

Département d'informatique
IFT436 — Algorithmes et structures de données

Plan de cours

Enseignant :

Courriel :
Site du cours :
Disponibilité :

Professeur responsable :

Horaire :

Description officielle de l'activité pédagogique¹

Cible(s) de formation	Comprendre le rôle des structures de données et des stratégies de conception dans la création d'algorithmes. Déterminer la complexité de calcul d'algorithmes à l'aide d'outils mathématiques.
Contenu	Outils mathématiques pour l'analyse de complexité algorithmique : analyse combinatoire, séries géométriques et résolution d'équations de récurrence. Notations asymptotiques. Utilisation d'assertions. Stratégies de conception : force brute, gloutonne, inductive, diviser-pour-régner, programmation dynamique, recherche dans un espace d'états. Illustration des concepts avec des algorithmes variés.
Crédits	3
Organisation	3 heures d'exposé magistral par semaine 1 heure d'exercices par semaine 5 heures de travail personnel par semaine
Préalable	IFT 339
Particularités	Cette activité pédagogique se présente sous la forme d'un cours.

¹ <http://www.usherbrooke.ca/fiches-cours/ift436>

1 Présentation

Cette section présente les objectifs spécifiques et le contenu détaillé de l'activité pédagogique. Cette section, non modifiable sans l'approbation d'un comité de programme du Département d'informatique, constitue la version officielle.

1.1 Mise en contexte

Un peu d'histoire...

Bien avant l'apparition des ordinateurs vers l'an 1945, les humains ont imaginé des séquences d'opérations enchaînées selon une procédure fixée à l'avance. Vers 300 av. J.-C., Euclide a décrit, dans le livre VII des Éléments, un algorithme permettant de déterminer le plus grand commun diviseur de deux entiers. Vers le milieu du septième siècle, des mathématiciens indiens sont parvenus à convertir des procédures de calcul en algorithmes applicables à des nombres abstraits de n'importe quelle taille. Au fil des siècles suivants, les savants ont inventé toutes sortes de machines afin d'automatiser des algorithmes car, il faut bien le dire, calculer avec les moyens de l'époque était long et fastidieux. Certaines de ces inventions furent des échecs, d'autres ont mené aux ordinateurs modernes. Aujourd'hui, l'algorithme est une des activités fondamentales de l'informatique.

À propos de la place de cette activité pédagogique dans votre programme...

L'activité pédagogique intitulée Algorithmes et structures de données appartient à la chaîne de cours analyse et programmation, où elle apparaît après IFT 159 et IFT339. Dans ces deux derniers cours, l'étudiante ou l'étudiant a appris à écrire et à implémenter des programmes pour effectuer certaines tâches, ainsi qu'à structurer les données afin que ces programmes soient plus efficaces. Dans le cours IFT 436, l'étudiante ou l'étudiant porte principalement son attention au travail d'analyse qui précède la programmation. La notion d'efficacité est formalisée, afin de pouvoir établir des comparaisons significatives entre les diverses solutions algorithmiques qui peuvent exister pour un même problème. Un des points fondamentaux consiste à mettre en évidence le fait que concevoir un programme équivaut à solutionner un problème abstrait, et que savoir identifier et formuler ce problème permet de chercher des solutions efficaces dans les références techniques ou, le cas échéant, d'en construire soi-même. Dans ce but, diverses stratégies de conception d'algorithmes sont présentées et illustrées avec des problèmes abstraits qui figurent parmi les plus courants de la pratique.

1.2 Objectifs spécifiques

À la fin de cette activité pédagogique, l'étudiante ou l'étudiant sera capable :

1. d'analyser un algorithme et de déterminer son temps de calcul en notation asymptotique;
2. de comprendre et d'utiliser les principales stratégies de conception d'algorithmes;
3. de comprendre le rôle des structures de données dans la conception d'algorithmes;
4. de comparer des algorithmes selon des critères d'efficacité consacrés par la pratique;
5. de comprendre des algorithmes pour divers types d'applications;
6. de mettre en pratique des stratégies de conception d'algorithmes.

L'expression comprendre un algorithme signifie être capable d'identifier les situations où l'emploi d'un algorithme donné est approprié et de modifier un algorithme pour l'adapter au contexte particulier dans lequel il est utilisé.

1.3 Contenu détaillé

Le tableau suivant contient la matière présentée dans ce cours. L'ordre de présentation est laissé à la discrétion de l'enseignante ou l'enseignant.

Les thèmes écrits en italiques peuvent, au choix de l'enseignante ou l'enseignant, être présentés dans des chapitres qui leur sont spécifiquement consacrés, ou bien être abordés à l'intérieur d'un chapitre consacré à un autre thème.

