UCLouvain

linfo1103

2026

Introduction to algorithms

The version you're consulting is not final. This course description may change. The final version will be published on 1st June.

5.00 credits	30.0 h + 30.0 h	Q2

Teacher(s)	Dupont Pierre ;
Language :	French
Place of the course	Louvain-la-Neuve
Prerequisites	This course assumes that the student already masters the basics of programming covered by the course LINFO1101.
Main themes	• Design and implementation of iterative or recursive algorithms: path, counting, sorting, searching in collections • Computational complexity • Basic data structures: arrays, stacks, queues, linked lists • Recursive data structures: tree structures, binary search trees • Invariants
Learning outcomes	**At the end of this learning unit, the student is able to :** Given the learning outcomes of the "Bachelor in Computer science" program, this course contributes to the development, acquisition and evaluation of the following learning outcomes: • S1.I2, S1.I3, S1.G1 • S2.2, S2.3, S2.4 • S5.5 Students who have successfully completed this course will be able to: • justify a choice between several algorithmic solutions to solve a given problem; • analyze algorithms, iterative or recursive, to represent and manipulate collections and to propose variants thereof; • choose, design and use data structures, including recursive; • give a reasoned estimate of the time complexity of iterative algorithms and the spatial complexity of data structures; • reasoning about properties of algorithms or data structures in terms of invariants. Students will have developed methodological and operational skills. In particular, they have developed their ability to: • to take a critical look and make a reasoned analysis of a solution or set of solutions that could be made to a given problem by setting quality criteria; • realize small programs using conventional algorithms and data structures.

Evaluation methods	**Computation of the overall course grade** A PARTICIPATION grade reflects the involvement of the student during the year for solving the practice problems on Inginious and when implementing the final project. It is calculated at the end of the semester (week 13) on the basis of all these activities, to be realized **weekly** according to the calendar available on Moodle. In the first session, the participation grade is worth 20% of the overall score + 80% for the final exam. The participation grade can not be reassessed. In the second session, participation grade and the final exam are worth respectively 10 % and 90% of the overall score. The (closed book) final exam is a written exam on a computer or, when appropriate, on paper. **Rules for student collaboration and use of external resources, including generative AIs** Active collaboration between students is encouraged during practical work sessions and via an exchange forum on Moodle. Each student is expected to submit a personal solution to the final project. The use of public resources (e.g. stackoverflow.com), including generative AIs (e.g. chatGPT) is permitted, as long as each (fragment of) code submitted by the student **specifically** mentions all the resources used. The distribution or exchange between students of (fragments of) code is not authorized by any means (GitHub, Facebook, Discord, etc.), even after the project deadline. Failure to comply with these rules for the project will result in an overall participation grade of 0. The final computer exam must be carried out without access to any external resources. These rules are explained in detail during the first class (see course Moodle site).
Teaching methods	• Lectures • Practical sessions on the Inginious server • One project at the end of the semester
Content	Algorithmics is concerned with solving problems by implementing sequences of elementary operations according to a predefined process or procedure leading to a solution. This discipline is both abstract and put into practice through programs (e.g. implemented in Python) and run on a computer. • Time and space complexity • Search algorithms through arrays • Abstract data types, data structures: stacks, queues, dynamic arrays, linked lists • Sorting algorithms • Recursion • Recursive data types • Computaional complexity of recursive algorithms, recurrence equations • Binary trees and dictionnaries • Invariants
Inline resources	moodle.uclouvain.be/course/view.php?id=1916
Bibliography	Il n'y a pas d'ouvrage de référence obligatoire mais, à titre complémentaire, des ouvrages sont recommandés sur le site Moodle.
Faculty or entity in charge	INFO

Programmes containing this learning unit (UE)

Program title	Acronym	Credits	Prerequisite	Learning outcomes
Master [120] in Data Science : Statistic	DATS2M	5		
Master [120] in Linguistics	LING2M	5		
Additionnal module in Geography	APPGEOG	5		
Bachelor in Mathematics	MATH1BA	5		
Bachelor in Computer Science	SINF1BA	5		
Minor in Computer Sciences	MINSINF	5		
Minor in Statistics, Actuarial Sciences and Data Sciences	MINSTAT	5		
Master [120] of Education, Section 4 : Mathematics	MATH2M4	5		
Minor in mathematics teaching	APPENSMAT	5		