanks and Regards,
Aniruddha M Agrawal
Lead Technical Trainer
School of Computing Science & Engineering (SCSE),
Galgotias University
Former Software Development Engineer, Amazon
Founder, CodeTatva
Mobile: +91-9520401457
E-mail: aniruddha.agrawal@galgotiasuniversity.edu.in
LinkedIn:
https://www.linkedin.com/in/aniruddha-m-agrawal/