# Techniques mathématiques pour l'informatique
## TD – Séance n° 2

---

**Exercice 1 – Récurrences - fonction Mystère**

Donner la complexité de l'algorithme suivant :

```
MYSTÈRE(n)
calcule Mystre(n)
if n = 1 then
   return 3
else
   return Mystere(n/2) + 2
end if
```

**Exercice 2 – Récurrences - fonction Mystère2**

Donner la complexité de l'algorithme suivant :

```
MYSTÈRE2(n)
calcule Mystre2(n)
if n = 1 then
   return 3
else
   s ← 0
   for i ← 1 à n do
      s ← i
   end for
   return 3 * Mystere2(n/2) + s * Mystere2(n/2)
end if
```

**Exercice 3 – Récurrences - multiplication de deux polynômes**

Soit $P[X] = p_0 + p_1X + ... + p_nX^n$ et $Q[X] = q_0 + q_1X + ... + q_nX^n$ deux polynômes de degré $n$. Le but est de trouver des algorithmes permettant de calculer le produit de $P[X]$ et $Q[X]$.

1) Donner la complexité d'un algorithme naïf pour multiplier deux polynômes de degré $n$ (on exprimera la complexité en nombre d'opérations arithmétiques nécessaires).

2) Montrer comment multiplier deux polynômes linéaires $ax + b$ et $cx + d$ à l'aide de trois multiplications seulement (aide : l'une des multiplications est $(a + b)(c + d)$).

3) Le but est maintenant de développer un algorithme de type "diviser pour régner" ayant une complexité meilleure que l'algorithme naïf.

a) Montrer qu'il existe un polynôme $A[X]$ de degré $\lfloor \frac{n}{2} \rfloor$ et un polynôme $B[X]$ de degré

$\lfloor \frac{n}{2} \rfloor$ tels que $P[X] = A[X]X^{1+\lfloor \frac{n}{2} \rfloor} + B[X]$ (on a donc aussi de la même manière $C[X]$ et $D[X]$ tels que $Q[X] = C[X]X^{1+\lfloor \frac{n}{2} \rfloor} + D[X]$ )

b) En utilisant 2) et 3)a), donner une formule de la multiplication de deux polynômes de degré $n$ à l'aide de seulement trois multiplications de polynômes de degré $\lfloor \frac{n}{2} \rfloor$ (et un nombre linéaire d'additions).

c) Donner l'équation de récurrence définissant la complexité de cet algorithme, et la résoudre grâce au théorème général.