

# Algorithmes et structures de données 1

<b>Domaine</b>	Ingénierie et Architecture
<b>Filière</b>	Informatique
<b>Orientation</b>	Logiciel (IL)
<b>Mode de formation</b>	Plein temps

## Informations générales

Nom:	:	Algorithmes et structures de données 1
Identifiant:	:	ASD1
Années académiques	:	2016-2017, 2017-2018, 2018-2019, 2019-2020
Responsable:	:	Laura Elena Raileanu
Charge de travail:	:	150 heures d'études
Périodes encadrées:	:	96 (= 72 heures)

Semestre	E1	S1	S2	E2	S3	S4	E3	S5	S6
Cours			64						
Laboratoire			32						

## Connaissances préalables recommandées

L'étudiant-e doit connaître et savoir utiliser les notions suivantes : langage C++: constituants d'un programme, types simples et structurés, structures de contrôle, sous-programmes, exceptions, pointeurs ; pseudo-code, schémas illustrant les structures de données de base (tableau, liste chaînée par pointeurs). Les unités INF1 et INF2 (informatique) permettent d'acquérir ces connaissances.

## Objectifs

A l'issue de cette unité d'enseignement, l'étudiant-e sera capable de :

- concevoir un type de données abstrait à partir de spécifications données en langage naturel ;
- expliquer la notion de complexité et l'illustrer sur des cas simples ;
- concevoir et réaliser des algorithmes et des structures de données récursifs ou non ;
- analyser et expliquer un algorithme ou une structure de données ;
- choisir et justifier le choix d'un algorithme ou d'une structure de données ;
- choisir un algorithme de tri et expliquer son fonctionnement ;
- concevoir une structure linéaire particulière de manière statique ou dynamique ;
- appliquer les parcours d'arbre sur des exemples concrets : arbres binaires, arbres arbitraires, arbres de recherche ;
- appliquer les algorithmes d'insertion et de suppression d'éléments dans un arbre binaire et un arbre de recherche ;
- appliquer les parcours en profondeur et en largeur sur des graphes non pondérés, orientés ou non ;
- appliquer les parcours ARPM et ACPC sur des graphes pondérés, orientés ou non ;
- calculer les composantes fortement connexes d'un graphe orienté ;

A l'issue des travaux pratiques en laboratoire, l'étudiant-e sera en outre capable de :

- implémenter un type de données abstrait en respectant soigneusement les règles habituelles de réalisation (réutilisabilité, encapsulation, traitement d'exceptions, etc.) ;
- utiliser des types de données abstraits existants chaque fois que la réutilisation de code est possible ;
- réaliser et utiliser des algorithmes récursifs ou non ;
- implémenter une structure de données particulière ;
- utiliser une structure de données particulière dans le cadre d'une application plus globale ;
- exercer le test de programmes ;
- exploiter l'expérience acquise dans l'analyse de code, la détection et la correction d'erreurs de programmation ;
- rédiger de petits rapports de laboratoire.

## Contenu et formes d'enseignement

*Répartition des périodes indiquée à titre informatif.*

**Cours:** 64 périodes

- Introduction, motivation, pseudo-code, principes de base pour l'analyse d'un algorithme, complexité, structure de données, type de données abstrait, paradigme type de données abstrait 8
- Récursivité, règles de la récursivité, comparaison avec l'itération, exemples, complexités 8
- Tris, classification de tris, tri bulles, tri par sélection, tri par insertion, tri rapide, tri par fusion, tri de shell, tri par tas, complexités 8
- Structures linéaires, listes simples, listes doubles, multilistes, piles, queues, queues de priorité, tas, complexité des algorithmes étudiés, réalisation statique et dynamique 12
- Arbres, définitions, arbres binaires, parcours d'arbres, arbres de recherche, notion de tas, équilibrage, complexité des algorithmes étudiés, réalisation dynamique 16
- Graphes non orientés et non pondérés: définitions et algorithmes; 12

**Laboratoire:** 32 périodes

- Conception, réalisation et test d'un type de données abstrait 4
- Récursivité, implémentation d'un algorithme en version itérative et récursive, comparaison 6
- Mesure de performances d'algorithmes de tri, conception et réalisation d'algorithmes de tri (génération des nombres aléatoires, application des algorithmes, modélisation, comparaison entre valeurs prédites et valeurs estimées) 8
- Structures linéaires : applications à la gestion de données comme par exemple la gestion d'adresses ou d'échéances et réalisations de types particuliers comme les listes triées, les multilistes 6
- Application des parcours des arbres, arbres de recherche, implémentation 8

## Bibliographie

1. "Algorithms", 4/E, Robert Sedgewick and Kevin Wayne, Addison-Wesley Professional, 2011, ISBN 0132762560, 9780132762564.
2. " Introduction to algorithms", Thomas Cormen, Charles Leiserson, Ronald Rivest, Clifford Stein, Third Edition, MIT Press, 2009, ISBN-10: 0262033844.

3. "Algorithmes: Notions de base", Thomas Cormen, Éditeur Dunod, 2013, ISBN 2100702904, 9782100702909.

## Contrôle de connaissances

### Cours:

l'acquisition des matières de cet enseignement sera contrôlée au fur et à mesure par des tests et des travaux personnels tout au long de son déroulement. Il y aura au moins 2 tests d'une durée totale d'au moins 4 périodes.

### Laboratoire:

ils seront évalués sur la base des rapports de manipulation, à 3 reprises au minimum.

### Examen:

L'atteinte de l'ensemble des objectifs de formation sera vérifiée lors d'un contrôle final commun écrit d'une durée de 90 minutes.

### Matériel autorisé:

- Information communiquée directement par l'enseignant.

## Calcul de la note finale

Note finale = moyenne cours x 0.3 + moyenne laboratoire x 0.2 + moyenne examen x 0.5

Fiche validée le 16.08.2016 par Donini Pier