



Guide pédagogique

Module « Algorithmique et complexité »

Options IASD et IAIL – 8.2 (4 crédits ECTS)

Place du module et enjeux

Les traitements informatiques, pour être performants, impliquent que les structures de données employées soient adaptées à l'application visée et à son contexte. La complexité des algorithmes doit être optimisée et les langages de programmation adaptés au contexte d'usage. Des outils peuvent être utilisés pour assurer une bonne qualité du code. Après avoir posé les bases de l'informatique théorique, ce module aborde ces différents aspects. Les applications, dans le cadre de cet enseignement, reposant principalement sur le langage C, une introduction à ce langage est également proposée.

Teaching guide and syllabus

« Algorithmics and complexity » module

DSAI and AISE options – 8.2 (4 ECTS credits)

Subject matter importance and associated issues

To be efficient, software developments requires data structures and algorithms adapted to the targeted application and its context. Algorithm complexity should be optimized to suit volume of data and/or constraints combinations and the programming languages adapted to the context of use. Some tools are dedicated to verification to ensure good code quality. This module lays the foundations of theoretical computer science and discusses the aforementioned points. The applications, within the framework of this teaching, are based mainly on the C language, an introduction to this language is also proposed.

Responsable : Michel Vasquez

Téléphone : 04 34 24 62 86

Courriel : michel.vasquez@mines-ales.fr



IMT Mines Alès
École Mines-Télécom

ENSEIGNEMENTS ACADÉMIQUES	Volume horaire	Détail des coefficients	Crédits
Algorithmique et complexité	50 h		
○ Introduction à l'informatique théorique	10	1	4
○ Programmation C	20	1	
○ Complexité des algorithmes et récursivité	20	1	

Matière 1 :

Titre de la matière : Introduction à l'informatique théorique	
Code : 2IA-8.2.1	Titre du module : « Algorithmique et complexité »
Semestre : S8	Cursus de rattachement : département 2IA

Heures présentiel	Heures total	Cours	TD	TP	Projet	Contrôles	Travail personnel	Coef /module	ECTS
10	13	9				1	3	1	1

Titre	Introduction à l'informatique théorique
Résumé	Mathématiques discrètes, calculabilité et principaux théorèmes d'informatique théorique.

Responsable	David Delahaye – LIRMM ¹
Équipe enseignante	David Delahaye – LIRMM

Mots-clés	Informatique théorique, Structures de données.
Prérequis	Connaissances mathématiques et informatique de L3

Contexte et objectif général :

Ce cours se destine à des étudiants généralistes ayant des connaissances de base en informatique, auxquels il est nécessaire de montrer que l'informatique s'appuie également sur des fondements théoriques solides et accessibles

Programme et contenu :

- Historique rapide.
- Structures mathématiques discrètes (ensembles, mots, graphes).
- Premiers modèles de calculs : automates finis et machine de Turing. Formalisation du calcul de fonctions. Problèmes de décision.
- Limites du calcul : indécidabilité. Problème de l'arrêt. Thèse de Church.
- Second modèle de calcul : fonctions récursives.
- Grands problèmes ouverts en informatique théorique.

Méthode et organisation pédagogique :

Les enseignements sont prévus pour 60 élèves.

Le découpage est prévu comme suit :

- 9h de cours/TD
- 1h d'évaluation

Acquis d'apprentissage visés :

Connaître les limites théoriques aux systèmes informatisés. Savoir s'appuyer sur des modèles théoriques pour prouver certaines propriétés des programmes. Connaître le socle commun aux langages impératifs.

Évaluation :

Contrôle écrit individuel 1h (coefficient 1)

Retour sur l'évaluation fait à l'élève :

Consultation des copies. Délai de correction : 3 semaines.

Support pédagogique et références :

1 Polycopié

¹ Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier.

