



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO



**TECNOLÓGICO NACIONAL DE MÉXICO INSTITUTO
TECNOLÓGICO DE TIJUANA**

SUBDIRECCIÓN ACADÉMICA

DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

SEMESTRE FEBRERO-JUNIO 2022

CARRERA

Ingeniería en informática

MATERIA

Datos masivos

TÍTULO

Práctica#2

Integrantes:

Munguía silva Edgar Geovanny #17212344

Alicia Guadalupe Pérez López#18210514

NOMBRE DEL MAESTRO

Jose Christian Romero Hernandez

Tijuana Baja California 09 de marzo del 2022

Having the pseudocode of the Fibonacci sequence in the provided link, implement Algorithm 1, Algorithm 2, Algorithm 3, Algorithm 4, Algorithm 5 using Scala.

Algorithm#1:

Algoritmo 1 Versión recursiva descendente (Complejidad $O(\varphi^n)$)

```
función fib( $n$ )  
    si  $n < 2$  entonces  
        devuelve  $n$   
    en otro caso  
        devuelve fib( $n - 1$ ) + fib( $n - 2$ )
```

Input:

```
/**  
 *Algorithm#1  
 */  
  
def alghth(n:Int):Int={  
  if(n<2){  
    return n  
  }  
  else{  
    return alghth(n-1)+alghth(n-2)  
  }  
}
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  
  
    | return n  
    | }  
    | else{  
    | return alghth(n-1)+alghth(n-2)  
    | }  
    | }  
alghth: (n: Int)Int  
  
scala> alghth(10)  
res0: Int = 55  
  
scala> |
```

Algorithm#2:

Algoritmo 2 Versión con fórmula explícita (6) (Complejidad $O(\log_2(n))$)

```
función fib( $n$ )  
    si  $n < 2$  entonces  
        devuelve  $n$   
    en otro caso  
         $\varphi \leftarrow ((1 + \sqrt{5}) \div 2)$   
         $j \leftarrow ((\varphi^n - (1 - \varphi)^n) \div (\sqrt{5}))$   
    devuelve  $j$ 
```

Input:

```
/**  
 *Algorithm#2  
 */  
  
def algh2(n:Int):Double={  
  if(n<2){  
    return n  
  }  
  else{ var p=((1+math.sqrt(5))/2)  
        var j=((math.pow(p,n)-(math.pow((1-p),n)))/(math.sqrt(5)))  
        return j  
  }  
}
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  
  
|    return n}  
|    else{ var p=((1+math.sqrt(5))/2)  
|          var j=((math.pow(p,n)-(math.pow((1-p),n)))/(math.sqrt(5)))  
|          return j  
|    }  
|  }  
algh2: (n: Int)Double  
  
scala> algh2(10)  
res1: Double = 55.000000000000014  
  
scala> |
```

Algorithm#3:

Algoritmo 3 Versión iterativa
(Complejidad $O(n)$)

función fib(n)

$a \leftarrow 0$

$b \leftarrow 1$

c

para k desde 0 hasta n **hacer**

$c \leftarrow b + a$

$a \leftarrow b$

$b \leftarrow c$

devuelve a

Input:

```
/**
 *Algorithm#3
 */
def alghm3(n:Int):Int={
var a=0
var b=1
var c=0
var i=0
if(i<n){
c=b+a
a=b
b=c
i=i+1}
return a
}
```

Output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

    alghnm4(10)
    ^

scala> alghnm3(10)
<console>:24: error: not found: value alghnm3
    alghnm3(10)
    ^

scala> alghnm3(10)
res28: Int = 55

scala> 
```

Algorithm#4:

Algoritmo 4 Versión iterativa
2 variables (Complejidad $O(n)$)

función fib(n)

$a \leftarrow 0$

$b \leftarrow 1$

para k desde 0 hasta n **hacer**

$b \leftarrow b + a$

$a \leftarrow b - a$

devuelve b

Input:

```
/**
 *Algorithm#4
 */
def algthm4(n:Int):Int={
var a=0
var b=1
var i=0
if(i<n){
b=b+a
a=b-a
i=i+1
}
return b}
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

| a=b-a
| i=i+1
| }
| return b}
algthm4: (n: Int)Int

scala>

scala> algthm4(10)
res29: Int = 1

scala> 
```