

# Reporte de algoritmo Fibonacci

## Matemáticas computacionales

*Ismael Medina Robledo*

*1744617*

### Resumen

La serie de Fibonacci, descubierta por Leonardo de Pisa, tiene numerosas aplicaciones en ciencias de la computación, matemática y teoría de juegos. También aparece en configuraciones biológicas, como por ejemplo en las ramas de los árboles, en la disposición de las hojas en el tallo, en las flores de alcachofas y girasoles, en las inflorescencias del brécol romanesco y en la configuración de las piñas de las coníferas. De igual manera, se encuentra en la estructura espiral del caparazón de algunos moluscos, como el nautilus.

- **Algoritmo de números primos**

Elaboramos un algoritmo que identificaba si un número era primo o no. El algoritmo inicialmente tiene una complejidad de  $n$  operaciones o sea el mismo número que se teclea, aunque puede llegar a ser más eficiente.

```
1  import math
2
3
4
5  def primo(n):
6
7      cnt=0
8
9      for i in range(2,math.ceil(math.sqrt(n))):
10
11          cnt=cnt+1
12
13          if((n%i)==0):
14
15              break
16
17      return cnt
```

- **Fibonacci 1**

Este algoritmo tarda más en encontrar el número de la serie de Fibonacci en la posición deseada.

---

```
1  >>> cnt=0
2  >>> def fibonacci(n):
3      global cnt
4      cnt+=1
5      if n==0 or n==1:
6          return(1)
7      else:
8          return fibonacci(n-2)+fibonacci(n-1)
9
```

- **Fibonacci 2**

Con este otro algoritmo se agiliza la búsqueda del número de la posición que se teclea.

```
11  >>> def fibo(n):
12      global cnt
13      if n==0 or n==1:
14          return(1)
15      r, r1, r2 = 0, 1, 2
16      for i in range (2,n):
17          cnt+=1
18          r=r1+r2
19          r2=r1
20          r1=r
21      return (r)
```

- **Fibonacci 3**

Por ultimo tenemos este algoritmo cuya ventaja es guardar los números de la posición que se haya tecleado para agilizar la operación de una nueva posición a buscar.

```
1  >>> memo={}
2  >>> cnt=0
3  >>> def fibonacci(n):
4      global memo,cnt
5      cnt+=1
6      if n==0 or n==1:
7          return(1)
8      if n in memo:
9          return memo[n]
10     else:
11         val=fibonacci(n-2)+fibonacci(n-1)
12         memo[n]=val
13         return val
14
15
16 >>> fibonacci(7)
17 21
18 >>> memo
19 {2: 2, 3: 3, 4: 5, 5: 8, 6: 13, 7: 21}
```

- **Conclusiones**

Los diferentes tipos de algoritmos que hay para calcular los números de la serie de Fibonacci mediante el tecleo de la posición son muy variados. Obviamente los más intuitivos son los más fáciles de realizar pero realizan un mayor número de operaciones.