Matière 2 :

Titre de la matière : Programmation C	
Code : 2IA-8.2.2	Titre du module : « Algorithmique et complexité »
Semestre : S8	Cursus de rattachement : Département 2IA

Heures présentiel	Heures total	Cours	TD	TP	Projet	Contrôles	Travail personnel	Coef /module	ECTS
20	26	10		10			6	1	1,5

Titre	Programmation C
résumé	Étude du langage, de ses particularités et de ses applications dans un contexte d'optimisation algorithmique.

Responsable	Binbin Xu – CERIS-I3A/IMT Mines Alès
Équipe enseignante	Binbin Xu – CERIS-I3A/IMT Mines Alès Doctorant en appui pour l'encadrement de TP

Mots-clés	Langage C
Prérequis	Développement d'algorithmes élémentaires dans un langage procédural

Contexte et objectif général :

Ce langage de 'bas niveau' reste un outil incontournable lorsque l'on veut maîtriser les ressources CPU (pour Central Processing Unit) pour le code généré. Il est suffisamment typé pour représenter les variables de nombreux problèmes réels afin d'implémenter des algorithmes de résolutions efficaces. La notion de pointeurs permet également de réaliser des calculs complexes en mémoire centrale.

Programme et contenu :

Compilation séparée et édition des liens, utilisation de bibliothèques. Implémentation d'un algorithme de calcul mathématique et d'une interface graphique.

Méthode et organisation pédagogique :

Les enseignements sont prévus pour 60 élèves. Les TP, par groupe de 2 étudiants, seront réalisés sur les ordinateurs personnels de ces derniers. Un sujet (projet) sera présenté dès le début du cours, les aspects du langage seront développés au fil de l'eau. Le découpage est prévu comme suit :

- 20h de cours/TP (en demi-groupes)

Acquis d'apprentissage visés :

Connaitre le langage C et savoir le mettre en application pour résoudre un problème donné.

Évaluation :

Évaluation en séance des comptes-rendus de TP

Retour sur l'évaluation fait à l'élève :

Mise à disposition des corrections, consultation des copies 3 semaines après la dernière séance.

Support pédagogique et références :

Slides et références bibliographiques

Matière 3 :

Titre de la matière : Complexité des algorithmes et récursivité	
Code : 2IA-8.2.3	Titre du module : « Algorithmique et complexité »
Semestre : S8	Cursus de rattachement : Département 2IA

Heures présentiel	Heures total	Cours	TD	TP	Projet	Contrôles	Travail personnel	Coef /module	ECTS
20	26	10		10			6	1	1,5

Titre	Complexité des algorithmes et récursivité
résumé	Fondamentaux de la complexité algorithmique : calcul de complexité, optimisation.

Responsable	Michel Vasquez – CERIS-I3A/IMT Mines Alès
Équipe enseignante	Michel Vasquez – CERIS-I3A/IMT Mines Alès Doctorant en appui pour l'encadrement de TP

Mots-clés	Complexité algorithmique et exploitation d'un langage à pile
Prérequis	Les étudiants doivent avoir une bonne expérience avec un environnement de développement tel que Visual C/C++ et bien connaître les notions de compilation séparée et d'édition des liens. Des éléments d'algorithmique sont souhaités.

Contexte et objectif général :

Ce cours est un complément à la programmation dans un langage fonctionnel tel que le langage C. Il met l'accent sur l'écriture de fonctions récursives tout en sensibilisant le programmeur aux notions de complexité spatiale et temporelle des algorithmes.

L'objectif de ce cours est, d'une part, d'apporter une meilleure compréhension d'un "langage à pile" et d'acquérir une première expérience dans l'écriture très expressive de fonctions récursives. D'autre part, ce cours présente les concepts élémentaires à l'évaluation, a priori, de la complexité du code développé pour traiter un ensemble donné d'information : notions indispensables pour concevoir un algorithme sensé résoudre un problème réel pour lequel la taille des données joue un rôle souvent critique.

