

UNIVERSITÉ DE MONTRÉAL
DÉPARTEMENT D'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE
IFT 2125 — Introduction à l'algorithmique — A07

Professeur : Gilles Brassard, 2215 André-Aisenstadt, brassard@iro.umontreal.ca

Horaire des cours : lundi 16h30–17h30, AA-1177 ; mardi 15h30–17h30, AA-1409

Premier cours : le mardi 4 septembre 2007, 15h30, AA-1409

Objectifs : Comment développer un algorithme efficace pour résoudre un problème donné? Parmi plusieurs algorithmes résolvant un même problème, lequel choisir? Pour illustrer l'importance de ces questions, considérez le problème du calcul du déterminant. Un algorithme classique, dû à Gauss et Jordan, permet de calculer le déterminant d'une matrice 20×20 en une fraction de seconde sur un ordinateur contemporain. Un autre algorithme tout aussi classique, basé sur la définition récursive du déterminant, prendrait des milliers d'*années* pour arriver au même résultat!

L'algorithmique propose des réponses à ces questions. Pour la première, il y a un ensemble de techniques générales de conception d'algorithmes. Nous étudierons par exemple l'approche vorace, la technique diviser-pour-régner, la programmation dynamique et l'approche probabiliste. Pour la seconde question, l'algorithmique offre des techniques d'analyse de l'efficacité d'algorithmes basées principalement sur la résolution de récurrences et les notations asymptotiques. À l'aide de ces méthodes, il est possible de prédire la quantité de temps ou de mémoire requise à l'exécution d'un algorithme sur des exemplaires de grande taille du problème à résoudre. Cette analyse constitue une base de comparaison pour guider le choix de l'algorithme.

Le cours IFT 2125 vous permettra d'apprendre à concevoir des algorithmes, d'analyser l'efficacité de ceux-ci et de vous familiariser avec des techniques mathématiques pertinentes. Vous développerez le réflexe de ne pas vous contenter de la première méthode trouvée mais plutôt de chercher l'algorithme le plus efficace possible pour résoudre le problème auquel vous serez confrontés.

Évaluation : Le cours ne demande pas de programmation. Il y aura un examen partiel, un examen final cumulatif et un certain nombre d'exercices théoriques.

Barème (avec seuil) :

Examen partiel	30%	(le mardi 16 octobre, 15h30–17h20, Z-220)
Examen final	40%	(le mardi 18 décembre, 15h30–18h20, AA-1355)
Exercices	30%	

Le seuil : Pour que les exercices comptent dans la note finale, vous devez obtenir une moyenne pondérée d'au moins 40% aux examens.

Livre obligatoire : Gilles Brassard et Paul Bratley, *Fundamentals of Algorithmics*, Prentice-Hall, 1996. **Note :** Certaines sections du livre correspondent à des sujets que vous devriez déjà bien connaître. Celles-ci sont indiquées ci-dessous comme *lecture libre* et il incombera à chacun de s'assurer de la maîtrise de cette matière. Si nécessaire, quelques-unes de ces notions pourront être survolées en démonstration.

Autres livres utiles : Thomas H. Cormen, Charles E. Leiserson et Ronald L. Rivest, *Introduction à l'algorithmique*, 2e édition, Dunod, 2002. Alfred V. Aho, John E. Hopcroft et Jeffrey D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974. Dexter C. Kozen, *The Design and Analysis of Algorithms*, Springer-Verlag, 1992.

Plan approximatif et quelque peu spéculatif du cours :

- ◇ 1ère semaine: 4 septembre. Motivations: §§1.1, 1.2, 7.1; Introduction: Chapitre 2.
[*Lecture libre*: §§1.3, 1.4, 1.5, 1.6, 1.7 (sauf §1.7.4).]
- ◇ 2ième semaine: 10 et 11 septembre. Notation asymptotique: Chapitre 3.
- ◇ 3ième semaine: 17 et 18 septembre. Résolution de récurrences: §4.7.
- ◇ 4ième semaine: 24 et 25 septembre. Résolution de récurrences (suite): §4.7.
Analyse d'algorithmes: §§4.1–4.5.
- ◇ 5ième semaine: 1 et 2 octobre. Algorithmes voraces: §§6.1, 6.2, 6.3, 6.5.
[*Lecture libre*: §§5.1–5.5, 5.7, 6.4.]
- ◇ 6ième semaine: 9 octobre. Diviser-pour-régner: §§7.1 (rappel), 7.2, 7.4.
- ◇ 7ième semaine: 15 octobre. Diviser-pour-régner (suite): §§7.6, 7.7.
Examen partiel le 16 octobre;
- ◇ 8ième semaine: 29 et 30 octobre. Cryptographie: §7.8;
Programmation dynamique: §§8.1.1, 8.2, 8.3.
- ◇ 9ième semaine: 5 et 6 novembre. Programmation dynamique (suite): §§8.4, 8.6, 8.7, 8.8.
[*Lecture libre*: §8.5.]
- ◇ 10ième semaine: 12 et 13 novembre. Graphes de jeu: §9.1;
Parcours de graphes: §§9.3, 9.4, 9.5. [*Lecture libre*: §9.2.]
- ◇ 11ième semaine: 19 et 20 novembre. Graphes implicites et retour arrière: §9.6;
Algorithmes probabilistes: §§10.5.1, 10.7.1, 10.1, 10.2, 10.3, 10.4, 10.5.2.
- ◇ 12ième semaine: 26 et 27 novembre. Algorithmes probabilistes (suite): §10.6.
- ◇ 13ième semaine: 3 et 4 décembre. Algorithmes probabilistes (suite): §10.7;
Introduction à la complexité: §§12.1, 12.2.
- ◇ 14ième semaine: 10 décembre. Introduction à la complexité (suite): §12.4.