Algoritmos sobre Números

Prof. Martín Vigil

Sequência de Fibonacci

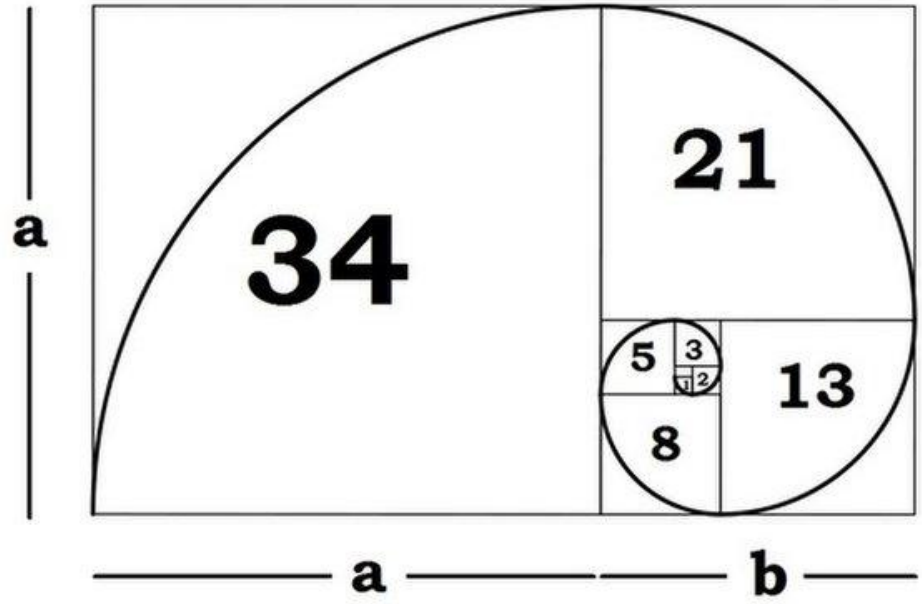
1, 1, 2, 3, 5, 8, 13, 21, 34...

Sequência de Fibonacci

Fib(n) = n-ésimo elemento da série

n	**Fib(n)**
1	1
2	1
3	2
4	3
5	5
6	8
...	
k	Fib(k-1) + Fib(k-2)

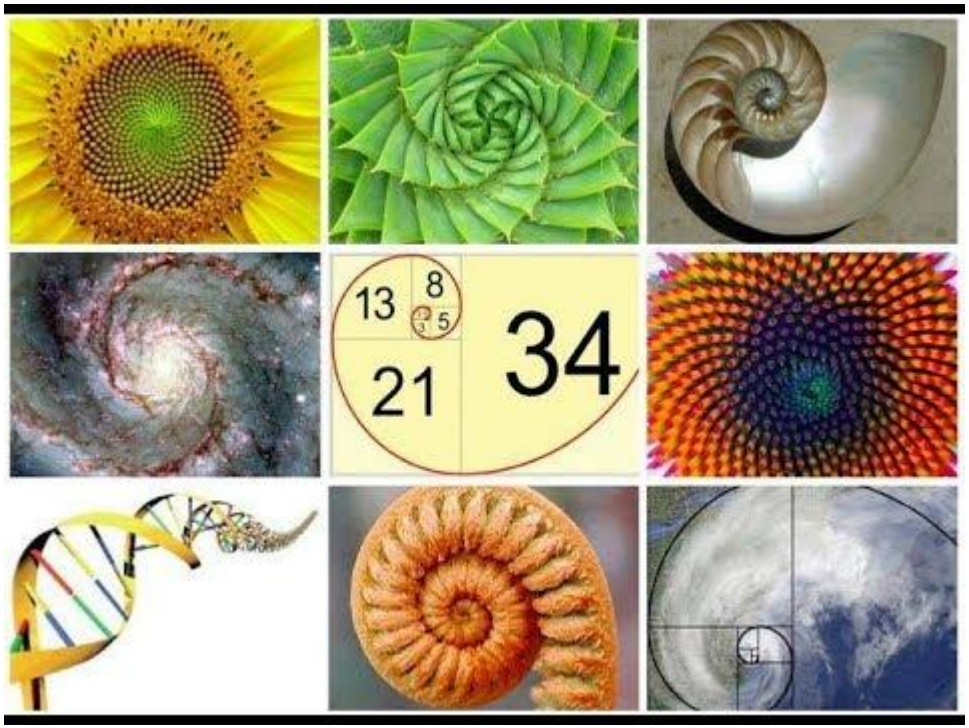
Sequência de Fibonacci: razão de ouro (φ)

