DSA Assessment 2

Instructions:

Tool used: VSCode or Intellij Idea with proper environment setup

Topic Covered: Binary Search

Test date: 15th Jan, 2025

Test mode: Offline (A-006)

Reporting time: 10:10 AM - 10:25 AM

Test time: 10:30 AM - 12:30 PM

Submission time: 12:30 PM - 12:45 PM

Solution discussion time: 2:00 PM - 3:30 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:

Given a **sorted** array **arr[]** and an integer **target**, the task is to find the number of occurrences of **target** in given array.

Input: arr[] = [1, 1, 2, 2, 2, 2, 3], target = 2
Output: 4
Explanation: 2 occurs 4 times in the given array.

Input: arr[] = [1, 1, 2, 2, 2, 2, 3], target = 4
Output: 0
Explanation: 4 is not present in the given array.

Problem 2:

Koko loves to eat bananas. There are n piles of bananas, the ith pile has piles[i] bananas. The guards have gone and will come back in h hours.

Koko can decide her bananas-per-hour eating speed of k. Each hour, she chooses some pile of bananas and eats k bananas from that pile. If the pile has less than k bananas, she eats all of them instead and will not eat any more bananas during this hour.

Koko likes to eat slowly but still wants to finish eating all the bananas before the guards return.

Return the minimum integer k such that she can eat all the bananas within h hours.

Example 1:

Input: piles = [3,6,7,11], h = 8

Output: 4

Example 2:

Input: piles = [30,11,23,4,20], h = 5

Output: 30

Example 3:

Input: piles = [30,11,23,4,20], h = 6

Output: 23

Constraints:

1 <= piles.length <= 10^4

piles.length <= h <= 10^9

1 <= piles[i] <= 10^9

Problem 3:

You are given an integer array ranks representing the ranks of some mechanics. ranksi is the rank of the ith mechanic. A mechanic with a rank r can repair n cars in r * n2 minutes.

You are also given an integer cars representing the total number of cars waiting in the garage to be repaired.

Return the minimum time taken to repair all the cars.

Note: All the mechanics can repair the cars simultaneously.

Example 1:

Input: ranks = [4,2,3,1], cars = 10

Output: 16

Explanation:

- The first mechanic will repair two cars. The time required is 4 * 2 * 2 = 16 minutes.

- The second mechanic will repair two cars. The time required is 2 * 2 * 2 = 8 minutes.

- The third mechanic will repair two cars. The time required is 3 * 2 * 2 = 12 minutes.

- The fourth mechanic will repair four cars. The time required is 1 * 4 * 4 = 16 minutes.

It can be proved that the cars cannot be repaired in less than 16 minutes.

Example 2:

Input: ranks = [5,1,8], cars = 6

Output: 16

Explanation:

- The first mechanic will repair one car. The time required is 5 * 1 * 1 = 5 minutes.

- The second mechanic will repair four cars. The time required is 1 * 4 * 4 = 16 minutes.

- The third mechanic will repair one car. The time required is 8 * 1 * 1 = 8 minutes.

It can be proved that the cars cannot be repaired in less than 16 minutes.

Constraints:

1 <= ranks.length <= 10^5

1 <= ranks[i] <= 100

1 <= cars <= 10^6