Programme et contenu :

Complexité des algorithmes.
Exploration arborescente.

Méthode et organisation pédagogique :

Les enseignements sont prévus pour 60 élèves. Les TP seront réalisés sur les ordinateurs personnels de ces derniers.

Le découpage est prévu comme suit :

- 10h de cours
- 10h de TP

Acquis d'apprentissage visés :

Évaluer le coût en temps et en ressource mémoire d'un algorithme face aux données qu'il est sensé traiter. Décider, avant codage, si un algorithme est réaliste ou pas.

Exploiter la souplesse et l'efficacité d'un langage à "pile", et exprimer en quelques lignes de code C des traitements relativement complexes.

Évaluation : TP à rendre ; la note finale sera la moyenne des comptes-rendus de TP

Retour sur l'évaluation fait à l'élève :

3 semaines après la dernière séance.

Support pédagogique et références :

Slides du cours et références bibliographiques

Méthode et organisation pédagogique

Il s'agit d'un enseignement relativement classique avec une partie réalisée en cours magistral et une partie appliquée au travers de TP.

Modalité d'évaluation

Le niveau d'acquisition des compétences sera évalué selon les exigences suivantes :

N° indicateur	Indicateur
1	connaître les savoirs formels et pratiques du socle des fondamentaux
2	Exploiter les savoirs théoriques et pratiques
3	Analyser, interpréter, modéliser, émettre des hypothèses, et résoudre

Répartition

Matière	Contrôle	Coefficients	Type de notation	Indicateurs évalués	Chapitres
Introduction à l'informatique théorique	Devoir sur table	1	Individuelle	1	Tous
Programmation C/C++	TP notés	1	En trinôme	2 et 3	Tous
Complexité des algorithmes et récursivité	TP notés	1	En trinôme	2 et 3	Tous

Des évaluations non prévues à l'emploi du temps (contrôles surprise) peuvent advenir.

Engagement de l'étudiant, éthique et professionnalisme

La démarche éthique est définie dans le règlement intérieur de l'établissement. Chaque étudiant s'engage à en prendre connaissance et à la respecter.

Obligation des cours : Présence obligatoire pour tous à chaque séance

Nombre d'heures estimées de travail personnel : pour acquérir les compétences demandées, il est nécessaire que l'étudiant consacre minimum 45 min de travail personnel de compréhension et d'approfondissement par séance de cours.

15h de travail personnel estimées sur la totalité du module.

Nombre d'heures estimées de préparation aux travaux dirigés (TD) : 1 à 2h

Pénalité pour retard

Tout travail remis en retard sans motif valable peut être pénalisé de 1 point par jour de retard.

Équipe enseignante

Nom	Domaine d'expertise	Courriel/Téléphone
Michel VASQUEZ	Complexité algorithmique, optimisation combinatoire, résolution de problèmes.	michel.vasquez@mines-ales.fr 04 34 24 62 86
David DELAHAYE LIRMM / Université de Montpellier	Méthodes et langages formels, démonstration automatique de théorèmes, modélisation de systèmes, formalisation mathématique.	David.Delahaye@lirmm.fr 04 67 41 86 01
Binbin XU	Algorithmique, apprentissage automatique, Techniques d'intelligence artificielle	binbin.xu@mines-ales.fr 04 66 78 50 00

ACADEMIC TEACHING	Teaching hours	Coefficients	Credits
Algorithmics and complexity	50 h		
○ Introduction to theoretical Computer Science	10	1	4
○ C programming language	20	1	
○ Algorithmics complexity and recursivity	20	1	

Class 1

Title : Introduction to theoretical Computer Science	
Code : 2IA-8.2.1	Module title : « Algorithmics and complexity »
Semester: S8	Classification : CSAI department

Hours of presence	Total hours	Lectures	Work shop	Labs	Project	Testing	Personal work	Coef /module	ECTS
10	13	9				1	3	1	1

Title	Introduction to theoretical Computer Science
Summary	Discret mathematics, calculability, main theorems in theoretical computer science.,

Head	David Delahaye – LIRMM ²
Teaching team	David Delahaye – LIRMM

Key words	Theoretical Computer Science, Discrete Mathematics, Finite Automata, Decidability, Recursive Functions.
Prerequisites	Mathematical basis and computer science studied in S5 and 6

Context and general objective:

This lecture is intended for generalist students with basic computer skills, to whom it is necessary to show that computer science also relies on a solid and accessible theoretical foundation.