Thème	Contenu	Heures	Objectifs	Références
1	Introduction : présentation du plan de cours et du domaine.	2		chap. 1 de [1]
2	<i>Analyse des algorithmes : rappels de notions de mathématiques discrètes; notations asymptotiques; analyse des algorithmes itératifs.</i>	8	1,4	chap. 2 et 3 de [1]
3	<i>Réurrences : analyse des algorithmes récursifs, théorème maître.</i>	8	1	chap. 4 de [1]
4	Stratégies de conception des algorithmes : approche inductive, gloutonne, diviser-pour-régner, programmation dynamique; application des stratégies à un même problème abstrait.	12	1, 2, 4, 5	chap. 33 et 9 de [1]
5	Algorithmes pour la sélection et le tri : exemples d'algorithmes de tri; algorithme pour la sélection (médiane).	8	1, 3, 4, 5, 6	chap. 6 et 20 de [1]
6	Introduction à la théorie des graphes: graphes orientés et non orientés, arbres; accessibilité, composantes connexes; représentation des graphes.	2		
7	Algorithmes pour les graphes : algorithmes de base; algorithmes de parcours de graphe; tri topologique, calcul de l'arbre de recouvrement minimal.	6	1, 3, 4, 5, 6	chap. 21 à 24 de [1]
8	Filtrage de chaînes de caractères : exemples d'algorithmes de filtrage.	4	1, 4, 5, 6	chap. 32 de [1]

2 Organisation

Cette section propre à l'approche pédagogique de chaque enseignante ou enseignant présente la méthode pédagogique, le calendrier, le barème et la procédure d'évaluation ainsi que l'échéancier des travaux. Cette section doit être cohérente avec le contenu de la section précédente.

2.1 Méthode pédagogique

Une semaine comporte quatre heures de présence en classe réparties dans une proportion de trois heures de cours magistral et d'une heure d'exercices. Les exercices sont faits par les étudiantes ou les étudiants, autant que possible à la fin d'un thème. Ainsi, dans une semaine, il est possible qu'il n'y ait aucune séance d'exercices, mais que dans la semaine suivante il y ait une séance d'exercices de deux heures.

Bien qu'il s'agisse d'un cours théorique, les étudiantes et les étudiants sont fortement encouragés à programmer les algorithmes vus en cours, afin de vérifier qu'ils comprennent bien leur fonctionnement.

2.2 Calendrier

Semaine	Thèmes	Lecture	Exercices	Devoirs
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15-16				

2.3 Évaluation

Devoirs :

Examen périodique :

Examen final :

Qualité du français :

Les appareils électroniques (calculatrice, portable, téléphone cellulaire) sont interdits pendant les examens. Conformément à l'article 17 du Règlement facultaire d'évaluation des apprentissages :

<http://www.usherbrooke.ca/sciences/intranet/informations-academiques/reglement-d-evaluation/>

L'enseignant peut retourner à l'étudiante ou à l'étudiant tout travail non conforme aux exigences quant à la qualité de la langue et aux normes de présentation.

Toute situation de plagiat sera traitée en conformité, entre autres, avec l'article 8.1.2 du *Règlement des études* de l'Université de Sherbrooke disponible à l'adresse :

<http://www.usherbrooke.ca/programmes/references/reglement>

Vous trouverez en annexe un document d'information relatif à l'intégrité intellectuelle qui fait état de l'article 8.1.2.

2.4 Échéancier des travaux

Les directives, la date de remise et le barème relatifs aux devoirs seront connus à la remise de l'énoncé de chaque devoir aux étudiantes et aux étudiants.

Directives particulières : Un petit devoir est donné chaque semaine ou aux deux semaines aux étudiantes et aux étudiants pour qu'ils assimilent les concepts présentés dans le cours. Les devoirs peuvent être faits individuellement ou par équipe d'au plus deux personnes. Les devoirs comportent l'écriture de programmes en langage de programmation C++. Aucun devoir ne peut être remis par courrier électronique. Les devoirs plagiés ou non remis reçoivent automatiquement la note zéro.

La correction des devoirs et des examens est entre autres basée sur le fait que chacune de vos réponses soit :

- claire, c'est-à-dire lisible et compréhensible pour le correcteur;
- précise, c'est-à-dire exacte ou sans erreur;
- complète, c'est-à-dire que toutes les étapes de résolution du problème sont présentes;
- concise, c'est-à-dire que la méthode de résolution est la plus courte possible.

3 Matériel nécessaire pour le cours

Aucun manuel n'est obligatoire. Tous les programmes et les diapositives présentés dans le cours sont disponibles dans le répertoire public *Public/Cours* dont l'accès est décrit dans la page Web :

<http://www.usherbrooke.ca/informatique/intranet/ressources-et-documentation/faq/acces-aux-lecteurs-reseaux/>