

## Exercices INF7440 : série #1

1. À l'aide des définitions, montrez les propriétés suivantes :

a.  $an^2 + bn + c \in O(n^2)$  (en supposant  $a, b, c \geq 0$ ).

b.  $(1 + 2 + \dots + n) \in O(n^2)$

c.  $(1 + \frac{1}{2} + \frac{1}{4} + \dots + \frac{1}{2^n}) \in \Omega(1)$

Rappel de propriétés utiles (pour  $0 < a \neq 1$ ) :

$$\sum_{i=0}^n i = \frac{n(n+1)}{2}$$
$$\sum_{i=0}^n a^i = \frac{1 - a^{n+1}}{1 - a}$$

2. Trouvez un estimé  $O$  le plus précis possible (en termes des classes de complexité) pour chacune des fonctions suivantes, définies sur les naturels ( $n \geq 0$ ) :

a.  $\lg(n \times 2^n)$

b.  $n \times \lg(2^{n^2})$

c.  $\lg(n + 4)^2$

d.  $2^{1000} + 2n^2$

e.  $2^{(3-n)}$

Note : Un logarithme  $\lg$  est en base 2. Autrement,  $\log_b$  est utilisé pour spécifier la base du logarithme.

3. Soit l'algorithme suivant, qui trouve le *mode* d'une suite de nombres (c'est-à-dire, le nombre qui apparaît le plus souvent dans la suite) :

```
PROCEDURE trouverMode( a1, ..., an: Entier ) mode: Entier;
DEBUT
  // Pre-condition: n > 0
  nbOccurrencesMode <- 0
  POUR i <- 1 A n FAIRE
    nbOccurrences <- 0
    POUR j <- 1 A n FAIRE
      SI ai == aj ALORS
        nbOccurrences <- nbOccurrences + 1
    FIN
  FIN
  SI nbOccurrences > nbOccurrencesMode ALORS
    nbOccurrencesMode <- nbOccurrences
    mode <- ai
  FIN
FIN
```

Quelle est la complexité asymptotique de cet algorithme? S'agit-il d'une complexité dans le pire cas, dans tous les cas?

4. Soit les algorithmes suivants :

```
FONCTION sc( n: Natural ): Natural
DEBUT
  somme <- 0
  TANTQUE n > 0 FAIRE
    somme <- somme + (n MOD 10)
    n <- n / 10 // Division entiere
  FIN
  RETOURNER somme
FIN
```

```
FONCTION sc_r( n: Natural ): Natural
DEBUT
  SI n == 0 ALORS
    RETOURNER 0
  SINON
    RETOURNER (n MOD 10) + sc_r(n / 10) // Division entiere
  FIN
```

- Quelle est la complexité asymptotique de chacun de ces algorithmes — pour simplifier, vous pouvez supposer que  $n = 10^k$  pour  $k \geq 0$ ?
- Si on code ces deux algorithmes dans un même langage, compilés et exécutés dans un même environnement, lequel des algorithmes serait très certainement plus efficace que l'autre? Expliquez.