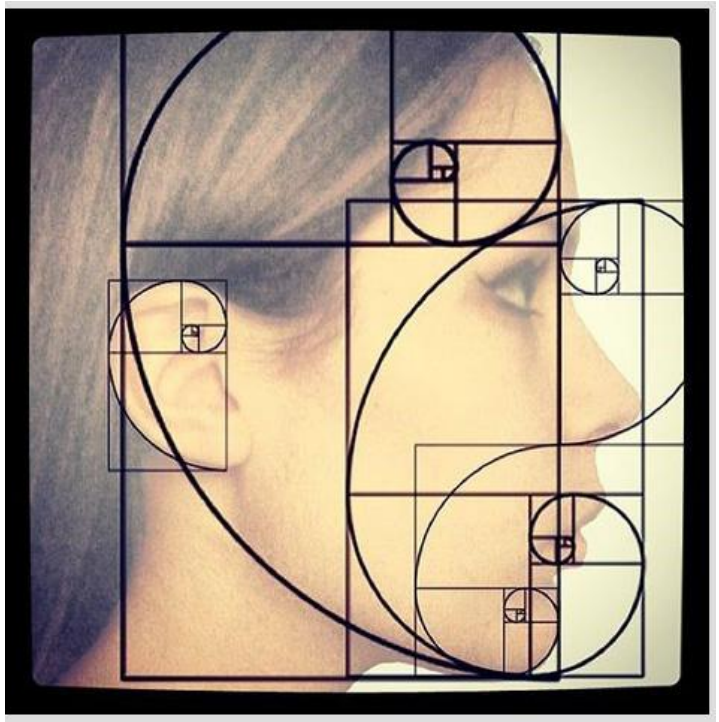


F(n)	F(n-1)	F(n)/F(n-1)
1	1	1
2	1	2
3	2	1.5
5	3	1.666666667
8	5	1.6
13	8	1.625
21	13	1.615384615
34	21	1.619047619
55	34	1.617647059
89	55	1.618181818

a/b = (a+b)/a = 1.6180339887498948420 …

Sequência de Fibonacci: razão de ouro (**φ**)





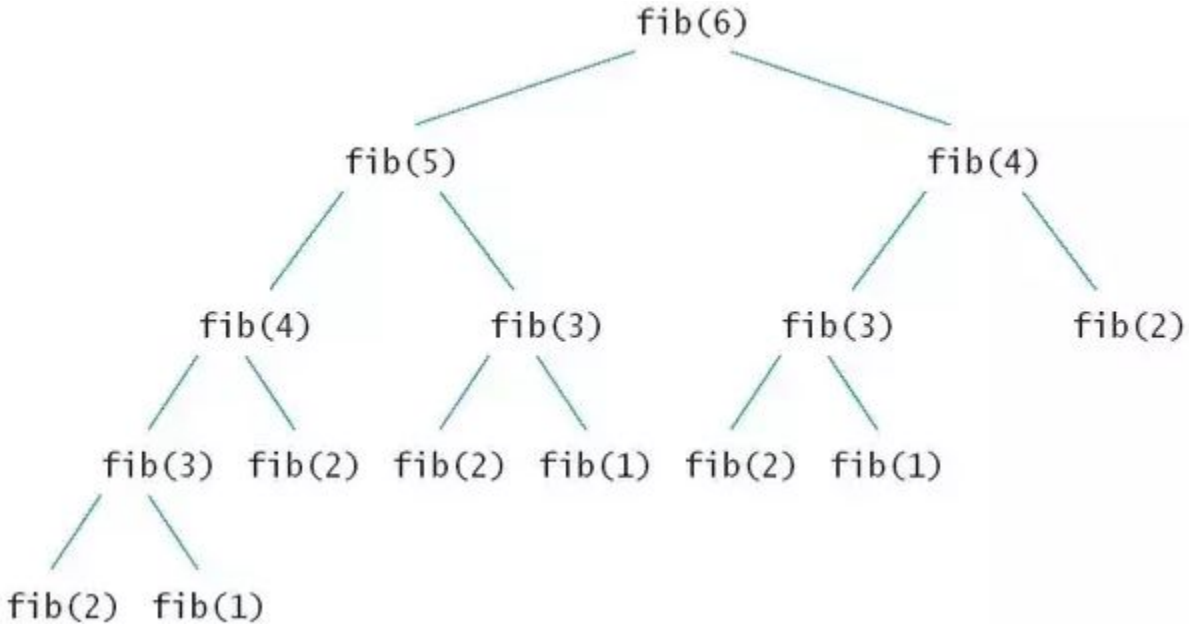
Algoritmo Recursivo para Fib(n)

