Université Mohamed Premier **Master MS2I - MRI**
Faculté des Sciences, Oujda **Année académique 2012 - 2013**

Complexité Algorithmique & Optimisation Combinatoire

Corrigé de l'épreuve écrite de complexité algorithmique

(Session de rattrapage)

##########################

Réponses

1. $\sum_{i=1}^{n} i^k \leq \sum_{i=1}^{n} n^k == n \times n^k = n^{k+1}$. Cela montre que $\sum_{i=1}^{n} i^k = O(n^{k+1})$.
2. – Comme $2^{n+1} = 2 \times 2^n \leq C \times 2^n$ pour toute constante $C \geq 2$ et pour tout entier $n \geq 0$, alors $2^{n+1} = O(2^n)$.
   – Si $2^{2n} = O(2^n)$, alors il existe une constante $C > 0$ et un entier $n_0 \geq 0$ tels que $2^{2n} = 2^n \times 2^n \leq C \times 2^n$, $\forall n \geq n_0$. Ce qui entraîne que $2^n \leq C$, $\forall n \geq n_0$. Cela est impossible, car la suite $(2^n)_{n\geq 0}$ n'est pas majorée, puisque $\lim_{n\to+\infty} 2^n = +\infty$.
3. $\Omega((n!)^2) \supseteq \Omega(10^n) \supseteq \Omega((1.5)^n) \supseteq \Omega(n^{100}) \supseteq \Omega(n^{99} + n^{98}) \supseteq \Omega(\sqrt{n}\log n) \supseteq \Omega((\log n)^3)$.
4. – $O(\log n)$: recherche dichotomique.
   – $O(n)$: recherche séquentielle.
   – $O(n \log n)$: tri par fusion.
5. Tri par sélection, tri par insertion, tri à bulles.
6. – 1er algorithme : $\Omega((\sqrt{2})^n)$.
   – 2eme algorithme : $\Theta(n)$.
   – 3eme algorithme : $\Theta(\log n)$.
7. Un invariant de boucle est une relation mathématique qui relie les variables d'une boucle et qui reste vraie à chaque passage de la boucle (itération).
8. – Complexité de l'algorithme A : il y a une instruction d'affectation simple, puis une boucle imbriquée dans une autre. Le corps de la boucle interne comprend 3 instructions élémentaires (une addition, une multiplication et une affectation). Donc $T(n) = 1 + \sum_{i=1}^{3}\sum_{j=1}^{4} 3 = 37 = \Theta(1)$.
   – Complexité de l'algorithme B : il y a deux instructions d'affectation simples, suivies d'une boucle simple. Le corps de la boucle comprend 4 instructions élémentaires (une addition, une multiplication et 2 affectations). La boucle s'exécute $(\lfloor \log_2 n \rfloor + 1)$ fois. Donc $T(n) = 2 + 4 \times (\lfloor \log_2 n \rfloor + 1) = 4 \times (\lfloor \log_2 n \rfloor) + 6 = \Theta(\log_2 n)$.

9. $T(n) = 3T(\frac{2}{3}n) + \Theta(1)$ pour $n \geq 2$. Ici, $a = 3$, $b = \frac{3}{2}$ et $d = 0$. Comme $\log_b a = 2.709 > d$, alors $T(n) = \Theta(n^{2.709})$.
10. C'est l'algorithme de tri à bulles. Sa complexité est $\Theta(n^2)$.
11. Complexité de l'algorithme de Karatsuba est :

$$T(n) = \begin{cases} 1 & si\ n = 1 \\ 3T(\frac{n}{2}) + O(n) & si\ n \geq 2 \end{cases}$$

D'après le Master théorème : $T(n) = O(n^{\log_2 3}) = O(n^{1.58})$.

12. Complexité de l'algorithme de Strassen :

$$T(n) = \begin{cases} 1 & si\ n = 1 \\ 7T(\frac{n}{2}) + 2n^2 & si\ n \geq 2 \end{cases}$$

D'après le Master théorème : $T(n) = O(n^{\log_2 7}) = O(n^{2.807})$.

13. Un algorithme "**diviser pour régner**" suit la stratégie suivante :
   - **Diviser** : le problème en sous-prolèmes dont la résolution est identique à celle du problème initial.
   - **Régner** : sur les sous-prolèmes en les résolvant de manière récursive.
   - **Combiner** : les solustions des sous-prolèmes pour obtenir la solution du problème initial.

   Complexité d'un algorithme "diviser pour régner" : soit a le nombre de sous-problèmes, n/b la taille d'un sous-problème et $f(n)$ le coût de combinaision. Alors : $T(n) = aT(n/b) + f(n)$. Dans le cas où $f(n) = \Theta(n^d)$, on utilise le master théorème pour calculer $T(n)$.
14. $T_A = \sum_{i=1}^{n} p_i X(i)$.
15.
```
Procédure dirBonjour(n : Entier);
Début
   Si(n = 0) Alors Afficher("Bonjour");
      Sinon  { dirBonjour(n - 1);  dirBonjour(n - 1); }
   FinSi
Fin
```
   Complexité en nombre d'affichage : $T(0) = 1$ et $T(n) = 2T(n-1) = 2^2T(n-2) = ... = 2^nT(n-n) = 2^nT(0) = 2^n$, $\forall n \geq 1$.
16. Le détail qui manque : $z = power_to_bin(n/2)$;
   L'équation de récurrence pour le temps d'exécution $T(n)$ de l'algorithme :

$$T(n) = \begin{cases} \Theta(1) & si\ n = 1 \\ T(\frac{n}{2}) + O(n^{1.58}) & si\ n = 2^k \geq 2 \end{cases}$$

   Master théorème : $a = 1$, $b = 2$ et $d = 1.58$ et $\log_b a = 0 < d$, donc $T(n) = O(n^{1.58)}$.