

## Trois algorithmes de calcul des nombres de Fibonacci

Dans cette série d'exercices, nous nous intéressons de la complexité dite *arithmétique*. Ce modèle prend en compte uniquement le nombre des opérations arithmétiques, sans se soucier de la taille des nombres en question.

L'objectif des trois algorithmes présentés ci-dessous est le calcul du  $n$ ème nombre de Fibonacci  $f_n$ .

**Exercice 1 (Algorithme récursif)** Soit l'algorithme suivant :

```
si  $n = 0$  ou  $n = 1$  alors
     $fib(n) = 1$ 
sinon
     $fib(n) = fib(n - 1) + fib(n - 2)$ 
fin-si
```

Estimer la complexité de cet algorithme.

**Exercice 2 (Algorithme itératif)** Soit l'algorithme suivant :

```
si  $n = 0$  ou  $n = 1$  alors
     $fib(n) = 1$ 
sinon
     $a = 1$ 
     $b = 1$ 
    pour  $i$  de 2 à  $n$  faire
         $c = a + b$ 
         $a = b$ 
         $b = c$ 
    fin-pour
     $fib(n) = c$ 
fin-si
```

Estimer la complexité de cet algorithme.

**Exercice 3 (Algorithme matriciel)** Soit  $A$  la matrice suivante :

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

1. Montrer que

$$A \cdot \begin{pmatrix} f_{n-1} \\ f_n \end{pmatrix} = \begin{pmatrix} f_n \\ f_{n+1} \end{pmatrix}.$$

2. Conclure que

$$\begin{pmatrix} f_n \\ f_{n+1} \end{pmatrix} = A^n \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

3. Proposer un algorithme *très* efficace de calcul de  $f_n$  basé sur l'observation précédente. (Une présentation formelle de cet algorithme n'est pas demandée.) Estimer la complexité de cet algorithme.

**Exercice 4 (Généralisation)** Adapter la même méthode à la suite récurrente suivante :

$$a_0 = 1$$

$$a_1 = 2$$

$$a_2 = 1$$

$$a_n = 2a_{n-1} + 5a_{n-2} - a_{n-3} \quad \text{pour } n \geq 3.$$