

# 3IF1010 - Algorithmique avancée

Responsables : **Joanna TOMASIK**

Langues d'enseignement : **FRANCAIS**

Campus où le cours est proposé : **CAMPUS DE PARIS - SACLAY**

Nombre d'heures d'études élèves (HEE) : **35**

Nombre d'heures présentielles d'enseignement (HPE) : **18**

Année académique : **2024-2025**

Niveau avancé : **non**

## Présentation, objectifs généraux du cours :

L'objectif de ce cours est d'enrichir les connaissances et de renforcer les compétences acquises en cours Algorithmique et Complexité offert pour tous les élèves en première année.

Il vise également à mettre les élèves dans la situation semblable à celle qui est pratiquée lors des entretiens de recrutement par des acteurs majeurs de l'Internet et des éditeurs de logiciels réputés - une compétition en programmation : efficacité de solutions algorithmiques, performance de l'implémentation, rapidité de programmation.

## Période(s) du cours (n° de séquence ou hors séquence) :

SD9

## Prérequis :

1CC1000 : Systèmes d'information et programmation

1CC2000 : Algorithmique et Complexité

## Plan détaillé du cours (contenu) :

**Le cours est organisé en deux blocs.**

Intitulés des cours du bloc **Analyse fine des problèmes difficiles**, thématique des TD/TP :

**Cours 1** : Les liens entre espace mémoire et temps de calcul

**TD/TP 1** : Exercices pour le cours 1

**Cours 2** : Affiner la classification des problèmes P et NP (approximation et inapproximation)

**TD/TP 2** : Exercices pour le cours 2

**Cours 3** : Affiner la classification des problèmes P et NP (complexité paramétrée)

Intitulés des cours du bloc **Algorithmes randomisés, algorithmes online**, thématique des TD/TP :

**Cours 1** : Algorithmes randomisés : Las Vegas et Monte Carlo

**Cours 2** : Introduction à l'algorithmique online : prise de décision avec l'information incomplète et analyse compétitive

**TD/TP 1** : Exercices pour les cours 1 et 2

**Cours 3** : Algorithmes online des systèmes d'exploitation : gestion de ressources ; principe de Yao

**TD/TP 2** : Exercices pour les cours 3

Le module sera conclu par une session de 3 heures de retours après la compétition en conception d'algorithmes et en programmation.

## Déroulement, organisation du cours :

Chaque bloc thématique est organisé en :

- trois sessions de cours magistral en amphithéâtre de 1.5 heure présentant des bases théoriques et des approches principales de la conception d'algorithmes,
- deux séances de TD/TP de 1.5 heure pendant laquelle les élèves, guidés par un enseignant, étudieront un panel de problèmes difficiles, analyseront (ou proposeront) leurs solutions algorithmiques accompagnées d'une étude de leur qualité et les mettront en œuvre en langage python.

## Organisation de l'évaluation :

Les élèves seront confrontés à des problèmes d'optimisation. Ils seront demandés de :

- proposer une solution algorithmique pour le résoudre,
- calculer la complexité de l'algorithme en temps et en espace,
- déterminer la garantie sur la qualité de la solution que leur algorithme produit,
- implémenter l'algorithme en langage python en utilisant des méthodes de programmation et des structures de données les plus adaptées pour que le temps d'exécution soit le plus court possible (en temps du processeur),
- une note sera une fonction de la qualité théorique de l'algorithme et de la manière de l'obtenir ainsi que de l'efficacité du code en python,
- l'évaluation de la performance du programme résultera d'une compétition lancée sur une plateforme numérique et sera calculée sur la base du classement,
- les résultats de cette compétition seront commentés lors d'une session de retours de 3 heures.

## Moyens :

**Les cours en amphithéâtre donnés par :**

- Joanna TOMASIK (Joanna.Tomasik@centralesupelec.fr)
- Marc-Antoine WEISSER (Marc-Antoine.Weisser@centralesupelec.fr)

**Les chargés de TD/TP :**

- Johanne COHEN (Johanne.Cohen@centralesupelec.fr)
- Arpad RIMMEL (Arpad.Rimmel@centralesupelec.fr)
- Joanna TOMASIK (Joanna.Tomasik@centralesupelec.fr)
- Benoît VALIRON (Benoit.Valiron@centralesupelec.fr)
- Marc-Antoine WEISSER (Marc-Antoine.Weisser@centralesupelec.fr)

Programmation en python. Une plateforme numérique pour abriter une compétition des implémentations et pour réaliser leur classement.

## Acquis d'apprentissage visés dans le cours :

---

- l'analyse fine des problèmes difficiles : des problèmes pouvant être résolus en temps pseudo-polynomial (les problèmes faiblement NP-difficiles), des éléments de l'analyse de la complexité paramétrée (FTP), des problèmes dont la solution peut être approchée arbitrairement en temps étant un polynôme de la taille du problème et l'inverse de la précision (FPTAS), des problèmes restant toujours difficiles (les problèmes fortement NP-difficiles)
- les algorithmes randomisés et les algorithmes prenant la décision avec l'information incomplète : des algorithmes randomisés pour résoudre des problèmes d'optimisation difficiles (Las Vegas et Monte Carlo), des algorithmes online, déterministes et randomisés, l'analyse compétitive évaluant leur performance dans le pire des cas, des techniques de gestion des structures de données dynamiques et leur qualité, des algorithmes online utilisés par les systèmes d'exploitation
- l'implémentation des solutions algorithmiques proposées en langage de programmation en tenant compte de sa qualité en termes de consommation de mémoire et de temps d'exécution

## Description des compétences acquises à l'issue du cours

- to know theoretical approaches to determine the nature of a difficult optimization problem (problem classes: weak/strongly hard, FTP, FPTAS, APX) ; la compétence C2.1 (Avoir approfondi un domaine ou une discipline relatifs aux sciences fondamentales ou aux sciences de l'ingénieur)
- to know how to design randomized algorithms; la compétence C2.1 (Avoir approfondi un domaine ou une discipline relatifs aux sciences fondamentales ou aux sciences de l'ingénieur)
- to know how to make a decision with incomplete information by proposing online algorithms and to know how to determine the quality of the proposed algorithmic solutions (competitive analysis) ; la compétence C2.1 (Avoir approfondi un domaine ou une discipline relatifs aux sciences fondamentales ou aux sciences de l'ingénieur)
- to know advanced techniques of algorithm design ; la compétence C6.4 (Résoudre des problèmes dans une démarche de pensée computationnelle)
- to put into practice algorithms designed (the implementation in python) ; la compétence C6.4 (Résoudre des problèmes dans une démarche de pensée computationnelle)

## Bibliographie :

Les références bibliographiques sur des articles scientifiques traités seront données sur les transparents des cours concernés.