

Introduction to algorithms

5.00 credits

30.0 h + 30.0 h

Q2

Teacher(s)	Derval Guillaume (compensates Dupont Pierre) ;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: \$1.12, \$1.13 \$2.2.4 \$6.2 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. 					

IndicatesIndicatesIn IndicatesInterventionIt is calculated at the end of the semester (week 13) on the basis of all these activities. In the first session, the participation grade is worth 20% of the final grade + 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 ovisore. 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 Active collaboration between students is encouraged during practical work sessions and via an exchange for on Moodle. Each student is expected to submit a personal solution to the final project. The use of public resources (stackoverflow.com), including generative AIs (e.g., chatGPT) is permitted, as long as each (fragment of) or stackoverflow.com), including generative AIs (e.g., chatGPT) is permitted, as long as each (fragment of) or stackoverflow.com), including generative AIs (e.g., chatGPT) is permitted, as long as each (fragment of) or stackoverflow.com), including generative AIs (e.g., chatGPT) is permitted, as long as each (fragment of) or stackoverflow.com), including generative AIs (e.g., chatGPT) is permitted, as long as each (fragment of) or stackoverflow.com), including generative AIs (e.g., chatGPT) is permitted, as long as each (fragment of) or The distribution or exchange between students of (fragments of) code is not authorized by any means (Gith Facebook, Discord, etc.), even after 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• Ledures • Practical sessions on the loginio		
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. In the first session, the participation grade is worth 20% of the final grade + 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 ow 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 Active collaboration between students is encouraged during practical work sessions and via an exchange for on Moodle. Each student is expected to submit a personal solution to the final project. The use of public resources (stackoverflow.com), including generative AIs (e.g. chatCPT) is permitted, as long as each (fragment of c submitted by the student mentions all the resources used. The distribution or exchange between students of (fragments 0) code is not authorized by any means (Gitf Facebock, 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 Peractical sessions on the Inginious server One project at the end of the semester One project at	Evaluation methods	Computation of the overall course grade
In the first session, the participation grade is worth 20% of the final grade + 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 ow 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 Active collaboration between students is encouraged during practical work sessions and via an exchange for on Moodle. Each student is expected to submit a personal solution to the final project. The use of public resources (submitted by the student memotions all the resources used. The distribution or exchange between students of (fragments of) code is not authorized by any means (Gith 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 assistions 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 accore to a predefined process or procedure leading to a solution. This discipline is both abstract and put i		
In the second session, participation grade and the final exam are worth respectively 10 % and 90% of the own 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 Active collaboration between students is encouraged during practical work sessions and via an exchange for on Moodle. Each student is expected to submit a personal solution to the final project. The use of public resources (submited by the student mentions all the resources used. The distribution or exchange between students of (fragments of) code is not authorized by any means (Gith 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 •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 accor 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 rur a computer. •Time and space complexity Search algorithms		
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 Active collaboration between students is encouraged during practical work sessions and via an exchange fo on Moodle. Each student is expected to submit a personal solution to the final project. The use of public resources (stackoverflow.com), including generative AIs (e.g. chaGPT) is permitted, as long as each (fragment of) o submitted by the student mentions all the resources used. The distribution or exchange between students of (fragments of) code is not authorized by any means (Gitt 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 • Lectures • One project at the end of the semester Content Algorithmics is concerned with solving problems by implementing sequences of elementary operations accord to a predefined process or procedure leading to a solution. This disciplication is that a structures: stacks, queues, dynamic arrays, linked lists Sorting algorithms • Sorting algorithms • Algorithmics is concerned with solving problems, recurrence equations • Time and space complex		
Rules for student collaboration and use of external resources Active collaboration between students is encouraged during practical work sessions and via an exchange for on Moodle. Each student is expected to submit a personal solution to the final project. The use of public resources (stackoverflow.com), including generative AIs (e.g., cha(GPT) is permitted, as long as each (fragment of c submitted by the student mentions all the resources used. The distribution or exchange between students of (fragments of) code is not authorized by any means (Gitt Facebook, Discord, etc.), even after the project deadline. Failure to comply with these rules for the project deadline. Failure to computer exam must be carried out without access to any external resources. The end is essions 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 accom 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 rur a computer. • Time and apace complexity • Search algorithms through arrays • Abstract data types, data structures: stacks, queues, dynamic arrays, linked lists • Sorting algorithms • Computer is and dictionnaries • Novariants Inline resources		
Active collaboration between students is encouraged during practical work sessions and via an exchange for on Moodle. Each student is expected to submit a personal solution to the final project. The use of public resources (stackoverflow.com), including generative AIs (e.g. cha(GPT) is permitted, as long as each (fragment of) or submitted by the student mentions all the resources used. The distribution or exchange between students of (fragments of) code is not authorized by any means (Gith Facebook, Discord, etc.), even after 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 accord 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 rule a computer. •Time and space complexity •Search algorithms •Recursion •Recursion •Recursion •Recursive data types •Content Algorithmics is doncerned with solving problems by implementing sequences of elementary operations accord to a predefined process or procedure leading to a solution. <td< td=""><td></td><td>The (closed book) final exam is a written exam on a computer or, when appropriate, on paper.</td></td<>		The (closed book) final exam is a written exam on a computer or, when appropriate, on paper.
