Algorithmics and Programming II: Introduction

Jordi Cortadella and Jordi Petit
Department of Computer Science

Algorithmics and Programming II

- Lecturers:
  - Jordi Cortadella (jordi.cortadella@upc.edu)
  - Jordi Petit (jordi.petit-silvestre@cs.upc.edu)
  - Pau Fernández (pau.fernandez@upc.edu)
- Teaching Assistant:
  - Laia Mogas (laia.mogas.pladevall@estudiantat.upc.edu)
- Sessions:
  - Theory (Jordi Cortadella)
  - Lab (1.1-1.2: Jordi Petit; 1.3: Pau Fernández)

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

Slides of the lectures

Algorithmics and Programming II

This material is used for the course Algorithmics and Programming II of the Degree Data Science and Engineering.

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

Lectures

1. Introduction	
2. Abstract Data Types (I)	
3. Abstract Data Types (II)	
4. Algorithm Analysis (I)	

Evaluation

- Evaluation items:
  - Projects: P1, P2
  - Exams: Partial Lab (PL), Final Theory (FT), Final Lab (FL)
- Grading:
  - Projects: $P = (P1 + P2)/2$
  - Exams: $X = \max \begin{cases} 0.3\ PL + 0.4\ FT + 0.3\ FL \\ 0.5\ FT + 0.5\ FL \end{cases}$
  - Final grade:

$$N = \begin{cases} \max(0.3\ P + 0.7\ X, 0.15\ P + 0.85\ X) & \text{if } X \geq 4 \\ 0.15\ P + 0.85\ X & \text{if } X < 4 \end{cases}$$

First project: Cities and Roads

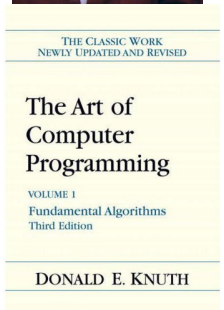
Inspired by the game Catan™.

Peer and self assessment

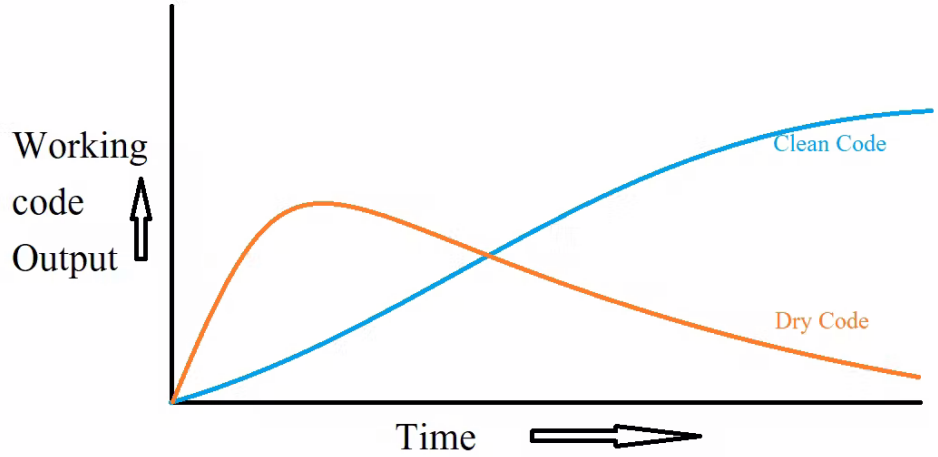
- The project will be evaluated by the students themselves.
- Each project will be evaluated by three students. The grade will be calculated as the average grade given by the students.
- The evaluation will be completely blind.
- Biased evaluations will be detected and penalized.
- Each student will have the right to request the evaluation by the professor (who can upgrade or downgrade the evaluation given by the students).

Donald Knuth (Turing award, 1974)

- "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."



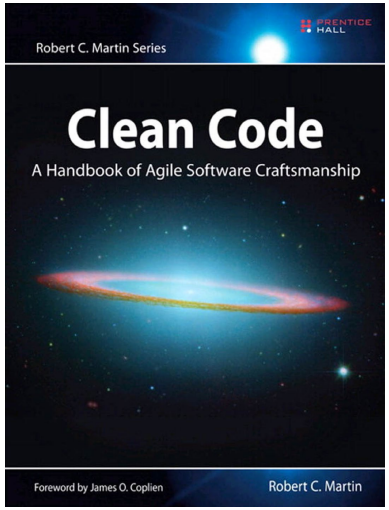
Clean Code Vs Quick Code



Clean Code: Good Practices P1

Must for juniors and seniors to know about "Clean Code"

Rahul Singh · Feb 1, 2023



"Truth can only be found in one place: the code."

"Indeed, the ratio of time spent reading versus writing is well over 10 to 1. We are constantly reading old code as part of the effort to write new code. ...[Therefore,] making it easy to read makes it easier to write."

