TD R1.01.P2 – Semaine 47

Objectifs du TD/TP

- Etudes et comparaison d'algorithmes
- Efficacité théorique d'un algorithme
- Complexité des algorithmes et ordres de grandeur

Ce TD dure 1 X 1h30

Exercice 1. Ordres de grandeur d'algorithmes

Pour chacun des 4 algorithmes ci-dessous :

- Dire ce qu'affiche le code.
- En fonction de $n \in \mathbb{N}$ quel est le nombre **exact** d'opérations que l'algorithme effectue ?
- En déduire l'ordre de grandeur θ en fonction de n de l'algorithme.

Algo1

```
int a = 0;
for ( int i=0; i<n+1; i++ ) {
   a = i;
   if ( a >= n ){
      a = a - i;
   }
}

System.out.println ( "Algo1 val de a = " + a );
```

Algo2

```
int a = 0;
for ( int i=1; i<=n; i++ ) {
   for ( int j=1; j<n+1; j++ ) {
      a = a + i;
   }
}

System.out.println ( "Algo2 val de a = " + a );
```

Exercice 2. Efficacité non triviale d'un algorithme

Soit l'algorithme :

```
int algoMystere ( int n ) {
   int valeur, nb1, reste;
   int ret;

   valeur = n;
   nb1 = 0;
   while ( valeur != 0 ) {

      reste = valeur % 10;           // valeur modulo 10
      valeur = (int) (valeur / 10);  // division entière

      if ( reste = = 1 ) {
         nb1++;
      }
   }

   if ( (nb1 % 2) = = 0 ) {
      ret = n * 10;
   }

   else {
      ret = (n * 10) + 1;
   }

   return ret;
}
```

Remarque importante : on supposera que l'entier passé en paramètre « n » est positif et uniquement composé de 0 et de 1 et qu'il commence nécessairement par le chiffre 1.

a. En supposant, dans un premier temps, que le corps de la boucle est exécutée « k » fois, calculez l'expression **exacte** de *f(k)*, le coût de l'algorithme en nombre d'opérations élémentaires effectuées en fonction de « k ». On se placera uniquement dans le pire des cas et on ne comptabilisera pas les opérations de passage du paramètre et du retour. Donnez la valeur de « n » qui fait atteindre ce pire des cas avec exactement 3 itérations (donc quand $k = 3$).

b. Dans un second temps, exprimez « k » en fonction de « n » qui vous permettra de calculer cette fois *f(n)*, le coût **exact** dans le pire des cas de l'algorithme en nombre d'opérations élémentaires effectuées en fonction de « n » (l'entier passé en paramètre à la méthode).

   Indication de départ, le paramètre « n » en entrée peut toujours s'écrire sous la forme $n = m\ 10^p$ où :

   - *m* est un réel qui commence par le chiffre 1 (i.e. 1,…)
   - *p* est un exposant (entier positif)

c. Pour des valeurs de « n » élevées, en déduire l'ordre de grandeur θ dans le pire des cas en fonction de « n » de l'algorithme.