on Moodle. Each student is expected to submit a personal solution to the final project. The use of public resources (stackoverflow.com), including generative AIs (e.g. chatGPT) is permitted, as long as each (fragment of) or submitted by the student mentions all the resources used. The distribution or exchange between students of (fragments of) code is not authorized by any means (Gitt 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 accord 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 rur a computer. • Time and space complexity • Search algorithms • Search algorithms • Recursion • Recursion • Necursion • Recursion the secursice algorithms, recurrence equations • Dinaryt rees and dictionnaries • Invariants Inline resources moodle.uclouvain.be/course/view.php?id=1916		Rules for student collaboration and use of external resources
stackoverflow.com), including generative Als (e.g. chatGPT) is permitted, as long as each (fragment of) of submitted by the student mentions all the resources used. The distribution or exchange between students of (fragments of) code is not authorized by any means (Gith 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 accord 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 rur a computer. • Time and space complexity • Search algorithms through arrays • Abstract data types, data structures: stacks, queues, dynamic arrays, linked lists • Sorting algorithms • Recursive data types • Computational complexity of recursive algorithms, recurrence equations • Binary trees and dictionnaries • Invariants Inline resources moodle.uclouvain.be/course/view.php?id=1916		Active collaboration between students is encouraged during practical work sessions and via an exchange forum on Moodle.
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 accome 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 a computer. • Time and space complexity • Search algorithms through arrays • Abstract data types, data structures: stacks, queues, dynamic arrays, linked lists • Sorting algorithms • Recursive data types • Computational complexity of recursive algorithms, recurrence equations • Binary trees and dictionnaries • Invariants		Each student is expected to submit a personal solution to the final project. The use of public resources (e.g. stackoverflow.com), including generative Als (e.g. chatGPT) is permitted, as long as each (fragment of) code submitted by the student mentions all the resources used.
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 accord 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 a computer. Time and space complexity • Search algorithms through arrays • Abstract data types, data structures: stacks, queues, dynamic arrays, linked lists • Sorting algorithms • Recursion • Recursion • Recursive data types • Computational complexity of recursive algorithms, recurrence equations • Invariants Inline resources moodle.uclouvain.be/course/view.php?id=1916		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.
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 accord 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 rur 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 • Computational complexity of recursive algorithms, recurrence equations • Binary trees and dictionnaries • Invariants Inline resources moodle.uclouvain.be/course/view.php?id=1916		Failure to comply with these rules for the project will result in an overall participation grade of 0.
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 accord 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 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 • Computational complexity of recursive algorithms, recurrence equations • Binary trees and dictionnaries • Invariants Inline resources moodle.uclouvain.be/course/view.php?id=1916		The final computer exam must be carried out without access to any external resources.
• 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 accord 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 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 • Computational complexity of recursive algorithms, recurrence equations • Binary trees and dictionnaries • Invariants		These rules are explained in detail during the first class (see course Moodle site).
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 rur 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	Teaching methods	Practical sessions on the Inginious server
This discipline is both abstract and put into practice through programs (e.g. implemented in Python) and rur 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	Content	Algorithmics is concerned with solving problems by implementing sequences of elementary operations according to a predefined process or procedure leading to a solution.
 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 Invariants In n'y a pas d'ouvrage de référence obligatoire mais, à titre complémentaire, des ouvrages sont recommandée		This discipline is both abstract and put into practice through programs (e.g. implemented in Python) and run on
 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 Invariants In n'y a pas d'ouvrage de référence obligatoire mais, à titre complémentaire, des ouvrages sont recommandée		Time and space complexity
 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 II n'y a pas d'ouvrage de référence obligatoire mais, à titre complémentaire, des ouvrages sont recommandée		
• Recursion • Recursion • Recursive data types • Computaional complexity of recursive algorithms, recurrence equations • Binary trees and dictionnaries • Invariants Inline resources Inline resources II n'y a pas d'ouvrage de référence obligatoire mais, à titre complémentaire, des ouvrages sont recommandée		
• 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 II n'y a pas d'ouvrage de référence obligatoire mais, à titre complémentaire, des ouvrages sont recommandée		8 8
Computaional complexity of recursive algorithms, recurrence equations Binary trees and dictionnaries Invariants Inline resources Invariants Invari		
Binary trees and dictionnaries Invariants Inline resources Invariants Invatta Invariants Invariants Invariants Invariants Invariants		
Invariants Inline resources II n'y a pas d'ouvrage de référence obligatoire mais, à titre complémentaire, des ouvrages sont recommandé		
Il n'y a pas d'ouvrage de référence obligatoire mais, à titre complémentaire, des ouvrages sont recommandé		
	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 su le site Moodle.
Faculty or entity in INFO	Faculty or entity in	INFO
charge		

Programmes containing this learning unit (UE)							
Program title	Acronym	Credits	Prerequisite	Learning outcomes			
Master [120] in Linguistics	LING2M	5		٩			
Additionnal module in Geography	APPGEOG	5		٩			
Bachelor in Mathematics	MATH1BA	5		٩			
Bachelor in Computer Science	SINF1BA	5		٩			
Minor in Computer Sciences	MINSINF	5		٩			