85301 – Algorithms and Data Structures in Biology

Lab 1 – Introduction to the Lab Module

Enrico Malizia and Riccardo Treglia

DISI
University of Bologna, Italy

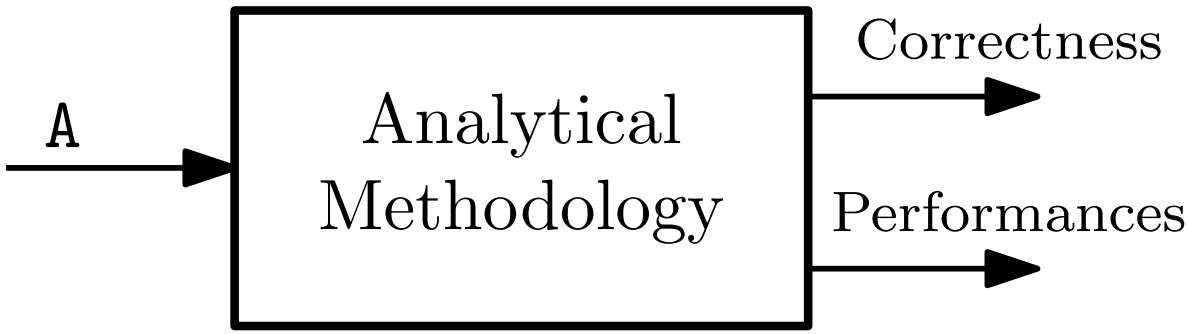
2nd semester, 2022/23

(parts of these slides are based on material by Prof. Ugo Dal Lago)

The Module's Objectives

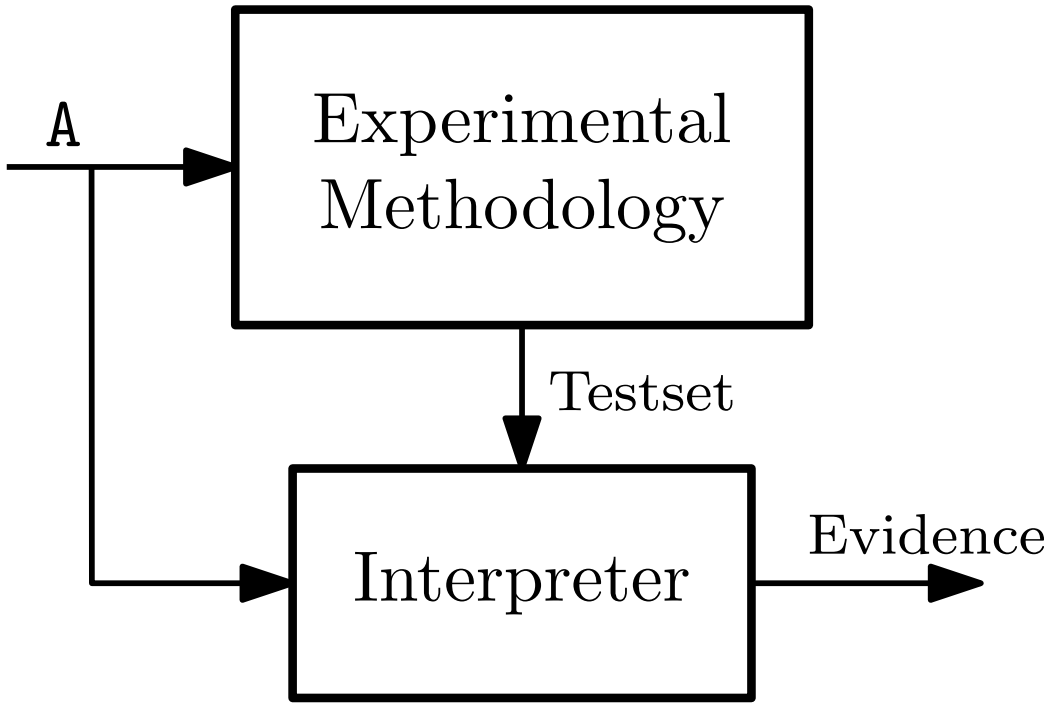
- *Two Approaches to the Evaluation of Algorithms*
  - *Analytical Approaches*
    - ★ The algorithm is evaluated against some (formal or informal) specifications, but *without* executing it
    - ★ The specification concerns either the behaviour of the algorithm or its performance (i.e., the amount of resources it consumes)
  - *Experimental Approaches*
    - ★ The algorithm is evaluated by *executing* it on a given set of input instances
    - ★ When the property of interest is qualitative (e.g., the algorithm's correctness), the outcome is just *checked* against the specification
    - ★ When the property of interest is quantitative (e.g., the algorithm's performance), the outcome is kept track of, and manipulated, perhaps using some statistical tools

