TD1 : décompte d'opérations et notations asymptotiques

Exercice 1 :

Montrer que :

1. $n^2 + 2n + 3 = \mathcal{O}(n^2)$,
2. $n^2 + 2n + 3 = \Omega(n)$,
3. n^3 n'appartient pas à $\Theta(n^2)$

Exercice 2 :

Regrouper en classes d'équivalence pour la relation Θ les fonction suivantes :

1. $f_1(n) = n$
2. $f_2(n) = 2^n$
3. $f_3(n) = n \log n$
4. $f_4(n) = n - n^3 + 7n^5$
5. $f_5(n) = n^2 + \log n$
6. $f_6(n) = n^2$
7. $f_7(n) = \log n$
8. $f_8(n) = n^3$
9. $f_9(n) = \sqrt{n} + \log n$
10. $f_{10}(n) = (\log n)^2$
11. $f_{11}(n) = n!$
12. $f_{12}(n) = \ln n$

Exercice 3 :

Donner et montrer la borne asymptotique des fonctions :

- $f_1(n) = n^3 - n^2 + 2^{21}$
- $f_2(n) = \sum_{i=1}^{n} i^2$

Exercice 4 :

Montrer que pour tout couple d'entiers (n, k) on a :

$$\sum_{i=1}^{n} i^k = \Theta(n^{k+1})$$

Exercice 5 :

```
function stupide1 (n : CARDINAL) : CARDINAL;
var i, j, k, r : CARDINAL;
begin
  r := 0;
  for i := 1 to n-1 do
    for j := i + 1 to n do
      for k := 1 to j do
        r := r + i + j + k;
  stupide1 := r;
end {stupide1};
```

Q 5.1 Calculer le nombre d'additions effectuées en fonction de n.

Q 5.2 En déduire le comportement asymptotique correspondant.

Exercice 6 :

```
function stupide2 (t : TABLEAU) : CARDINAL;
var i, j, r : CARDINAL;
begin
  i := low(t);
  j := high(t);
  r := 0;
  while ((i+j) div 2) <> i do begin
    j := (i+j) div 2;
    inc(r);
  end {while};
  stupide2 := r;
end {stupide2};
```

Q 6.1 Calculer le nombre d'additions effectuées en fonction de n.

Q 6.2 En déduire le comportement asymptotique correspondant.

Exercice 7 :

Nous disposons de plusieurs algorithmes calculant la même chose. Le nombre d'opérations est compté sur des opérations dont chacune prend un temps unitaire de 1ms. On souhaite calculer la *taille maximale* de la donnée que l'on peut traiter en fonction du temps où on laisse s'exécuter l'algorithme. Remplir le tableau suivant :

Nombre d'opérations de l'algorithme	Temps de calcul						
	1ms	10ms	100ms	1s	1mn	1h	1j
$\log n$							
n							
$n \log n$							
n^2							
n^3							
n^4							
2^n							