Program and contents:

- Quick history
- Discrete mathematical structures (sets, words, graphs)
- First models of calculations: finite automata and Turing machine. Formalization of function calculation.
- Decision problems.
- Calculation limits: undecidability. Termination problem. Church thesis.
- Second calculation model: recursive functions.
- Great open problems in theoretical computer science.

Method and pedagogical organisation:

Lessons are planned for 60 students as follows: 9 hours of lecture and practices, 1 hour of evaluation.

Targeted skills or knowledge:

To know the theoretical limits to computer systems. To be able to rely on theoretical models to prove certain properties of programs. To know the common basis to imperative languages.

Evaluation:

One hour individual written exam (coef. 1)

Feedback made to the student: making corrections available, consulting copies of exams.

Deadlines for giving correction of exams: 3 weeks

Teaching material and references :

1 Photocopied material – textbook – articles.

² The Montpellier Laboratory of Informatics, Robotics and Microelectronics (LIRMM) is attached to Montpellier University and the French National Center for Scientific Research (CNRS)

Class 2

Title : C programming	
Code : 2IA-8.2.2	Module title : « Algorithmics and complexity »
Semester: S8	Classification : CSAI department

Hours of presence	Total hours	Lectures	Work shop	Labs	Project	Testing	Personal work	Coef /module	ECTS
20	26	10		10			6	1	1,5

Title	C programming
Summary	Study of C language, its particularities and application in a context of algorithmic optimization.

Head	Binbin Xu – CERIS-I3A/IMT Mines Alès
Teaching team	Binbin Xu – CERIS-I3A/IMT Mines Alès Additional assistant for Labs

Key words	C programming language,
Prerequisites	Development of elementary algorithms in a procedural language

Context and general objective:

This 'low level' language remains an essential tool when one wants to control the CPU (Central Processing Unit) resources of the generated code. It is sufficiently typed to represent the variables of many real problems in order to implement efficient resolution algorithms. The notion of pointers also makes it possible to perform complex computations in central memory.

Programme and contents:

Separate compilation and linking, use of library. Implementation of a mathematical computation algorithms and a graphical interface.

Method and pedagogic organisation:

Lessons are planned for 60 students. Workshops, in groups of 3 students, will be carried out on their personal computers. A subject (project) will be presented at the beginning of the course; the aspects of the language will be developed during the sessions.

The cutting is planned as follows:
- 20h of lessons / TP

Targeted skills or knowledge :

Know the basis of C programming language and how to apply it to solve a given problem.

Evaluation :

In-session evaluation of Labs reports

Feedback made to the student :

In-session feedbacks with personalised comments. Marks given 3 weeks after the last session.

Teaching material and references :

Lecture slides and references textbook.

Class 3

Title : Complexity of Algorithms and Recursivity	
Code : 2IA-8.2.3	Module title : « Algorithmics and complexity »
Semester: S8	Classification : CSAI Department

Hours of presence	Total hours	Lectures	Work shop	Labs	Project	Testing	Personal work	Coef /module	ECTS
20	26	10		10			6	1	1,5

Title	Complexity of Algorithms and Recursivity
Summary	Fundamentals of algorithmic complexity: complexity calculation, optimization

Head	Michel Vasquez – CERIS-I3A/IMT Mines Alès
Teaching team	Michel Vasquez – CERIS-I3A/IMT Mines Alès Additional assistant for Labs

Key words	Algorithmics, complexity, getting used with the stack programming language
Prerequisites	Students should have good practice with an IDE such as Visual C / C ++ and be familiar with the concepts of separate compilation and link of code. Basic knowledges in algorithmic are desirable.