Evaluating Algorithms Analytically



- **PROS**: the guarantees are often absolute; it does not require the implementation of the algorithm
- **CONS**: analysis is difficult

Evaluating Algorithms Experimentally



- **PROS**: evaluating an algorithm experimentally is typically easier, and can be done more quickly
- **CONS**: the guarantees are not absolute; experimentally evaluating an algorithm requires to implement it in a given programming language: the results are about an implementation, and not about the algorithm

Profiling Programs

- Profiling a program means running it with the aim of measuring, e.g.,
  - How much time it takes
  - How much memory it uses
  - How many times certain instructions in the programs are used
  - The frequency in which certain functions are called
- Profiling of `Python` programs can be done *by the programmer* by instrumenting their program in such a way that it keeps track of time and space consumption (e.g., explicitly measuring the execution time, or explicitly counting the number of times a function is called, etc.)
- An alternative approach consists in making use of one of the many `Python` modules which are specifically designed to help the programmer in the task of profiling; the one we will use is `cProfile`:

  `https://docs.python.org/3/library/profile.html`

Use of cProfile

Basic cProfile usage:

```
import cProfile

def fibonacci(n):
  if n <= 1:
    return n
  else:
    return fibonacci(n-2) + fibonacci(n-1)

cProfile.run('fibonacci(12)')
```

Use of cProfile

cProfile.run() output:

```
>>> cProfile.run('fibonacci(12)')

468 function calls (4 primitive calls) in 0.000 seconds

   Ordered by: standard name

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
    465/1    0.000    0.000    0.000    0.000 <stdin>:2(fibonacci)
        1    0.000    0.000    0.000    0.000 <string>:1(<module>)
        1    0.000    0.000    0.000    0.000 {built-in method builtins.exec}
        1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
```

Use of cProfile

cProfile.run() output:

```
>>> cProfile.run('fibonacci(24)')

150052 function calls (4 primitive calls)
                                        [ in 0.060 seconds

   Ordered by: standard name

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
150049/1    0.060    0.000    0.060    0.060 <stdin>:2(fibonacci)
        1    0.000    0.000    0.060    0.060 <string>:1(<module>)
        1    0.000    0.000    0.060    0.060 {built-in method builtins.exec}
        1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
```

Use of cProfile

cProfile.run() output:

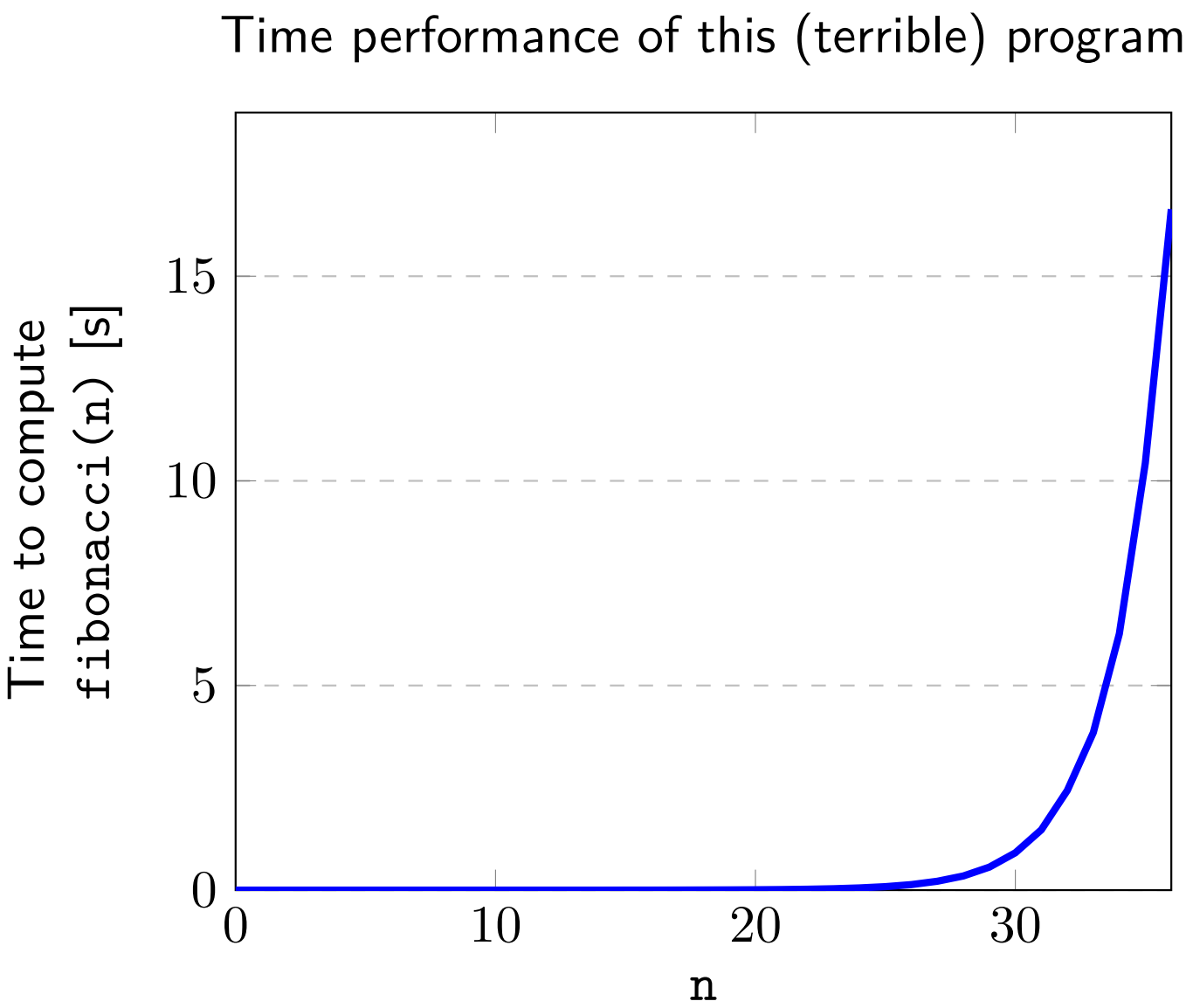
```
>>> cProfile.run('fibonacci(36)')

48315636 function calls (4 primitive calls)
                                        [ in 16.675 seconds

   Ordered by: standard name

    ncalls  tottime  percall  cumtime  percall filename:lineno(function)
48315633/1   16.675    0.000   16.675   16.675 <stdin>:2(fibonacci)
         1    0.000    0.000   16.675   16.675 <string>:1(<module>)
         1    0.000    0.000   16.675   16.675 {built-in method builtins.exec}
         1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
```

