info505 : mathématiques pour l'informatique
TD 1 : complexité

Pierre Hyvernat
Laboratoire de mathématiques de l'université de Savoie
bâtiment Chablais, bureau 22, poste : 94 22
email : Pierre.Hyvernat@univ-savoie.fr
www : http://www.lama.univ-savoie.fr/~hyvernat/
wiki : http://www.lama.univ-savoie.fr/wiki/

Exercice 1 : classes de complexité

Question 1. À l'aide d'une calculatrice ou d'un ordinateur, remplissez le tableau suivant. La première colonne indique la complexité en microseconde d'un algorithme pour une entrée de taille n. Pour chaque colonne, estimez le temps d'exécution de cet algorithme pour une entrée de la taille donnée.

	5	10	20	40	100	500
n	5×10^{-6} s	10^{-5} s	2×10^{-5} s	4×10^{-5} s	10^{-4} s	5×10^{-4} s
$n\log(n)$						
n^2						
n^3						
n^5						
2^n						
3^n						
n^n						
2^{2^n}						

Question 2. Quelles sont les classes de complexité "utilisables" ?

Si on estime que la loi de Moore est valide (la puissance de calcul double tous les deux ans), quelles sont les classes de complexité qui peuvent basculer du "infaisable" dans le "raisonnable" ?

Exercice 2 : Un exemple complet

Question 1. Écrivez, dans un langage algorithmique de votre choix, l'algorithme du tri par sélection. (On cherche le minimum, puis le deuxième élément etc.)

Question 2. Calculer la complexité de ce tri en faisant attention aux détails.

Question 3. Si vous connaissez le tri fusion, essayer d'estimer sa complexité. (Utilisez la version récursive...)

Exercice 3 : Un exemple fondamental

Question 1. En cryptographie, on manipule régulièrement des nombres entiers de quelques centaines de chiffres. Il faut donc utiliser une structure de données différente du type `int` de votre langage favori. (Un `int` n'est stocké que sur un ou deux octets...)

Donnez un type de données que vous pourrez utiliser pour ces "grands entiers".

Question 2. Donnez les algorithmes de multiplication et addition de grands entiers. Quelles sont leurs complexités ?

Réfléchissez à la division et au modulo...

Question 3. Une autre opération fondamentale en cryptographie est la puissance. Étant donné un grand entier x et un autre grand entier n, on veut calculer le résultat de

$$x^n \pmod{m}$$

où m est un grand entier.

Essayez de programmer cette opération, et estimez sa complexité *en fonction du nombre d'opérations arithmétiques sur les grands entiers*.

Qu'en pensez-vous ? Cette opération est-elle utilisable sur des grands entiers ? Pouvez-vous l'améliorer ?