

# Les nombres de Fibonacci

## Une image



Figure 1: Fibonacci

Pour d'autres informations sur Fibonacci, cliquez [ici](#)

## Des Lapins

Le problème de Fibonacci est à l'origine de la suite dont le  $n$ -ième terme correspond au nombre de paires de lapins au  $n$ -ième mois. Dans cette population (idéale), on suppose que :

- au (début du) premier mois, il y a juste une paire de lapereaux ; les lapereaux ne procréent qu'à partir du (début du) troisième mois ;
- chaque (début de) mois, toute paire susceptible de procréer engendre effectivement une nouvelle paire de lapereaux ;
- les lapins ne meurent jamais (donc la suite de Fibonacci est croissante).

## Des mathématiques

Notons  $f_n$  le nombre de couples de lapins au début du mois  $n$ . Jusqu'à la fin du deuxième mois, la population se limite à un couple (ce qu'on note :  $f_1 = f_2 = 1$ ).

N'engendrent au mois  $n+2$  que les couples pubères, c'est-à-dire ceux qui existent deux mois auparavant. On a donc, pour tout entier  $n$  strictement positif :

$$f_{n+2} = f_{n+1} + f_n$$

On choisit alors de poser  $f_0 = 0$ , de manière que cette équation soit encore vérifiée pour  $n = 0$ .

On obtient ainsi la forme récurrente de la suite de Fibonacci : chaque terme de cette suite est la somme des deux termes précédents ; pour obtenir chacun de ces deux termes, il faut faire la somme de leurs termes précédents.

## Des Nombres

Voici les nombres de fibonacci pour  $0 \leq n \leq 10$  :

$n$	$f_n$
0	0
1	1
2	1
3	2
4	3
5	5
6	8
7	13
8	21
9	34
10	55

## Fibonacci et la récursivité

On donne deux implémentations récursives du calcul de la suite :

### première version

```
def fibo1(n):  
    """  
    :param n: (int) un entier  
    :return: (int) le terme  $f_n$  de la suite de Fibonacci  
    :CU: Aucune  
    :Exemples:  
  
    >>> fibo1(0)  
    0  
    >>> fibo1(7)  
    13  
    """  
    if n == 0 or n == 1:  
        return n
```

```

else:
    return fibo1(n-1) + fibo1(n-2)

```

## Deuxième version

```

def fibo2(n, fm1 = 1, fm2 = 0):
    """
    :param n: (int) un entier
    :return: (int) le terme f_n de la suite de Fibonacci
    :CU: Aucune
    :Exemples:

    >>> fibo2(0)
    0
    >>> fibo2(7)
    13
    """
    if n == 0:
        return fm2
    else:
        return fibo2(n-1, fm2, fm1+fm2)

```

## Questions

1. Lequel de ces deux algorithmes est récursif terminal ?
2. Démontrer que le nombre d'additions de termes de la suite nécessaires pour calculer  $f_n$  à l'aide du premier algorithme est pour  $n \geq 1$ :

$$u_n = f_{n+1} - 1$$

3. Démontrer que le nombre d'additions de termes de la suite nécessaires pour calculer  $f_n$  à l'aide du second algorithme est pour  $n \geq 0$ :

$$u_n = n$$