Algorithmics and Programming II: Introduction

Jordi Cortadella and Jordi Petit
Department of Computer Science
• Lecturers:
  – Jordi Cortadella (jordi.cortadella@upc.edu)
  – Emma Rollón (erollon@cs.upc.edu)
  – Jordi Petit (jordi.petit-silvestre@cs.upc.edu)

• Sessions:
  – Theory & (Jordi C.)
  – Lab (Emma & Jordi P.)
Material

• Slides, exercises:

   https://www.cs.upc.edu/~jordicf/Teaching/AP2

• Jutge (for lab sessions):

   https://jutge.org

• Lliçons (by J. Petit and S. Roura):

   https://lliçons.jutge.org
The lectures of the course are combined with practical programming sessions using a virtual learning environment for computer programming (Jutge.org).

### Lectures

<table>
<thead>
<tr>
<th>Lecture</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction</td>
<td></td>
</tr>
<tr>
<td>2. Abstract Data Types</td>
<td></td>
</tr>
<tr>
<td>3. Algorithm Analysis</td>
<td></td>
</tr>
<tr>
<td>4. Divide and Conquer</td>
<td></td>
</tr>
<tr>
<td>5. Memory Management</td>
<td></td>
</tr>
<tr>
<td>6. Containers: Stack</td>
<td></td>
</tr>
<tr>
<td>7. Containers: Queue and List</td>
<td></td>
</tr>
<tr>
<td>8. Containers: Priority Queue</td>
<td></td>
</tr>
<tr>
<td>9. Graphs: Connectivity</td>
<td></td>
</tr>
<tr>
<td>11. Trees</td>
<td></td>
</tr>
<tr>
<td>12. Containers: Set and Dictionary</td>
<td></td>
</tr>
<tr>
<td>13. Hashing</td>
<td></td>
</tr>
<tr>
<td>14. Fast Fourier Transform</td>
<td></td>
</tr>
<tr>
<td>15. Cryptography</td>
<td></td>
</tr>
</tbody>
</table>
Evaluation

• Evaluation items:
  – Projects (Proj), Parcial Lab (PLab),
    Final Theory (FTh), Final (FLab).

• Grading:
  – $N_1 = 0.2 \text{ Proj} + 0.25 \text{ PLab} + 0.25 \text{ FLab} + 0.3 \text{ FTh}$
  – $N_2 = 0.2 \text{ Proj} + 0.4 \text{ FLab} + 0.4 \text{ FTh}$
  – $N = \max(N_1, N_2)$
First project: Containers

• Design a class to manage containers.
• Language: Python.
Peer and self assessment

• The project will be evaluated by the students themselves.

• Each project will be evaluated by three students. The grade will be calculated as the average grade given by the students.

• The evaluation will be completely blind.

• Biased evaluations will be detected and penalized.

• Each student will have the right to request the evaluation by the professor (who can upgrade or downgrade the evaluation given by the students).
• “Programming is an art of telling another human what one wants the computer to do.”

• “An algorithm must be seen to be believed.”

• “The real problem is that programmers have spent far too much time worrying about efficiency in the wrong places and at the wrong times; premature optimization is the root of all evil (or at least most of it) in programming.”
Street map represented as a graph

Second Project: GPS

Telegram

Language: Python
Objective of the course

Confronting large and difficult problems. How?

- Skills for abstraction and algorithmic reasoning.
- Design and use of complex data structures.
- Techniques for complexity analysis.
- Methodologies for modular programming.
- High-quality code.
Compute the convex hull of $n$ given points in the plane.
The Closest-Points problem

- **Input:** A list of $n$ points in the plane
  \[ (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \]
- **Output:** The pair of closest points
- **Simple approach:** check all pairs $\rightarrow O(n^2)$
- We want an $O(n \log n)$ solution!
Navigation: find the shortest path
How to encrypt messages?

1. This file is written in plaintext
2. Encryption Algorithm
   - Key
   - Output: 2q9733i1o79c4ffsfsf24yu3
3. Decryption Algorithm
   - Key
4. This file is written in plaintext
The secret: training, training, training, training ...
... up to the finish line