Computer Programming 30509

Introduction to the course
A.Y. 2022/23

Bocconi

Instructors

- Luca Saglietti

- Luca Silva (TA)

- Office hours: I could fix official ones,
  but it is best (and more flexible) if you just send me an email and we arrange a meeting
- Also: USE THE FORUM! (everybody benefits)

Prerequisites

- **I'll assume basic but solid knowledge of Python (basically the Think Python book, more or less)**

Tools

- We will use Python 3 (I suggest Python 3.6 at least)
- We will also use some scientific libraries (numpy, scipy, matplotlib)
- We will use the Spyder Integrated Development Environment (IDE)
- All of this comes bundled in the Anaconda distribution
- During practice sessions, bring your laptop and follow along

Course organization

- There will be a programming part and a theory part.
- They will be tied and proceed in parallel
- The theory part is (mostly) intended to introduce concepts that we will then put in practice into code
- There might be some tangential topics covered

This course is still work-in-progress

- **The course has been (and will be) continually adjusted**
  - Based on students' feedback and our observations

Prof. Carlo Baldassi

- **Feedback/constructive criticism is very welcome!**

Course objectives

- **The idea is to learn some more advanced programming patterns/techniques/algorithms, and keep gaining familiarity**
  - Side objective: learn to navigate documentation
- **Also, to cover topics that fit into the course of studies, and prepare for the following semester courses (e.g. machine learning)**
- **Also, to make the topics as interesting/relevant as possible**
- **Also, to have a somewhat coherent framework**

Course topics

- **The main idea is to cover several families of optimization problems, and algorithmic techniques related to them**
- **Then, we will use those as an excuse to learn more advanced programming**
- **Broadly, 4 main topics**
  - Monte Carlo, Simulated Annealing
  - Graph algorithms and Dynamic programming
  - Non-linear optimization, Gradient descent
  - A few selected Data structures

Course organization (II)

- For each of the first 3 main topics, we will have the theory part first
- Then we will have the programming part related to that (both generic algorithms and examples)
- The first two practice sessions will be a bit special because we need some foundations (learn Python's numeric/scientific libraries)

Practice sessions organization

- Mostly I will do live-coding explaining what I do and why and how
- You're expected to follow along
- You're expected to ask questions (this is not the time be shy or passive)
- We will also do exercises in class
- I'll give exercises for you to do on your own
- Feedback is welcome

Exams

- The exam will consist in programming exercises (coding) with some open-ended more theoretically-oriented questions
- Single exam, 0 to 31 grade, passing grade is 18, limited amount of renormalization
- You'll have access to a mock exam and past exams
- More about this during the last lecture

Some general advice

- **Practice, practice, practice**
- **The basics (basic Python + first few lectures) must be rock-solid**
- **Don't fall behind. There is no midterm, you'll need to discipline yourselves**
- **Experiment as much as you can with code**
- **Be curious. Always investigate anything you don't understand 100%. Feel free to ask (lectures, forum, office hours...)**
- **Never look at the solution of an exercise before having given your best**

Questions before we start?