Data Structures and Algorithm Analysis **BSDS F23**

Lab 13 – 15-12-2024

Note: Solve following tasks according to required time complexity.

Task 1: You are given a vector of integers. Your task is to find and return the first non-repeating element in the vector. If all elements repeat, return -1. You are not allowed to use any additional data structures, solve this problem using only a vector<int>. You have to solve it in linear time.

Constraints:

- The vector can contain up to 1,000,000 elements
- The elements in the vector can range from -1000 to 1000

Sample 1:

Input: 4, 5, 6, 7, 4, 6, 5, 6, 4, 5, 5, 6

- Output: 7

Sample 2:

Input: 1, 5, 2, 5, 4, 1, 2, 3, 4, 3, 5
Output: -1

Task 2: You are given a string that contains only uppercase English alphabets (A-Z). Your task is to print each letter in ascending order of how many times it appears in the string. If multiple letters appear the same number of times, print them in alphabetical order.

The string can contain up to **1,000,000 characters**, so the solution must run in **linear time**. If you sort the characters in the string, it will take at least **NLogN** time.

Sample 1:

Input: DACDACCDCBC
Output: BAADDDCCCCC

Sample 2:

Input: ZXZXYYZZYYXYYZZZWWQQQQQVVVVVVVVVWXX
Output: WWWQQQQQXXXXXYYYYYYZZZZZZZVVVVVVVVV

Hint: Use `vector<pair<int, int>> freq(26, pair<int, int>(0, 0));`

In first part of pair, store letter code, in second part store the frequency, sort using following code:

```
sort(freq.begin(), freq.end(), [](const pair<int, int>& a, const pair<int, int>& b) {
      return a.second < b.second;
});
```

Print the letter as per frequency using the code in the first part of the pair.