

## EA4 – Éléments d’algorithmique II

### Interrogation n° 1 – groupe INFO 3

Durée : 1h15

*Aucun document autorisé. Appareils électroniques rangés.*

*Le sujet est (trop) long, il en sera tenu compte dans la notation. Les exercices sont indépendants et ne sont absolument pas classés par ordre de difficulté.*

#### Exercice 1 :

Classer les 16 fonctions suivantes en fonction de leur ordre de grandeur dans les classes  $\Theta_1$  à  $\Theta_9$  : les fonctions appartiennent à la même classe  $\Theta_i$  si et seulement si elles sont du même ordre de grandeur et les classes  $\Theta_i$  sont rangées en ordre croissant :

$$\forall f, g, \quad f = \Theta(g) \iff \exists i, f, g \in \Theta_i$$

$$\forall i, \quad \forall f, g, \quad f \in \Theta_i \text{ et } g \in \Theta_{i+1} \implies f \in O(g)$$

Fonctions à traiter :  $5n^3 + n^{2015}$ ,  $2n^2 + \log_2 n$ ,  $\frac{1}{2}n^2 \cdot \log_2 n$ ,  $4^n$ ,  $42$ ,  $2^{2n+1}$ ,  $\log_3 n$ ,  
 $n \cdot \ln(n^n)$ ,  $10^9$ ,  $2^{n+4}$ ,  $\log_2(n^2)$ ,  $\log_2(2n+1)$ ,  $n^4 \cdot 3^n$ ,  $n!$ ,  $\log_{10}(n^2!)$ ,  $6 \times 9$

$\Theta_1$	$\Theta_2$	$\Theta_3$	$\Theta_4$	$\Theta_5$	$\Theta_6$	$\Theta_7$	$\Theta_8$	$\Theta_9$

#### Exercice 2 :

Dans la suite,  $f$  et  $g$  désignent des fonctions à valeurs **entières** strictement positives. Les affirmations suivantes sont-elles vraies ou fausses ? Justifier ou donner un contre-exemple.

	Vrai	Faux
$f = O(g) \implies f = \Theta(g)$	<input type="checkbox"/>	<input type="checkbox"/>
$f = o(g) \implies f + g = O(g)$	<input type="checkbox"/>	<input type="checkbox"/>
$fg = O(n^2) \implies f = O(n^2) \text{ ou } g = O(n^2)$	<input type="checkbox"/>	<input type="checkbox"/>
$fg = O(n^2) \implies f = O(n^2) \text{ et } g = O(n^2)$	<input type="checkbox"/>	<input type="checkbox"/>

**Exercice 3 :**

On considère l’algorithme suivant :

```
def foo(n) :  
    if n < 3 : return 1  
    else : return foo(n - 1) + foo(n - 3)
```

Soit  $A(n)$  le nombre d’additions effectuées lors de l’exécution de `foo(n)`.

Donner une définition de  $A(n)$  par récurrence. En déduire que  $A(n)$  est croissante.

---

---

---

Montrer que  $A(n) = \Omega(2^{n/3})$ .

---

---

---

---

---

---

---

---

---

---

Proposer un algorithme plus efficace pour calculer les mêmes valeurs et justifier sa complexité.

---

---

---

---

---

---

---

---

---

---

**Exercice 4 :**

Soit **T** le tableau suivant : 

3	2	8	5	1	7	6	4
---	---	---	---	---	---	---	---

Décrire le déroulement de l’algorithme de tri par insertion appliqué à **T**. Combien de comparaisons d’éléments sont effectuées (exactement) ?

**Exercice 5 :**

On considère l’algorithme suivant :

```
def bar(T, m) :  
    d = 0  
    for i in range(m) :  
        if T[i] > T[i + 1] :  
            T[i], T[i + 1] = T[i + 1], T[i]  
            d = i  
    if d > 0 : bar(T, d)
```

On suppose  $m < \text{len}(T)$ . Déterminer le nombre d’échanges effectués par `bar(T, m)` en fonction de  $m$  dans le pire des cas.

---

---

---

---

---

---

---

Déterminer le nombre de comparaisons effectuées par `bar(T, m)` dans le meilleur des cas. À quoi correspond ce cas ?

---

---

---

---

---

---

---

Que fait l’algorithme `bar(T, m)` ? (*ne pas oublier le paramètre  $m$* )

---

---

---

---

---

---

---