Série 0x27

La récursivité

INFO2-TIN 3 mars 2026, 7efbe5

Exercice 1 : Fibonacci

On considère la suite de Fibonacci, définie par la relation suivante :

$$\begin{cases} f_0 &= 0 \\ f_1 &= 1 \\ f_n &= f_{n-1} + f_{n-2} \end{cases}$$

(a) Écrire une fonction récursive qui calcule le terme de rang n de cette suite.
(b) Représentez par un arbre les appels de fonction imbriqués lors de l'évaluation pour l'appel de `fib(7)`.
(c) Quelle est la complexité en temps (*big-O*) de cette fonction ?
(d) Écrire une fonction itérative qui calcule le terme de rang n.
(e) Quelle est la complexité en temps de cette fonction itérative ?
(f) Écrire une fonction de mémoization en programmation dynamique qui encapsule votre fonction récursive afin de réduire la complexité.
(g) Quelle est la complexité de cette nouvelle fonction ?

Exercice 2 : Simulation numérique

Pour réaliser certaines fonctions de calcul dans une application de simulation numérique, on doit disposer d'une fonction calculant le polynôme suivant :

$$p(x) = \frac{0 \cdot x^0}{0!} + \frac{1 \cdot x^1}{1!} + \frac{2 \cdot x^2}{2!} + \frac{3 \cdot x^3}{3!} + \frac{4 \cdot x^4}{4!} + \frac{5 \cdot x^5}{5!}$$

Le calcul de ce polynôme a été programmé de la façon suivante dans cette application :

```
long factoriel(long n)
{
  return n == 0 ? 1 : n * factoriel(n-1);
}

double power(double x, int n)
{
  double result = 1.0;
  for (int i = 0; i < n; i++)
    result *= x;
  return result;
}

double compute_polynom(double x)
{
  double result = 0;
  for (int i = 0; i <= 5; i++)
    result += i*power(x, i)/factoriel(i);
  return result;
}
```

Cette fonction est appelée des millions de fois pendant le calcul de la simulation numérique, et les temps de calcul obtenus sont beaucoup trop élevés.

(a) Expliquez pourquoi la programmation de cette fonction n'est pas optimale au niveau des performances.
(b) Proposez des optimisations du code permettant de diminuer significativement les temps de calcul de cette fonction.