"It is not enough for code to work."

"So if you want to go fast, if you want to get done quickly, if you want your code to be easy to write, make it easy to read."

"You should name a variable using the same care with which you name a first-born child."

"Clean code always looks like it was written by someone who cares."

"Of course bad code can be cleaned up. But it's very expensive."

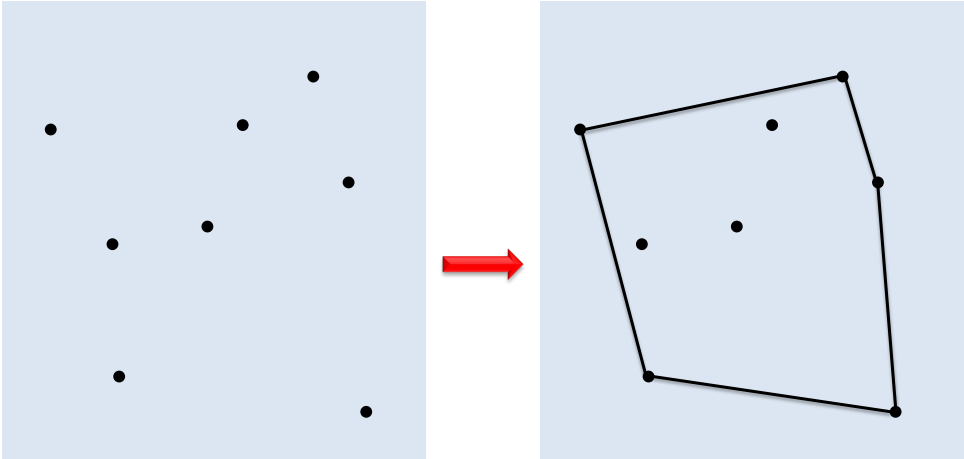
Second project: inferring graphs from GPS routes

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.

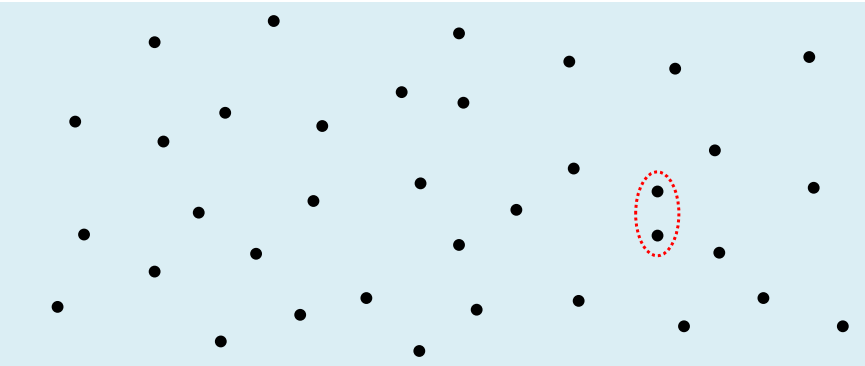
Problems on polygons



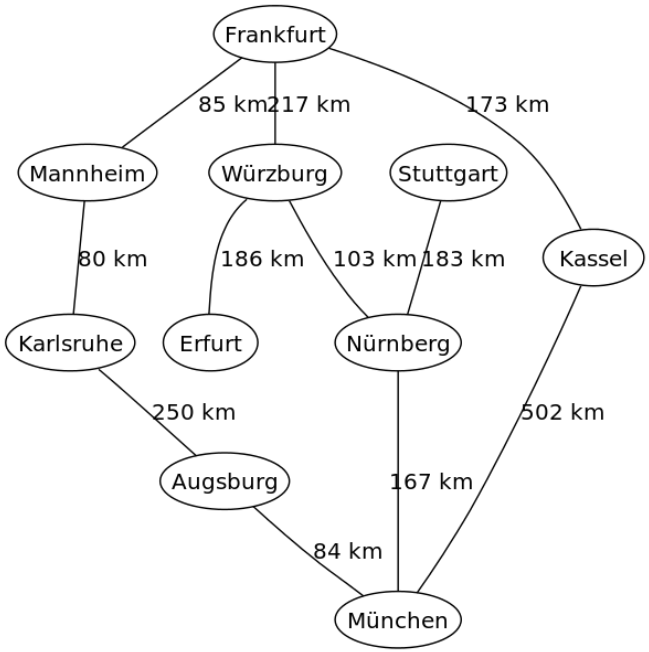
Compute the convex hull of n given points in the plane.

The Closest-Points problem

- **Input:** A list of n points in the plane
  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$
- **Output:** The pair of closest points
- **Simple approach:** check all pairs → $O(n^2)$
- We want an $O(n \log n)$ solution !



Navigation: find the shortest path



How to encrypt messages?