info614 : Mathématiques pour l'informatique
TD 0 : préliminaires

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 : Tours de Hanoi

Question 1. Hanoi généralisé. Peut-on remplacer les positions initiales et finales par des positions quelconques ? Est-ce que le nombre optimal de mouvements $(2^n - 1)$ est toujours valide pour cette variante ?

On considère maintenant la variante suivante : les positions de départ et d'arrivée sont identiques, mais les contraintes sont :
- on ne déplace qu'un disque à la fois
- on ne peut pas poser un disque sur un disque plus petit
- on ne peut pas faire de mouvement direct de A vers C (il faut le décomposer en $A \to B$ puis $B \to C$)

Question 2. résolvez le problème pour 1 et 2 disques. Combien de mouvements utilisez-vous ? Même question pour 3 disques...

Question 3. donnez une relation de récurrence pour calculer le nombre de mouvements que vous faites ; montrez que ce nombre est optimal.

Question 4. combien de mouvements utiliseriez-vous pour 4, 5 ou 6 disques ?

Question 5. résolvez cette récurrence.

Question 6. montrez que lors de la résolution de cette variante avec n disques, toutes les configurations valides (empilements de n disques sur trois emplacements) sont rencontrées.

Question 7. on revient maintenant à la règle originale, mais où chaque disque est dédoublé (on a donc $2n$ disques). Combien de mouvements faut-il pour transférer cette double tour de A vers C. (Les disques identiques peuvent être interchangés, et on peut les mettre les un sur les autres...)
Si vous trouvez que c'est trop facile, cherchez le nombre de mouvements optimal si on demande que l'ordre des disques à l'arrivée soit le même que au départ...

Question 8. on considère maintenant la variante de Hanoi avec un emplacement suplémentaire. Essayer de résoudre le problème pour 1, 2, 3, 4, 5 disques. Pouvez-vous trouver une relation de récurrence ?

Exercice 2 : une récurrence intéressante

Soit la récurrence suivante :

$$\begin{cases} u_0 = \alpha \\ u_1 = \beta \\ u_{n+2} = (1 + u_{n+1})/u_n \end{cases}$$

avec α et β deux nombres strictements positifs

Question 1. résolvez cette récurrence pour $\alpha = \beta = 1$.

Question 2. même question pour $\alpha = \beta = 2$.

Question 3. dans le cas général, montrez que $u_4 = (1+\alpha)/\beta$; déduisez en une formule générale pour u_n.

Exercice 3 : approximations asymptotiques

On rappelle les définitions du cours :
- $f(n) = O(g(n))$ si $\exists C\, \exists n_0$ t.q. $\forall n > n_0\, |f(n)| < C|g(n)|$
- $f(n) = \Omega(g(n))$ si $g(n) = O(f(n))$
- $f(n) = \Theta(g(n))$ si $f(n) = O(g(n))$ et $f(n) = \Omega(g(n))$

Question 1. donnez la définition de $f(n) = \Omega(g(n))$ avec des quantificateurs $\exists$ et $\forall$.

Idem pour $f(n) = \Theta(g(n))$.

Montrez que si $f(n) = O(g(n))$ et $g(n) = O(h(n))$ alors $f(n) = O(h(n))$.

Question 2. comme dans le cours, on note $f(n) \prec g(n)$ pour $\lim_{n\to\infty} f(n)/g(n) = 0$. On a la hiérarchie suivante

$$1 \prec \log(\log(n)) \prec n^{\varepsilon} \prec n \prec n^{c} \prec n^{\log(n)} \prec c^{n} \prec n^{n} \prec c^{(c^n)}$$

(où ε et c sont des constantes arbitraires avec $0 < \varepsilon < 1 < c$)

Où positionnez-vous les fonctions $\log(n)$, $\ln(n)$ et $n\log(n)$?

Question 3. la formule de Stirling est incroyable :

$$n! \quad = \quad \sqrt{2\pi n}\left(\frac{n}{e}\right)^n + O(1/n)$$

En utilisant cette formule, donnez une approximation de $\ln(1)+\ln(2)+\cdots+\ln(n) = \sum_{i=1}^{n} \ln(i)$ lorsque $n \to \infty$.

Question 4. 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 5. 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 4 : 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 5 : 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 ?