lsinc1121 — Algorithms and data structure

2025

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	Q1

This learning unit is not open to incoming exchange students!

Language :	French
Place of the course	Charleroi
Prerequisites	This course assumes the mastery of programming and program design in an object-oriented language such as Java, knowledge of elementary data structures and notions of recursion and computational complexity as targeted by the course LEPL1402. *The prerequisite(s) for this Teaching Unit (Unité d'enseignement – UE) for the programmes/courses that offer this Teaching Unit are specified at the end of this sheet.* *The prerequisite(s) for this Teaching Unit (Unité d'enseignement – UE) for the programmes/courses that offer this Teaching Unit are specified at the end of this sheet.*
Main themes	• Complexity measures of an algorithm and complexity analysis methods. • Dichotomic sorting and search algorithms. • Basic data structures (lists, trees, binary search trees): study of their abstract properties, their concrete representations, their application and the main algorithms that manipulate them. • Advanced data structures (union-find, hash tables, heaps, balanced binary trees, graph representation and manipulation, textual data processing, dictionaries).
Learning outcomes	**At the end of this learning unit, the student is able to :** Given the learning outcomes of the "Bachelor in Engineering" program, this course contributes to the development, acquisition and evaluation of the following learning outcomes: • AA1.1, AA1.2 • AA2.4, AA2.5, AA2.7 • AA3.2 • AA4.3 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.I1, S1.I3 • S2.2, S2.3, S2.4 • S4.3 • S5.4 • S6.1, S6.3 Students who have successfully completed this course will be able to : • make an informed choice on the use of the main data structures used to represent collections, • make good use of existing algorithms to manipulate these data structures and analyze their performance, • design and implement variants of the algorithms studied, • test algorithms and data structures, • make good use of algorithms and data structures documented in an API • abstract, model and implement effective solutions to algorithmic puzzle problems. **Students will have developed methodological and operational skills. In particular, they will have developed their ability to:** • analyze critically a problem, • to test and debug algorithmic programs, • effectively implement short but non-trivial algorithms.learn for themselves in a reference book and in the complementary technical documentation

Evaluation methods	For the exam, the students will have to program tasks on inginious https://inginious.info.ucl.ac.be followed by an optional discussion with the teacher as a regular oral exam in case the student does not think the ingininious score reflects his expertise in the course. One mid-term quizz quizz might be proposed on two points during smart week but it can only impact positively on your final grade. One participation grade will also be taken into account for two points for the first session, but not for the second session.
Teaching methods	The active pedagogy method followed in this course is inspired by reverse classes. There are six two-week modules. Each module includes an introductory course to the subject, theoretical exercises to prepare, chapters from the reference book to read, a practical work on correcting exercises in the middle of the model, work on inginious to be carried out (Java programs) and finally a restructuring course at the end of the module. One of the essential components of this pedagogy consists in making each student learn by himself. The success of the learning process therefore presupposes a significant involvement of each student. The actual learning remains the responsibility of each student. To pass the exam it is imperative that the student programs regularly.
Content	• Computational complexity, • Trees, binary search trees, • Balanced trees, • Dictionaries and hash tables, • Priority queues and heaps • Graphs, • Text processing (pattern matching, compression algorithms)
Inline resources	https://moodleucl.uclouvain.be/course/view.php?id=7682 + questions on the course website (reachable from Moodle).
Bibliography	Livre obligatoire: Algorithms, 4th Edition by Robert Sedgewick and Kevin Wayne, Addison-Wesley Professional. ISBN-13: 978-0321573513 ISBN-10: 032157351X Et plus généralement les documents (énoncés des missions, conseils pour l'examen, ...) disponibles sur : http://moodleucl.uclouvain.be/course/view.php?id=7682
Other infos	Background: • master an object-oriented programming language (p.e. Java) • know an use correctly basic data structures (stacks, queues, lists, etc..) • have basic knowledge of recursion and computational complexity. This background is part of the content of LEPL1401 and LEPL1402.
Faculty or entity in charge	SINC

Programmes containing this learning unit (UE)				
Program title	Acronym	Credits	Prerequisite	Learning outcomes
Bachelor in Computer Science	SINC1BA	5	LSINC1402	🔍