

Algorithmique et programmation 3

ECTS : 5

Volume horaire : 49.5

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 où dans les boucles type série harmonique, preuves courtes des propriétés de base des logarithmes. Notations asymptotiques et arrondis récursifs.
2. Complexité : algorithmes en $T(n)=aT(n-b) + \text{poly}(n)$, et application aux implémentations exponentielle/linéaire de Fibonacci et à l'algorithme d'Euler-Bachet-Bezout.
3. Récursivité de la forme $T(n)=aT(n/b) + \text{poly}(n)$: (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. Tri : variétés du concept de complexité (pire cas, moyenne, expression des données) avec les algorithmes classiques de tri (rappel: insertion, dénombrement, tas)
6. Force brute : algorithmes énumératifs, application à la résolution de systèmes d'équations et aux placements de reines sur échiquiers $n \times n$.

Compétence à acquérir :

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