IFT 2121

INTRODUCTION A L'ALGORITHMIQUE

Hiver 2006 Geňa Hahn

Comment développer un algorithme efficace pour résoudre un problème donné? Parmi plusieurs algorithmes résolvant un problème, lequel choisir? L'algorithmique propose certaines réponses à ces questions. Pour la première, il y a un bagage de techniques de conception d'algorithmes (par exemple les techniques gloutones, diviser-pour-régner et la programmation dynamique, sans compter des solutions parfois ingénieuses qui ne sont pas des applications de techniques générales mais qui sont taillées sur mesure, selon le problème particulier à résoudre). Pour la seconde question, l'algorithmique offre principalement l'analyse asymptotique du temps d'exécution d'un algorithme. A l'aide de méthodes mathématiques, 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. Ces prédictions constituent une base de comparaison qui pourra guider le choix d'un algorithme ou d'un autre.

Le cours IFT2121 permettra à l'étudiant(e) d'apprendre à concevoir des algorithmes, d'analyser l'effacité de ces algorithmes, de se familiariser avec certaines techniques mathématiques, et de développer, de manière générale, un réflexe, celui de ne pas se contenter de la première méthode trouvée mais plutôt de chercher la méthode la plus efficace pour résoudre un problème de calcul donné.

Plan du cours (esquisse: les chapitres mentionnés sont ceux du livre obligatoire; l'ordre peut changer; les chapitres 1 et 2 sont surtout des chapitres de référence - on suppose leur contenu connu, sauf exception):

1. Introduction, motivation, bases (Chapitre 2)
2. Outils pour l'analyse d'efficacité (Chapitres 3–4)
3. Algorithmes gloutons (Chapitre 6)
4. Diviser-pour-régner (Chapitre 7)
5. Programmation dynamique (Chapitre 8)
6. Algorithmes pour des graphes (Chapitre 9)
7. Algorithmes probabilistes (Chapitre 10)
8. Divers (Chapitres 11–12, transformée de Fourrier rapide, introduction à l'optimisation combinatoire, etc., suivant le temps)

Evaluation: Il y aura **deux (2)** intras d'une heure chaque, un final (ouff) de trois heures et beaucoup de petits exercices théoriques à raison d'un devoir par semaine sauf exception (autour des deux intras). Probablement pas de programmation. Les devoirs sont à remettre au plus tard au début du TP le vendredi de remise. Etant donné que les solutions seront données en TP, aucun devoir ne sera accepté en retard. *Vous êtes vivement encouragés à faire vos devoirs en groupes de deux et à remettre une seule copie par groupe de deux.* Toute solution copiée (d'un livre, des amis, de l'internet ou ailleurs) sans citation de source mérite 0 si découverte (plagiat répété mène directement à un F pour le cours).

Barèmes et dates (préliminaires) : SEUIL 45% (si la moyenne pondérée des examens n'atteint pas 45%, la note des devoirs est divisée par deux). Le seuil est atteint automatiquement par une note de 70% ou plus au final.

1er intra	10%	le 9 février
2ème intra	15%	le 23 mars
final	40%	lundi le 24 avril, 11h30 – 14h 30, AA 1360
devoirs	35%	à rendre le vendredi sauf le 10.2. et le 24.3.

Livre obligatoire: Gilles Brassard, Paul Bratley, *Fundamentals of Algorithms*, Prentice-Hall, 1996

Autres livres suggérés(la plupart en réserve à la bibliothèque) :

Alfred Aho, John Hopcrooft, Jeffrey Ullman, *The design and analysis of computer algorithms*, Addison-Wesley, 1974
Sara Baase *Computer Algorithms: Introduction to design and analysis*, Addison-Wesley, 1983
Donald Knuth, *The art of Computer programming*, Addison-Wesley, 1981
Thomas Cormen , Charles Leiserson , Ronald Rivest , Clifford Stein , *Introduction l'algorithmique*, Dunod, 2002

Premier cours le 9 janvier, premier TP le 13 janvier