

## Algorithmes et structures de données 2

<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 2
Identifiant:	:	ASD2
Années académiques	:	2016-2017, 2017-2018, 2018-2019, 2019-2020, 2020-2021
Responsable:	:	Laura Elena Raileanu
Charge de travail:	:	90 heures d'études
Périodes encadrées:	:	64 (= 48 heures)

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

### Connaissances préalables recommandées

L'étudiant doit connaître et savoir utiliser les notions suivantes :

tous les aspects séquentiels du langage C/C++ et les bases de l'objet, constituants d'un programme, types simples et structurés, structures de contrôle, sous-programmes, exceptions, bases des paquetages et de la généricité, pointeurs (access) ;

conception et implémentation d'un type de données abstrait, complexité, récursivité, tris, structures linéaires, arbres binaires, arbres de recherche, graphes, parcours de graphes;

Les unités d'enseignement INF1 et INF2 (Informatique), ASD1 (Algorithmes et Structures de Données), MAD (Mathématiques Discrètes) et MBT (Mathématiques de Base pour TIC) permettent d'acquérir ces connaissances.

### Objectifs

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

- 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 ;
- utiliser, implémenter et appliquer les fonctions de hachage et implémenter et utiliser les tables de recherche;
- analyser la complexité des opérations sur les tables de recherche et choisir la structure optimale;
- appliquer les algorithmes d'insertion et de suppression avec rééquilibrages sur des arbres AVL ;
- appliquer les algorithmes d'insertion et de suppression sur des B-trees;
- analyser et connaître les complexités des algorithmes étudiés au cours;

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

- utiliser une structure de données particulière dans le cadre d'une application plus globale;
- exercer le test de programmes;
- acquérir de l'expérience dans l'analyse et la conception de code, la détection et la correction d'erreurs de programmation;
- exercer la rédaction de rapports de laboratoire (taille moyenne);
- concevoir et développer des applications de taille moyenne, étudier des sujets plus conséquents algorithmiques.

## Contenu et formes d'enseignement

*Répartition des périodes indiquée à titre informatif.*

**Cours:** 32 périodes

- Graphes : récapitulation sur la représentation par matrice de contiguïté et par listes d'adjacence; cheminements et connexités (notions de chemin, chaîne, cycle et circuit, fermeture transitive d'un graphe, notions de connexité, notion de graphe eulérien, notion de graphe hamiltonien); arbres et arborescences; parcours de graphes: arborescence couvrante associée à un parcours, parcours en largeur, applications du parcours en largeur (composantes connexes d'un graphe non orienté, chercher le plus court chemin, sommets accessibles à partir d'un sommet initial), parcours en profondeur, applications du parcours en profondeur (recherche de circuits, tri topologique des sommets d'un graphe orienté, recherche des composantes fortement connexes d'un graphe orienté); plus courts chemins: définitions, algorithme de Dijkstra, algorithme de Bellman-Ford; arbres couvrants minimaux: algorithme de Kruskal, algorithme de Prim; complexités des algorithmes étudiés. 10
- Tables et recherche, définitions, recherches séquentielle, dichotomique et par arbre de recherche, fonction de hachage, hachage ouvert et fermé, complexité des algorithmes étudiés, réalisation 8
- Arbres équilibrés : arbres AVL, algorithmes d'insertion et de suppression d'éléments avec rééquilibrage, complexité des algorithmes étudiés 8
- B-arbres: algorithmes d'insertion et de suppression, complexité des algorithmes étudiés 4
- Applications, étude de cas, codage, exercices de récapitulation 2

**Laboratoire:** 32 périodes

- Algorithme de compression de Huffman 10
- Graphes : étude de problèmes liés à la représentation, application des parcours ARPM et ACPC par exemple aux réseaux de communication ou à l'optimisation en général, application des graphes orientés aux problèmes de tri et d'ordonnancement de tâches, etc. 12
- Correcteur d'orthographe en utilisant une structure de données de type arbre équilibré et des algorithmes de recherche d'information 10

## 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 2 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 60 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