

5 crédits	30.0 h + 30.0 h	Q1
-----------	-----------------	----

Enseignants	Schaus Pierre ;
Langue d'enseignement	Français
Lieu du cours	Louvain-la-Neuve
Préalables	<ul style="list-style-type: none"> <li>• Maîtriser la programmation dans un langage orienté-objet tel que Java</li> <li>• Connaître et utiliser correctement de structures de données élémentaires (piles, files, listes, etc.)</li> <li>• Avoir des notions en matière de récursion et de complexité calculatoire.</li> </ul>
Thèmes abordés	<ul style="list-style-type: none"> <li>• Mesures de complexité d'un algorithme et méthodes d'analyse de complexité.</li> <li>• Algorithmes de tris et recherche dichotomiques</li> <li>• Structures de données de base (listes, arbres, arbres binaires de recherche) : étude de leurs propriétés abstraites, de leurs représentations concrètes, de leur application et des principaux algorithmes qui les manipulent.</li> <li>• Structures de données avancées (union-find, tables de hachage, tas, arbres binaires équilibrés, représentation et manipulation de graphes, traitement de données textuelles, dictionnaires).</li> </ul>
Acquis d'apprentissage	<p><b>Les étudiants ayant suivi avec fruit ce cours seront capables de</b></p> <ul style="list-style-type: none"> <li>- faire un choix argumenté sur l'utilisation des principales structures de données utilisées pour représenter des collections,</li> <li>- utiliser à bon escient les algorithmes existants pour manipuler ces structures de données et analyser leur performance,</li> <li>- concevoir et mettre en oeuvre des variantes des algorithmes étudiés,</li> <li>- tester des algorithmes et des structures de données,</li> <li>- utiliser à bon escient les algorithmes et structures de données documentées dans une l'API ,</li> <li>1 - abstraire, modéliser et d'implémenter des solutions efficaces à des problèmes de type « puzzle » algorithmiques.</li> </ul> <p>Les étudiants auront développé des compétences méthodologiques et opérationnelles. En particulier, ils auront développé leur capacité à :</p> <ul style="list-style-type: none"> <li>- analyser de façon critique un problème posé,</li> <li>- tester et debugger des programmes algorithmiques,</li> <li>- implémenter efficacement des algorithmes courts mais non triviaux.</li> <li>- apprendre par eux-mêmes dans un ouvrage de référence et dans la documentation technique complémentaire</li> </ul> <p>-----                  La contribution de cette UE au développement et à la maîtrise des compétences et acquis du (des) programme(s) est accessible à la fin de cette fiche, dans la partie « Programmes/formations proposant cette unité d'enseignement (UE) ».</p>
Modes d'évaluation des acquis des étudiants	<p><b>Examen sur ordinateur à l'aide d'Inginious <a href="https://inginius.info.ucl.ac.be">https://inginius.info.ucl.ac.be</a>.</b></p> <p><b>Un quizz sur deux points peut être organisé lors de la semaine smart et ne compte dans la note de l'étudiant uniquement si il fait remonter celle-ci.</b></p>
Méthodes d'enseignement	<p>La méthode de pédagogie active suivie dans ce cours est inspirée des classes inversée. Il y a six modules de deux semaines. Chaque module comporte un cours d'introduction à la matière, des exercices théoriques à préparer, des chapitres du livre de référence à lire, un TP de correction des exercices en milieu de module, des travaux sur inginius à réaliser (programmes Java) et finalement un cours de restructuration en fin de module. Une des composantes essentielles de cette pédagogie consiste à faire apprendre chaque étudiant par lui-même. La réussite du processus d'apprentissage présuppose donc une implication significative de chaque étudiant. <b>L'apprentissage proprement dit reste de la responsabilité de chaque étudiant.</b> Pour réussir l'examen il est impératif que l'étudiant programme régulièrement.</p>
Contenu	<ul style="list-style-type: none"> <li>• Complexité calculatoire,</li> <li>• Arbres, arbres binaires de recherche,</li> <li>• Arbres équilibrés,</li> <li>• Dictionnaires et tables de hachage,</li> <li>• Files de priorité et tas,</li> <li>• Graphes</li> </ul>

	<ul style="list-style-type: none"> <li>• Manipulation de données textuelles (pattern matching et de compression)</li> </ul>
Ressources en ligne	<a href="https://moodleucl.uclouvain.be/course/view.php?id=7682">https://moodleucl.uclouvain.be/course/view.php?id=7682</a> <a href="https://lsinf1121.readthedocs.io">https://lsinf1121.readthedocs.io</a>
Bibliographie	<p>Required Textbook:            Algorithms, 4th Edition by Robert Sedgewick and Kevin Wayne, Addison-Wesley Professional.            ISBN-13: 978-0321573513            ISBN-10: 032157351X</p> <p>Exercices and documents  <a href="https://lsinf1121.readthedocs.io">https://lsinf1121.readthedocs.io</a>            Communication with students using moodle <a href="http://moodleucl.uclouvain.be/course/view.php?id=7682">http://moodleucl.uclouvain.be/course/view.php?id=7682</a></p>
Faculté ou entité en charge:	INFO

<b>Programmes / formations proposant cette unité d'enseignement (UE)</b>				
Intitulé du programme	Sigle	Crédits	Prérequis	Acquis d'apprentissage
Master [120] : ingénieur civil en mathématiques appliquées	MAP2M	5		
Bachelier en sciences mathématiques	MATH1BA	5		
Master [120] en science des données, orientation statistique	DATS2M	5		
Mineure d'accès en statistique et sciences des données	LSTAT100I	5		
Mineure en sciences de l'ingénieur : informatique	LSINF100I	5		
Mineure en sciences informatiques	LINFO100I	5		
Approfondissement en statistique et sciences des données	LSTAT100P	5		