

INF7440–40  
Conception et analyse des algorithmes  
Automne 2007

---

**Professeur :** Guy Tremblay

**Local :** PK-4435

**Téléphone :** 987-3000 poste 8213# (boîte vocale)

**Courriel :** tremblay.guy@uqam.ca

**Site web :** <http://www.info.uqam.ca/~tremblay/INF7440>

**Horaire du cours :** Jeudi, 18:00–21:00

---

## 1 Description *(selon l'annuaire)*

Rappels sur l'analyse des algorithmes : notations asymptotiques, types d'analyse (pire cas, cas moyen), équations de récurrence et techniques de résolution. Stratégies de conception d'algorithmes séquentiels (diviser pour régner, programmation dynamique, algorithmes voraces) : algorithmes déterministes d'exploration d'espaces combinatoires (marche arrière, avec séparation et évaluation progressive). Sujets divers : Algorithmes parallèles, algorithmes probabilistes (méthode Monte-Carlo, chaînes de Markov), heuristiques et algorithmes d'approximation pour problèmes difficiles.

---

## 2 Objectifs

### Généraux :

Le cours vise à initier les étudiant-e-s aux principes de base de la conception et de l'analyse des algorithmes, tant séquentiels *que parallèles*.

### Spécifiques :

À la fin du cours, l'étudiant-e devrait être capable ...

- de comprendre et d'expliquer les principales caractéristiques des différents modèles de machines et d'algorithmes séquentiels et parallèles ;
  - d'analyser la complexité et l'efficacité de différents types d'algorithmes, tant séquentiels que parallèles ;
  - de concevoir, en utilisant un certain nombre de stratégies de base, des algorithmes efficaces, séquentiels ou parallèles, pour divers types de problèmes ;
  - de coder un algorithme parallèle à l'aide d'un langage simple de programmation parallèle.
-

### 3 Contenu du cours

1. Rappels des notions base : algorithme, analyse d'algorithmes, notation asymptotique, résolution d'équations de récurrence. Introduction au langage MPD.
2. Techniques et stratégies séquentielles pour la résolution exacte de problèmes «non-difficiles»
  - Stratégie «diviser pour régner» (par ex., fouille binaire, tri par fusion et tri *quicksort*, algorithme de Strassen pour la multiplication de matrices) ;
  - Programmation dynamique (par ex., multiplication de chaînes matrices, sac à dos, distance d'édition) ;
  - Algorithmes et heuristiques voraces (par ex., arbre de recouvrement minimal, sac à dos) ;
  - Algorithmes avec *backtracking* et *branch-and-bound* (par ex., voyageur de commerce) ;
3. Introduction aux algorithmes parallèles
  - Notions de base sur les architectures parallèles : multi-processeurs vs. multi-ordinateurs ; communication, latence et fils d'exécution multiples ;
  - Modèles de calcul parallèle et stratégies de conception d'algorithmes parallèles : parallélisme itératif, parallélisme récursif, filtres et pipelines, parallélisme de données, sac de tâches ;
  - Langage MPD (suite) : création de processus, synchronisation, ressources ;
  - Mesures de performance et analyse des algorithmes parallèles : temps d'exécution et profondeur ; coût, travail et optimalité ; accélération et efficacité ;
  - Algorithmes pour le modèle PRAM :
    - Caractéristiques et variantes du modèle PRAM ;
    - Algorithmes sur les listes et les arbres : recherche du minimum, préfixes parallèles, calcul du rang, circuit eulérien ;
    - Simulations entre modèles PRAM ;
4. Algorithmes et heuristiques pour les problèmes difficiles
  - Algorithmes d'approximation et heuristiques déterministes ;
  - Algorithmes probabilistes ;
  - Métaheuristiques (par ex., recuit simulé, algorithmes évolutionnaires, colonies de fourmis) ;
5. Sujets divers (selon le temps disponible)
  - Analyse expérimentale d'algorithmes séquentiels et parallèles ;
  - Algorithmes de tri ;

---

## 4 Évaluation

### \* Examens :

- Un examen (à livre ouvert) : 35 %

### \* Devoirs :

- Trois (3) devoirs (trois meilleurs parmi quatre) : 40 %

### \* Travail de fin de session

- Conception, mise en oeuvre, analyse et comparaison d'algorithmes séquentiel et parallèles pour un problème choisi par l'étudiant : 25 %

---

L'utilisation de documentation personnelle (notes de cours, manuels) est permise à l'examen. Une moyenne d'au moins 50 % à l'examen est exigée pour réussir le cours.

