

Serie de Fibonacci

1.

Para generar la serie de numero hasta la posición n de Fibonacci se utilizan 3 parámetros, n, el numero anterior al actual y el número actual, n será el que nos ayudara a generar la recursividad, el valor de n será el enésimo numero de la serie de Fibonacci hasta el que se va a mostrar la serie, posteriormente el numero anterior al actual será el numero que inicia en 0 y en numero actual inicia en 1, en número anterior al actual se irá actualizando como el numero actual posteriormente de que se realice la suma del numero anterior al actual mas el numero actual para ir generando la sucesión, teniendo esto se ira mostrando la sucesión mientras se imprime el numero actual una y otra vez hasta llegar a la enésima posición de la serie de Fibonacci.

Ejemplo:

n= 4

ant=0

act=1

si n es mayor a 0 entonces se imprime el numero actual y retorna serie_fibonacci_n(n-1, act, ant+act) y así se realiza la sucesión hasta que n es igual a 0 y termina la recursividad.

Si la condición de $n > 0$ no se cumple solamente retorna el numero actual que es 1.

Dado el ejemplo propuesto sería así:

$$N=4$$

$$\text{Act}=0+1 = 1$$

$$\text{Ant}=1$$

$$N=3$$

$$\text{Act}=1+1=2$$

$$\text{Ant}=2$$

$$N=2$$

$$\text{Act}=2+1=3$$

$$\text{Ant}=3$$

$$N=1$$

$$\text{Act}=3+2=5$$

$$\text{Ant}=5$$

$$N=0$$

Fin.

Así se concluye la recursividad y se obtiene el resultado requerido.

2.

¿Cuál es el caso más pequeño?

$$F_0=0$$

¿Cuál es el caso general?

$$F_n = (F_{n-1}, \text{act}, \text{ant}+\text{act})$$

3.

```
1 package Estructura_de_datos;
2 public class serie_fibonacci_n {
3     public serie_fibonacci_n() {}
4     public int serie_fibonacci_n(int n, int ant, int act) {
5         if (n > 0) {
6             System.out.print(act + " ");
7             return serie_fibonacci_n(n-1, act, ant+act);
8         } else {
9             return act;
10        }
11    }
12 }
13
```

4.

run:

Ingresa el valor de n: 7

Serie de Fibonacci: 1 1 2 3 5 8 13 21 BUILD SUCCESSFUL (total time: 2 seconds)

run:

Ingresa el valor de n: 8

Serie de Fibonacci: 1 1 2 3 5 8 13 21 34 BUILD SUCCESSFUL (total time: 3 seconds)