Université de Haute-Alsace
ENSISA

1A - IR
Année 2025/2026

Algorithm and Data Structure

TD1 : Transforming your first algorithm into code

Objectives

Understanding how to transform an algorithm to an implementation using a specific programming language.

Developing your first simple algorithm.

Comparing runtime between algorithms of different complexities solving the same problem.

Help

1. Do not use an IDE just use any text editor.
2. No need for complex compilations just use a simple `gcc`.
3. Respect coding style. I advise you to follow the advice of the GNU Foundation [1] or CS50 [2].

1 Fibonacci

1. Implement the naive algorithm for calculating the n^{th} term of the Fibonacci sequence.
2. Implement the dynamic programming algorithm for calculating such term.
3. Compare the runtime of each algorithm in **seconds**, use `clock()` [3].

2 Prime Numbers

1. Develop a naive algorithm to count the primes less than an integer n.
2. Develop a more efficient algorithm, hint : multiples of prime numbers are not prime.
3. Show that the time complexity of the previously developed algorithm of question 2. is $\mathcal{O}(n\log(\log(n))$. Hint : Use the fact that :
$$\sum_{p\ prime\,,\ p\leq n} \frac{1}{p} \leq \log(\log(n+1)) + M \tag{1}$$
where M is the is the Meissel–Mertens constant [4].
4. Implement both naive and efficient algorithms and compare their runtime in seconds.

[1] GNU : https://www.gnu.org/prep/standards/html_node/Writing-C.html

[2] CS50 : https://cs50.readthedocs.io/style/c/

[3] Documentation C : https://devdocs.io/c/chrono/clock

[4] Meissel-Mertens constant : https://fr.wikipedia.org/wiki/Constante_de_Meissel-Mertens