Algorithms. Exercises from Week 4

The following exercises are not about divide-and-conquer and sorting itself, but they shall demonstrate that some problems become simple, once suitable items of the instances are sorted.

1. How much time is needed to decide whether two given sets of n numbers are equal? (Assumptions: The elements are given in no particular order, but any two numbers x, y can be compared in $O(1)$ time. Comparing means to find out whether $x < y$, $x > y$ or $x = y$.) Give an algorithm that is as fast as possible.

2. Given n intervals $[s_i, f_i]$ on the real line, we wish to compute the total length covered by them. Of course, we cannot simply add their lengths $f_i - s_i$ in $O(n)$ time, as the intervals may overlap. How can you compute the covered length correctly and efficiently?

3. Given a set W of n words, each with m characters, we would like to abbreviate each word by its shortest unique prefix. (No two words in W must get the same abbreviation, and no abbreviation must be prefix of another one.) An $O(n^2m)$ time algorithm would check any two words against each other, and extend their abbreviations until a pair of distinct characters is found. Is this time bound optimal, or can we do faster?

Example:
The set of words Hydrogen, Boron, Oxygen, Fluorine, Neon, Aluminium, Phosphorus, Sulfur, Argon, Calcium, Cobalt, Nickel, Gallium, Germanium would be abbreviated H, B, O, F, Ne, Al, P, S, Ar, Ca, Co, Ni, Ga, Ge.