Use of `cProfile`



The LaTeX Typesetting System

- LaTeX is a very effective typesetting system to produce scientific documents
- The advantages of the latter compared to word processors are:
  - The overall typographic quality of the produced document is much better
  - LaTeX allows you to focus on the *content* of your document, leaving to the underlying compiler the task of rendering the content in the best possible way
  - Mathematics, data, and programs, are handled very satisfactorily by some predefined packages
- In LaTeX, the workflow is deeply different from the one in ordinary word processors

The LaTeX Typesetting System

- To simplify things, you can use LaTeX via your browser (in a similar way to what you would do with Google Docs)
- Register a free account on `www.overleaf.com`
- In this way there is no need to install any software on your computer
- On Overleaf you can find many useful guides; among which:
  - Creating a document in Overleaf (link)
  - "Learn LaTeX in 30 minutes" (link)
  - And many more!
- Another resource to learn LaTeX: "A (Not So) Short Introduction to LaTeX 2_ε" (link)

The LaTeX Typesetting System

```
1  \documentclass[a4paper]{article}
2
3  %%% Required
4  \usepackage[utf8]{inputenc}
5  \usepackage[T1]{fontenc}
6
7  %%% To state that the language of the document is English
8  \usepackage[english]{babel}
9
10 %%% To have a bigger area on the page to write in
11 \usepackage[margin = 1in]{geometry}
12
13 %%% To use a more modern font
14 \usepackage{lmodern}
15
16 %%% To have some more control on the author/affiliation block
17 \usepackage{authblk}
18
19 \title{An Example Document in \LaTeX}
20
21 \author{Enrico Malizia}
22 \author{Riccardo Treglia}
23 \affil{DISI, University of Bologna, Italy}
24
25 \date{10 March 2022}
```

The LaTeX Typesetting System

```
27 \begin{document}
28
29 \maketitle
30
31 \section{Introduction}
32 This is an example of introduction.
33
34 \section{Body}\label{sec_body}
35 It is easy to write nice mathematics.
36 For example:
37 \[
38 \sum_{i=1}^n i = \frac{n(n+1)}{2}; \qquad \int x^2 dx = \frac{x^3}{3} + c
39 \]
40
41 \section{Conclusion}
42 In Section~\ref{sec_body} we have seen how to write some mathematical formulas.
43 In this section, we conclude blah blah\dots
44
45 \end{document}
46
```

The LaTeX Typesetting System

An Example Document in LaTeX

Enrico Malizia and Riccardo Treglia

DISI, University of Bologna, Italy

10 March 2022

1 Introduction

This is an example of introduction.

2 Body

It is easy to write nice mathematics. For example:

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}; \qquad \int x^2 dx = \frac{x^3}{3} + c$$

3 Conclusion

In Section 2 we have seen how to write some mathematical formulas. In this section, we conclude blah blah...