# Algorithms and Data Structures for Biology
# **16 April 2019 — Lab Session**

Ugo Dal Lago        Thomas Leventis

This assignment will be the first one to be marked. Please carefully read the instruction below before starting to work on the assignment

## 1 Instructions

This assignment must be completed individually, by May 1st 2019 at 23.59 CET. To complete the assignment, you need to send the following to the email addresses `ugo.dallago@unibo.it` and `thomas.leventis@irif.fr`:

- One Python file named `FIRSTNAME_LASTNAME.py`, with the code you have developed.
- One PDF file name `FIRSTNAME_LASTNAME.pdf`, preferably produced by way of LaTeX, including a description of the algorithms you designed, the Python code you wrote, and the experimental results.

The email should have the following subject: "`[ADSB] Assignment I`".

## 2 The Problem

We want to deal with the following problem. Suppose you are a biologist who will soon move to the franco-italian Antartic Station[1]. The Station is not covered by broadband Internet, and so most of the needed data must be brought with you. You can bring with you only one hard disk with a capacity of $n$ gigabytes. Inside the hard disk you need to fit as many of your databases as possible. You are currently working with $m$ different databases, call them $1, \ldots, m$, with database $j$ having a size equal to $S_j$ megabytes. Please notice that $\sum_j S_j$ is much bigger than $n$ gigabtyes How should you decide which ones of the databases you should bring with you? You first attribute to each database a measure $U_j$ of the utility the database has for you. Then, you realise you want to bring with you a set of databases with maximal total quality among those which fits into your hard disk.

This assignment asks you to:

- model the problem described above as a combinatorial optimization problem, abstracting away from unessential details.
- give an exhaustive search algorithm solving the combinatorial problem at the previous point; the key idea here consists in finding an appropriate way to define the search space, in such a way that exploring it turns out to be easy.
- analyse the algorithm complexity, expressing it as a function of $m$, the total number of databases.

[1] `http://www.isac.cnr.it/en/infrastructures/concordia-antarctic-station`, for those of you who are interested

- Design an improved version of your algorithm, in the spirit of the so-called branch-and-bound technique. In other words, the algorithm should build candidate solutions for the problem incrementally, this way avoiding the analysis of many other ones.
- Implement the two algorithms you designed in Python.
- Design and implement a testing routine which tests your two algorithms on randomly generated inputs in which $m$ is $5, 10, 15, 20, 25$, each of the $S_j$ and $U_j$ are random numbers between 1 and 10, and $n$ is $7m$.
- and finally use `cProfile` to experimentally test the runtime of your program, and give a graphical presentation of your results. Check the file `plotting.tex` to see how to draw graphs in LaTeX.