

## Graphes et réseaux

<b>Domaine</b>	Ingénierie et Architecture
<b>Filière</b>	Informatique et systèmes de communication
<b>Orientation</b>	Informatique logicielle (ISCL)
<b>Mode de formation</b>	Plein temps

### Informations générales

Nom	: Graphes et réseaux
Identifiant	: GRE
Années académiques	: 2021-2022, 2022-2023
Responsable	: Jean-François Hêche
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 :

- ensembles, relations et fonctions, calcul de séries élémentaires et manipulation du symbole somme, calcul matriciel, raisonnement par récurrence ;
- pseudocode, comportement asymptotique des fonctions (notation de Landau), complexité d'un algorithme ;
- propriétés des structures de données linéaires et arborescentes simples (tableaux, listes, tas) ;
- notions et algorithmes de base sur les graphes (définitions, représentation, exploration, calcul d'arbres et de chaînes/chemins).

Les unités d'enseignement MAD (mathématiques discrètes), MAT1 et MAT2 (mathématiques 1 et 2) et ASD (algorithmes et structures de données) permettent d'acquérir ces connaissances.

### Objectifs

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

- définir et utiliser correctement les notions de théorie des graphes du cours (graphe « orienté/non orienté », « arête/arc », « chaîne/chemin », « cycle/circuit », « biparti », « complet », « simple », « eulérien », etc) et les appliquer à la caractérisation d'un graphe donné ;
- passer d'une représentation d'un graphe à une autre (graphique, matricielle, par tableaux et listes) et discuter les avantages et inconvénients des représentations informatiques classiques ;
- décrire les problèmes classiques de la théorie des graphes et énoncer, appliquer et discuter le fonctionnement des algorithmes de résolution étudiés :
  - parcours de graphes (en largeur ou en profondeur et leurs applications : tri topologique, composantes connexes/fortement connexes), algorithme de Tarjan,
  - arbres et arborescences optimaux (arbres recouvrants de poids minimal, algorithme de Kruskal et de Prim, arborescences recouvrantes de poids minimal, algorithme de Chu-Liu, chaînes et chemins de section optimale),
  - plus courts chemins (algorithmes de Bellman-Ford et de Dijkstra, de Johnson, de Floyd-Warshall, ...),

- ordonnancement (tri topologique, graphe potentiels-tâches, chemin critique, applications et extensions),
- flots dans les réseaux (flot de valeur maximale, coupe de capacité minimale, algorithme de Ford-Fulkerson, flot à coût minimum, algorithme de Busacker et Gowen, variantes (problèmes de transbordement) et applications à des problèmes particuliers : couplages, postier chinois, ... ;
- modéliser une situation donnée à l'aide d'un graphe, identifier le(s) problème(s) de graphes à résoudre, établir une démarche de résolution, choisir l'algorithme étudié le plus adapté à la résolution de chaque problème et, selon les cas, les résoudre en appliquant la démarche retenue (ce point constitue l'un des objectifs majeurs du cours) ;
- comprendre et appliquer des algorithmes de graphes présentés sous forme d'un pseudocode, discuter leur complexité ;
- développer ou adapter des pseudocodes pour la résolution de problèmes proches de ceux étudiés ;
- mettre en oeuvre et tester des algorithmes étudiés ou développés et les appliquer à la résolution de problèmes d'optimisation ou d'aide à la décision.

### Contenu et formes d'enseignement

Répartition des périodes indiquée à titre informatif.

**Cours:** 64 périodes

- Rappels et précisions sur les définitions et notions de base	4
- Représentation des graphes : matrices d'adjacence et d'incidence, tableau de listes et tableau compact de successeurs/voisins	3
- Parcours en largeur et en profondeur et applications à la recherche de composantes	5
- Arbres et arborescences : arbres recouvrants optimaux, chaînes et chemins de section optimale, arborescences recouvrantes optimales, algorithmes et applications	8
- Plus courts chemins : poids quelconques et circuits absorbants, poids positifs ou nuls, technique de repondération, algorithmes matriciels	14
- Graphes sans circuits : caractérisations, tri topologique, plus courts et plus longs chemins, application à l'ordonnancement (graphe potentiel-tâche, chemin critique, ...)	6
- Graphes eulériens : caractérisations, recherche de parcours eulériens, problème du postier chinois	6
- Flots dans les réseaux : flot de valeur max, coupe de capacité min, flot à coût min, algorithmes et applications (couplages, transbordement, ...)	18

**Laboratoire:** 32 périodes

- Modélisation et résolution de problèmes	12
- Développement/adaptation d'algorithmes, réalisation informatique	20

### Bibliographie

- Ravindra K. Ahuja, Thomas L. Magnanti, James B. Orlin, Network Flows: Theory, Algorithms, and Applications, 1993, Prentice Hall.
- Thomas Cormen, Charles Leiserson, Ronald Rivest, Clifford Stein, Introduction à l'algorithmique, 2002, Sciences sup, Dunod, Paris.
- Reinhard Diestel, Graph Theory, 2012, Springer-Verlag, Heidelberg.
- Michel Gondran, Michel Minoux, Graphes et algorithmes, 2009, Lavoisier, Cachan.
- Jon Kleinberg, Éva Tardos, Algorithm Design, 2005, Pearson.
- Robert Sedgewick, Kevin Wayne, Algorithms, 2011, Addison-Wesley.

### 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 de rendus de codes ou de pseudocodes, de rapports de modélisation ou de résultats d'exécution, 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 0.4 + moyenne laboratoire x 0.1 + moyenne examen x 0.5