

COORDONNATEUR      WALSH, Timothy      walsh.timothy@uqam.ca      (514) 987-3000 6139      PK-4735

GROUPE      10 TREMBLAY, Guy      tremblay.guy@uqam.ca      (514) 987-3000 8213      PK-4435  
 Lundi, de 13h30 à 15h00 et mercredi, de 11h00 à 12h30 (cours) - Mercredi, de 8h30 à 10h30 (ateliers)

DESCRIPTION      Connaître les algorithmes de base de l'informatique. Être capable d'analyser leur complexité temporelle et spatiale. Connaître les grands principes de la conception des algorithmes et la programmation dynamique. Être capable d'appliquer ces principes. Comprendre la notion de problème NP-complet. Notations asymptotiques.  
 Opérations sur les notations asymptotiques. Équations de récurrence asymptotiques. Résolution de récurrences. Algorithmes et heuristiques voraces. Applications au problème de l'arbre de recouvrement minimal et à l'ordonnancement des travaux. Principe «diviser pour régner» (sélection en temps linéaire, arithmétique des grands entiers, calcul des nombres de Fibonacci, etc.). Principe de la programmation dynamique (multiplication chaînée de matrices, arborescences de fouille optimale, etc.). Exploration de graphes. Algorithmes à retour arrière. Introduction aux graphes de recherche de chaînes. Introduction à la NP-complétude.  
 Ce cours comporte une séance hebdomadaire de deux heures de travaux en laboratoire.  
 Préalables: INF3105 Structures de données et algorithmes

OBJECTIFS      Le cours vise à initier les étudiant-e-s aux principes de base de la conception et de l'analyse des algorithmes séquentiels. À la fin du cours, l'étudiant-e devrait être capable :

- de comprendre et d'expliquer les principales caractéristiques des algorithmes de base de l'informatique et des principales stratégies de conception d'algorithmes ;
- d'analyser la complexité et l'efficacité de différents types d'algorithmes ;
- d'expliquer les limites de l'analyse asymptotique et le rôle que peuvent jouer les outils de mesure de performances des programmes ;
- de concevoir et de mettre en oeuvre, dans un langage de programmation séquentiel, des algorithmes pour divers types de problèmes qui soient efficaces, tant de façon asymptotique que de façon empirique.

ÉVALUATION	Description sommaire	Date	Pondération
	Examen intra (à livre ouvert)	Mercredi (date à déterminer)	35%
	Examen final (à livre ouvert)	Mercredi 23 avril 2008	35%
	Trois (3) devoirs		30%

Les devoirs de programmation seront faits avec le langage MPD sur le serveur arabica.  
 Les examens sont **individuels** et les devoirs seront faits **en équipes comportant au plus deux étudiants**. Il y a 10% de pénalité par jour ouvrable de retard dans la remise des devoirs. Les règlements concernant le plagiat seront strictement appliqués.  
 L'utilisation de documentation personnelle (notes de cours, manuels) est permise à l'examen.  
 Une moyenne d'au moins 50 % aux examens est exigée pour réussir le cours.

**Politique d'absence aux examens**

Un étudiant absent à un examen se verra normalement attribuer la note zéro pour cet examen. Cependant, si l'étudiant était dans l'impossibilité de se présenter à l'examen pour un motif valable, certains arrangements pourront être pris avec son enseignant. Pour ce faire, l'étudiant devra présenter à son enseignant l'un des formulaires prévus à cet effet accompagné des pièces justificatives appropriées (par ex., attestation d'un médecin que l'étudiant était dans l'impossibilité de se présenter à l'examen pour des raisons de santé, lettre de la Cour en cas de participation à un jury).

Une absence pour cause de conflit d'horaires d'examen n'est pas considérée comme un motif valable d'absence, à moins d'entente préalable avec la direction du programme et l'enseignant durant la période d'annulation des inscriptions avec remboursement : tel qu'indiqué dans le guide d'inscription des étudiants, il est de la responsabilité d'un étudiant de ne s'inscrire qu'à des cours qui ne sont pas en conflit d'horaire.

