

Tous les deux ou trois ans, dans les milieux “autorisés”, une même idée rejaillit du fin fond du cerveau d’un éminent pédagogue : il faut rénover le Deug!! Cette brillante suggestion se traduit par une renumérotation de l’ensemble des modules, des horaires modifiés mais finalement un contenu qui change très peu! Par contre pour égérer encore un peu plus les étudiants dans la jungle de l’université, l’opération est une véritable réussite. Ce document correspond à l’ex-module I31 dans lequel 30h de cours était donné au lieu des 24h actuels. C’est pourquoi vous trouverez toute une partie sur les tableaux, sujet qui n’a pas été traité en I32. Ce n’est pas grave, cette partie vous permettra de vous remémorer l’ensemble des algorithmes classiques à connaître sur les tableaux et de voir une applicaton de la preuve de programmes.

Ce document n’est pas à proprement dit un polycopié de cours pour l’actuel module “Preuves et Analyses d’algorithmes”. Tout comme un livre, souvent un polycopié n’est pas lu de façon attentive car il contient énormément d’informations, certaines étant beaucoup plus importantes que d’autres. La plupart du temps l’étudiant (toujours très attentif pendant le cours!) essaie alors à partir du polycopié et du spectacle hebdomadaire des 4 guignols d’en extraire les résultats essentiels afin de fabriquer des mini-fiches “résumés”. Malheureusement parfois ces fiches sont plus que succinctes!!

Ce document essaie de se placer à mi-chemin entre le livre trop complet et la fiche de l’étudiant trop réduite. Il contient les résultats fondamentaux à connaître à la fin de ce module, et autant que possible chaque notion présentée est illustrée par un ou plusieurs exemples. Il est divisé en un ensemble de “mémos”, chacun d’entre eux essayant de faire le point sur une notion présentée en cours.

L’un des objectifs de ce cours est d’introduire une certaine notion de formalisme au niveau des algorithmes afin de pouvoir répondre à une question légitime de tout utilisateur d’un logiciel : “Ce produit fait-il correctement ce que j’en attends?”. Cette question, en apparence très simple, entraîne cependant du point de vue de l’informatique tout un ensemble de problèmes, à tel point que sur certains logiciels on peut trouver l’avertissement suivant :

(extrait de la convention de droits d’utilisation d’un produit très connu!)
“Le logiciel et la documentation qui l’accompagne sont fournis dans l’état où ils se trouvent et sans aucune garantie. En outre la société ne donne **aucune garantie** et ne fait aucune déclaration concernant l’emploi ou les résultats découlant de l’emploi du logiciel, ou de la documentation quant à leur **exactitude, fiabilité, actualité** ou autre”.

Accepteriez-vous une telle garantie de la part de votre garagiste ? Accepteriez-vous d’acheter du matériel Hi-Fi estampillé du label MP (“marche presque!”). Cependant force est de constater que de plus en plus de logiciels commerciaux mériteraient d’être estampillés du label MP! Dans toute activité scientifique, on utilise une abstraction de type mathématique (appelée modèle)

afin de représenter un phénomène réel. On analyse alors le modèle afin d'en déduire des informations quant aux déroulements d'expériences réelles. La logique va nous permettre d'élaborer un modèle (à l'aide des assertions) permettant d'étudier les algorithmes ainsi que leur relation avec les objets qu'ils manipulent. Nous serons alors capables de démontrer formellement qu'un algorithme calcule bien le résultat espéré (ce qui nous permettra de les estampiller du label FP : "fonctionne parfaitement").

L'informatique est prétexte à des absurdités en tout genre notamment dans le domaine cinématographique. L'ordinateur est une machine surpuissante capable de tout réaliser (notamment quand en plus on travaille dans le cyber-espace virtuel!!!!), il suffit de la programmer. Une idée fausse constamment véhiculée est qu'un ordinateur peut tout faire. Or l'informatique possède ses limites, et ceci reste peu enseigné aux non-spécialistes de l'informatique. Ce module ne suffirait pas à présenter tous les problèmes qui tournent autour de la possibilité ou non de résoudre un problème à l'aide d'un ordinateur. Cependant le deuxième objectif de ce cours est de vous sensibiliser à la complexité des algorithmes afin que vous preniez conscience que deux algorithmes réalisant le même travail peuvent être très différents en ce qui concerne leur temps d'exécution. Dans certains cas il ne faut donc pas se contenter de rédiger un algorithme, encore faut-il s'assurer que l'algorithme est "bien" écrit !



P.S : Et n'oubliez pas, l'anagramme d'algorithme est logarithme !