HORIZON SCHOOL OF DIGITAL TECHNOLOGIES	**Lab 1** **Advanced Algorithm and Complexity (Python)**	**M1** **Horizon School of Digital Technologies** **2024-2025** **Rania Yangui**

Lab 1 : Complexity

Exercise 1:

We have a one-dimensional array containing uppercase letters of the alphabet (from 'A' to 'Z'). The goal is to compute the frequency of each letter in the array.

Assume an auxiliary function position(letter) is available. This function returns the position of a letter in the alphabet (e.g., position('A') returns 0, position('B') returns 1, ..., and position('Z') returns 25).

Implement two solutions to produce a frequency array:

1. A first solution that scans the array 26 times, once for each letter in the alphabet.
2. A second, more efficient solution that computes the frequency in a single scan of the array.

Exercise 2:

1. Write a Python program to find the maximum value in a set of n elements. You are only allowed to use a comparison function.
2. What is the complexity of your algorithm in terms of the number of comparisons?
3. Prove that the program is optimal.

Exercise 3:

1. Assume that we have a data structure that does not contain duplicate values.
   Propose a simple program to find the second-largest element.
2. What is its complexity in terms of the number of comparisons?

3. Rewrite your program to find the maximum as a tournament (e.g., tennis, soccer, pétanque, or any other sport).
4. In how many comparisons was the second-largest element found to be the smaller of the two compared elements?
5. Propose a new program to find the second-largest element.
6. What is its complexity in terms of the number of comparisons?