Les devoirs et le travail de fin de session peuvent être réalisés individuellement ou en équipe de deux (2) personnes. Par contre, l'examen doit évidemment être fait de façon individuelle ;)

Aucun retard ne sera accepté pour les devoirs (choix des trois meilleurs résultats parmi quatre). La pénalité de retard pour le travail final est de 10 % par jour (ouvrable).

La qualité du français sera prise en considération, tant dans les exercices que dans le travail écrit (jusqu'à 10 % de pénalité).

Le sujet du travail de fin de session devra être approuvé par le professeur. Des sujets vous seront proposés ultérieurement.

---

## 5 Références

### Manuel de base (recommandé)

[NN04] R. Neapolitan and K. Naimipour. *Foundations of Algorithms Using C++ Pseudocode (Third Edition)*. Jones and Bartlett Publishers, 2004.

En vente à la COOP (87,25 \$ pour les membres).

### Notes de cours

- Des notes de cours (format PDF) seront disponibles sur le site web du cours :

<http://www.labunix.uqam.ca/~tremblay/INF7440/Materiel>

## Bibliographie

- [AHU83] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *Data Structures and Algorithms*. Addison-Wesley, 1983. [QA76.9D35A56].
- [AK89] E. Aarts and J. Korst. *Simulated Annealing and Boltzmann Machines*. John Wiley & Sons, Chichester, UK, 1989. [QA402.5A27].
- [And00] G.R. Andrews. *Foundations of Multithreaded, Parallel, and Distributed Programming*. Addison-Wesley, Reading, MA, 2000. [QA76.58A57 2000].
- [Bv00] S. Baase and A. van Gelder. *Computer Algorithms (Third Edition)*. Addison-Wesley, 2000. [QA76.9A43B33].
- [CLR94] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction à l'algorithmique*. Dunod, 1994. [QA76.6C65614].
- [CT93] M. Cosnard and D. Trystram. *Algorithmes et architectures parallèles*. InterEditions, 1993. [QA76.58C68].
- [DPST03] J. Dréo, A. Pétrowski, P. Siarry, and É. Taillard. *Métaheuristiques pour l'optimisation difficile*. Eyrolles, 2003. [T57.84M48].
- [Fos95] I. Foster. *Designing and Building Parallel Programs*. Addison-Wesley, 1995. <http://www-unix.mcs.anl.gov/dbpp>.
- [GGKK03] A. Grama, A. Gupta, G. Karypis, and V. Kumar. *Introduction to Parallel Computing (Second Edition)*. Addison-Wesley, 2003.
- [GUD96] M. Gengler, S. Ubéda, and F. Desprez. *Initiation au parallélisme—Concepts, architectures et algorithmes*. Masson, 1996. [QA76.58G45].
- [Han95] P.B. Hansen. *Studies in Computational Science—Parallel Programming Paradigms*. Prentice-Hall, Englewood Cliffs, NJ, 1995.
- [JaJ92] J. JaJa. *An Introduction to Parallel Algorithms*. Addison-Wesley Publishing Company, 1992. [QA76.58J34].
- [KKT01] J. Keller, C. Kessler, and J. Traff. *Practical PRAM Programming*. John Wiley & Sons, Inc., 2001.
- [LR03] A. Legrand and Y. Robert. *Algorithmique parallèle*. Dunod, 2003. [QA76.642L44].
- [MB00] R. Miller and L. Boxer. *Algorithms Sequential & Parallel*. Prentice-Hall, 2000. [QA76.9A43M55].
- [Mic93] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 1993. [QA76.9A43M53].
- [Roo00] S.H. Roosta. *Parallel Processing and Parallel Algorithms — Theory and Computation*. Springer-Verlag, 2000.