Context and general objective:

This course is a complement to the programming in a functional language such as the C language. It focuses on the writing of recursive functions while sensitizing the programmer to the notions of spatial and temporal complexity of the algorithms.

The objective of this course is, on the one hand, to bring a better understanding of a "stack language" and to gain a first experience in the very expressive writing of recursive functions. On the other hand, this course introduces the elementary concepts to the evaluation, *a priori*, of the complexity of the code developed to treat a given set of data essential notions to the design of an algorithm supposed to solve a real problem for which the size of the data often plays a critical role.

Programme and contents:

Complexity of algorithms.
Tree search.

Method and pedagogic organisation:

The course will be organized as follows (for 60 students):

- 10h of lectures
- 10h of labs

Targeted skills or knowledge :

Get an estimation of the cost in CPU time and memory of an algorithm regarding the data it is supposed to treat. Decide, before coding, whether an algorithm is realistic or not.
Exploit the flexibility and efficiency of a "stack" language, and express in a few lines of C code relatively complex processing.

Evaluation :

In-session evaluation of the labs reports.

Feedback made to the student :

In-session feedbacks with personalised comments. Marks given 3 weeks after the last session.

Teaching material and references :

Lecture slides and reference textbooks.

Method and teaching organisation

Teachings will consist of lectures, seminars and labs.

Testing procedures

The student's level of knowledge acquisition will be evaluated according to the following points:

N° Indicator	Indicator
1	To know the formal and practical knowledge constituting the foundation of a given field
2	Exploit theoretical and practical knowledge
3	Analyse, interpret, model, hypothesize and solve problems

Grading scheme:

Class	Exam	Coefficients	Administration mode	Evaluated Indicators	Chapters
Introduction to Theoretical Computer Science	Exam	1	Individual	1	all
C/C++ programming	Practical report	1	Trio	2 and 3	all
Algorithmic complexity and recursivity	Practical report	1	Trio	2 and 3	all

Unscheduled assessments (surprise checks) may occur.

Student commitments, ethics and professionalism

Expectations concerning ethics are defined in the establishment's code of conduct. Each student is expected to know and respect the code of conduct.

Obligatory presence in classes: Students must attend all courses, seminars and labs.

Estimated hours of personal study: in order to acquire the required learning level, the student is expected (must) to spend a minimum of 45min of personal study time per hour spent in class. 15h of personal work is estimated for this module.

Estimated hours of preparation required for labs/Work Shop: 45 min

Late penalties: Late works are subject to penalties as follows: 1 point per day (ratings between 0 and 20).

Teaching team

Name	Expertise	Email/Phone
Michel VASQUEZ	Algorithmic, optimisation, problem solving, combinatorial optimisation	michel.vasquez@mines-ales.fr 04 34 24 62 86
David DELAHAYE LIRMM / Université de Montpellier	Modeling of Critical (Embedded) Systems, Formal Methods, Interactive and Automated Theorem Proving, Formalization of Mathematics	David.Delahaye@lirmm.fr 04 67 41 86 01
Binbin XU	Algorithmic, Machine Learning, AI techniques	binbin.xu@mines-ales.fr 04 66 78 50 00

Approbation

Ce guide pédagogique entré en vigueur à compter du 07 janvier 2019 a été mis à jour en juillet 2022.
Il est porté à la connaissance des élèves par une publication sur le site de l'école

Rédaction	Vérification	Validation
L'enseignant responsable du module : Michel Vasquez	Le responsable d'UE / de département : Sylvie Ranwez	Le directeur de l'école, Pour le directeur et par délégation, Le directeur de la DFA / de la DE : Michel Ferlut