Algorithmique et programmation 3

ECTS : 4

Enseignant responsable : Denis CORNAZ (https://www.lamsade.dauphine.fr/~cornaz/)

Langue du cours : Français

Volume horaire : 49

Description du contenu de l'enseignement :

Chacun des points suivants sera présenté et expérimenté en langage Python :

1. Algorithmes et fonctions logarithmes : logarithmes naturels dans les appels récursifs ou dans les boucles type série harmonique, preuves courtes des propriétés de base des logarithmes. Notations asymptotiques et arrondis récursifs.
2. Complexité : algorithmes en T(n)=aT(n-b) + poly(n), et application aux implémentations exponentielle/linéaire de Fibonacci et à l'algorithme d'Euler-Bachet-Bezout.
3. Récursivité de la forme T(n)=aT(n/b) + poly(n): (rappel tri fusion), preuve courte du "Master Theorem", calcul rapide de complexité à partir du cas n puissance de b.
4. Performance des algorithmes : application du "Master Theorem" à la conception d'algorithmes de multiplication rapide d'entiers (Karatsuba), et de matrices (Strassen).
5. Force brute : algorithmes énumératifs, application à la résolution de systèmes d'équations et aux placements de reines sur échiquiers nxn.
6. Complexités des Tris : variétés du concept de complexité (pire cas, moyenne, structure des données) avec les algorithmes classiques de tri (rappel: insertion, dénombrement, tas)

Compétences à acquérir :

Fondements mathématiques de la complexité algorithmique et idées précises, avec connaissances profondes des exemples emblématiques, de ses paradigmes centraux. Maîtrise des mécanismes de base du langage Python.