

Licence Mathématiques

Mathématiques discrètes et algorithmique

Responsable	Descriptions	Informations
	Code : SMI5U23	Composante : Faculté des Sciences
	Nature : Unité d'enseignement	
	Domaines : Sciences et Technologies	

LANGUE(S) D'ENSEIGNEMENT

Français

CONTENU

Cette UE vise à travailler les compétences en algorithmique et programmation en revisitant et complétant les acquis mathématiques des années précédentes, principalement en arithmétique et en algèbre. La mise en oeuvre utilise le langage Python, et s'appuie sur les compétences acquises dans l'UE d'Algorithmique et programmation 1 suivie en L2 Mathématiques.

Programme

- Mise en route : calcul du n-ième terme de la suite de Fibonacci par des définitions récursive et itérative ; comparaison avec les résultats obtenus pour l'expression fonctionnelle (formule de Binet).
- Entiers relatifs : représentations d'un entier (décimal, binaire, en base quelconque), algorithme de division euclidienne à partir de (+, -, <), conversions entre bases, pgcd, tests de primalité et algorithme naïf de factorisation (discussion de la complexité). Réalisation en Python d'une sélection des algorithmes étudiés.
- Arithmétique modulaire : caractérisation des sous-groupes additifs de \mathbb{Z} , rappel de la définition de $\mathbb{Z}/n\mathbb{Z}$, opérations héritées de \mathbb{Z} , générateurs de $\mathbb{Z}/n\mathbb{Z}$, groupe multiplicatif, inverse modulaire (calcul par l'algorithme d'Euclide étendu), lemme chinois (démonstration effective et implémentation), indicatrice d'Euler (calcul de $\phi(p)$ avec p premier, $\phi(p^n)$, $\phi(mn)$, cas général, discussion de la complexité).
- Polynômes en une indéterminée $K[X]$: opérations de base, division euclidienne et calcul du pgcd, et discussion de la structure pertinente sur K pour réaliser ces opérations. Réalisation en Python dans le cas $K = \mathbb{Z}/p\mathbb{Z}$.
- Étude d'un algorithme avancé (par exemple : RSA ou Reed-Solomon) : prérequis mathématiques, principe de l'algorithme, implémentation, discussion de la robustesse.

COMPÉTENCES À ACQUÉRIR

Connaissance du cours

Ce cours contient assez peu de contenu mathématique nouveau, et n'introduit pas de connaissance nouvelle en informatique et programmation. Il s'agit plutôt de consolider les acquis mathématiques en variant le point de vue, et de développer une certaine aisance dans la pratique de la programmation dans un contexte mathématique.

Compétences À l'issue de cette UE, on s'attend à ce que les étudiants validant le cours soient capables de :

- mobiliser leurs connaissances mathématiques en algèbre et arithmétique pour établir ou contester la correction d'un algorithme ou d'un programme ;
- identifier, parmi les définitions et résultats qu'ils mobilisent en mathématiques, ceux qui sont présentés de manière effective, et donc directement susceptibles d'être réalisés algorithmiquement ;
- transcrire un algorithme élaboré (du niveau de ceux abordés explicitement dans leur cursus) en programme Python ;

- proposer ou critiquer des choix d'implémentation pour cette dernière tâche : types de données, structuration du programme, appel à des bibliothèques externes, etc.

MODALITÉS D'ORGANISATION

12h cours, 18h TP

VOLUME HORAIRE

- Volume total: 30 heures
- Cours magistraux: 12 heures
- Travaux pratiques: 18 heures

CODES APOGÉE

- SMI5U23L [ELP]

M3C

Aucune donnée M3C trouvée

POUR PLUS D'INFORMATIONS

[Aller sur le site de l'offre de formation...](#)



Dernière modification le 17/07/2024