Pour plus de détails sur la politique d'absence aux examens du Département d'informatique et pour obtenir les formulaires appropriés, consultez le site web suivant :  
<http://www.info.uqam.ca/enseignement/politiques/absence-examen>

CONTENU       Rappels mathématiques : sommation, puissance, logarithme.  
 Notions de base : taux de croissance des fonctions et notations asymptotiques ; analyse de l'efficacité

- d'algorithmes (tous les cas, pire cas) ; équations de récurrence et techniques de résolution : méthode par substitution, théorème général.
- ❑ Introduction au langage MPD (à l'aide d'exemples divers).
  - ❑ Stratégie "diviser pour régner" : fouille binaire, tri par fusion et tri rapide, algorithme de Strassen pour la multiplication de matrices ; patron générique de conception "template" pour l'approche diviser-pour-régner.
  - ❑ Programmation dynamique : mémorisation et approches descendante (récursive) vs. ascendante (itérative) ; nombres de Fibonacci, multiplication de chaînes matrices, sac à dos 0-1, distance d'édition.
  - ❑ Algorithmes voraces : arbre de recouvrement minimal, sac à dos fractionnaire.
  - ❑ Algorithmes déterministes d'exploration d'espaces combinatoires : marche arrière ("backtracking"), séparation et évaluation progressive ("branch-and-bound").
  - ❑ Ingénierie d'algorithmes : limites de l'analyse asymptotique ; analyse expérimentale ; optimisation et figlage de code ; profilage de code.
  - ❑ Sujets divers (selon le temps disponible)
    - Approches diverses pour le problème du commis voyageur ;
    - Théorie de la complexité et NP-complétude.

## RÉFÉRENCES

- NO Notes de cours de Guy Tremblay, en vente à la COOP.
- UO Site web du professeur: – <http://www.info2.uqam.ca/~tremblay/INF4100>
- VR Neapolitan, R. et Naimipour, K. – *Foundations of Algorithms Using C++ Pseudocode (Third Edition)* – Jones and Bartlett Publishers, 2004.  
En vente à la COOP.
- VC Aho, A.V., Hopcroft, J.E., Ullman, J.D. – *Data Structures and Algorithms* – Addison-Wesley, 1983.
- VC Aho, A.V., Ullman, J.D. – *Foundations of Computer Science* – Computer Science Press, 1992.
- VC Baase, S. – *Computer Algorithms: Introduction to the Design and Analysis of Algorithms* – (3e édition), Addison-Wesley, 2000.
- VC Brassard, G., Bratley, P. – *Fundamentals of Algorithmics* – Prentice-Hall, 1996.
- VC Brassard, G., Bratley, P. – *Algorithmique: conception et analyse* – Masson, 1987.
- AC Cattaneo, G. and Italiano, G. – *Algorithm engineering* – ACM Computing Surveys, 31(3es), 1999. Article No. 3.
- VC Cormen, T., Leiserson, C. et Rivest, R. – *Introduction à l'algorithmique* – Dunod, 2002 (version révisée en 2002).
- VC Goodrich, M.T. and Tamassia, T. – *Data Structures and Algorithms in Java* – John Wiley & Sons, 1998.
- VC Graham, R.L., Knuth, D.E., Patashnik, O. – *Concrete Mathematics: a Foundation for Computer Science* – Addison-Wesley, 1994.
- VC Harel, D. – *Algorithmics, The Spirit of Computing* – Addison-Wesley, 1987.
- VC Johnsonbaugh R. and Schaefer, M. – *Algorithms* – Pearson Education, 2004.
- VC Loukides, M. and Oram, A. – *Programming with GNU Software* – O'Reilly, 1997.
- CC Moret, B.M.E. – *Towards a discipline of experimental algorithmics. In Proc. 5th DIMACS Challenge, volume DIMACS Monographs 59, pages 197-213* – American Mathematical Society, 2002.
- VC Rosen, K.H. – *Discrete Mathematics and its Applications* – 1995 (version révisée en 1999).
- VC Sedgewick, R. – *Algorithms (2e édition)* – Addison-Wesley, 1988.
- VC Weiss, M.A. – *Data Structures and Algorithm Analysis* – Benjamin/Cummings, 1992.
- VC Zeller, A. and J. Krinke, J. – *Essential Open Source Toolset* – John Wiley & Sons, Ltd, Chichester, UK, 2005.

A : article – C : comptes rendus – L : logiciel – N : notes – R : revue –  
S : standard – U : uri – V : volume

C : complémentaire – O : obligatoire – R : recommandé