$$Fib(n) = Fib(n-1) + Fib(n-2)$$

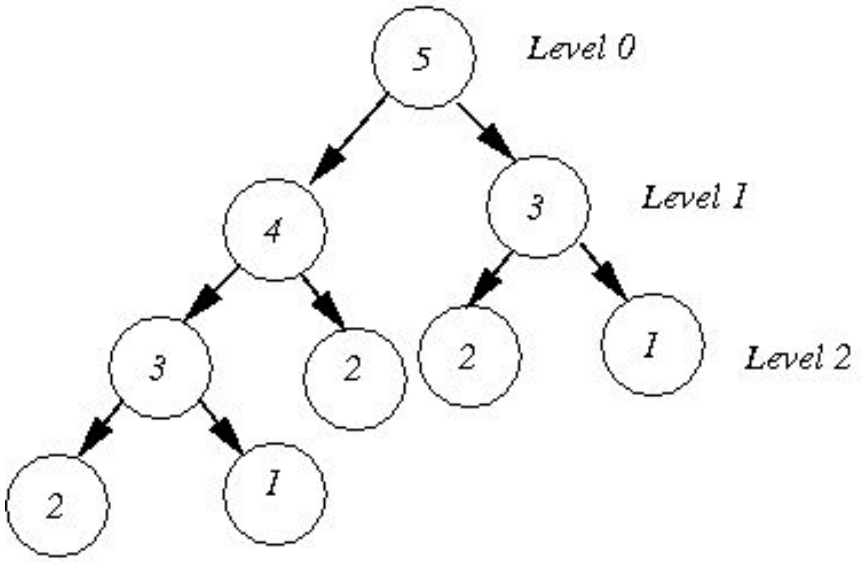
```
1.  função Fib(n)
2.      se n == 1 OU n == 2
3.          retorne 1
4.      senão
5.          retorne Fib(n-1) + Fib(n-2)
```

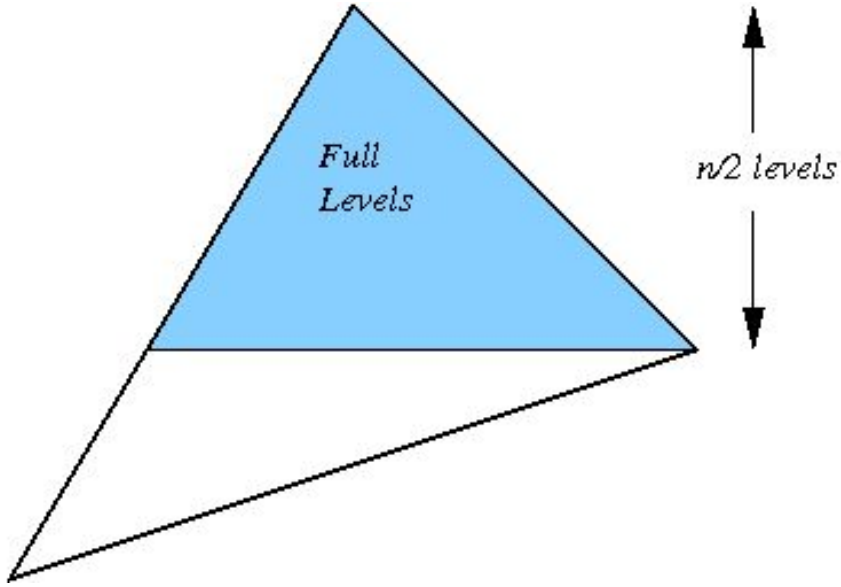
Executando Fib(6)

```
1.  função Fib(n)
2.      se n == 1 OU n == 2
3.          retorne 1
4.      senão
5.          retorne Fib(n-1) + Fib(n-2)
```



Executando Fib(5)





Análise do Algoritmo Recursivo para Fib(n)

$$Fib(n) = Fib(n-1) + Fib(n-2)$$

$$T(n) = T(n-1) + T(n-2) + 1$$

$$T(n) \in O(g(n))$$

Árvore Recursiva para Fib(n)

$T(n) = T(n-1) + T(n-2) + 1$

T(n)

Árvore Recursiva para Fib(n)

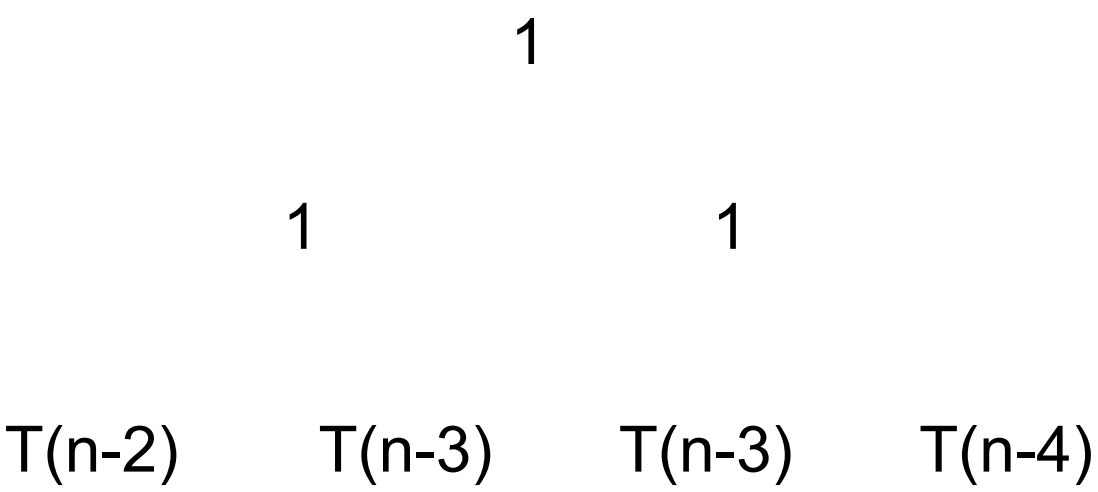
$T(n) = T(n-1) + T(n-2) + \Theta(1)$

$\Theta(1) = c$

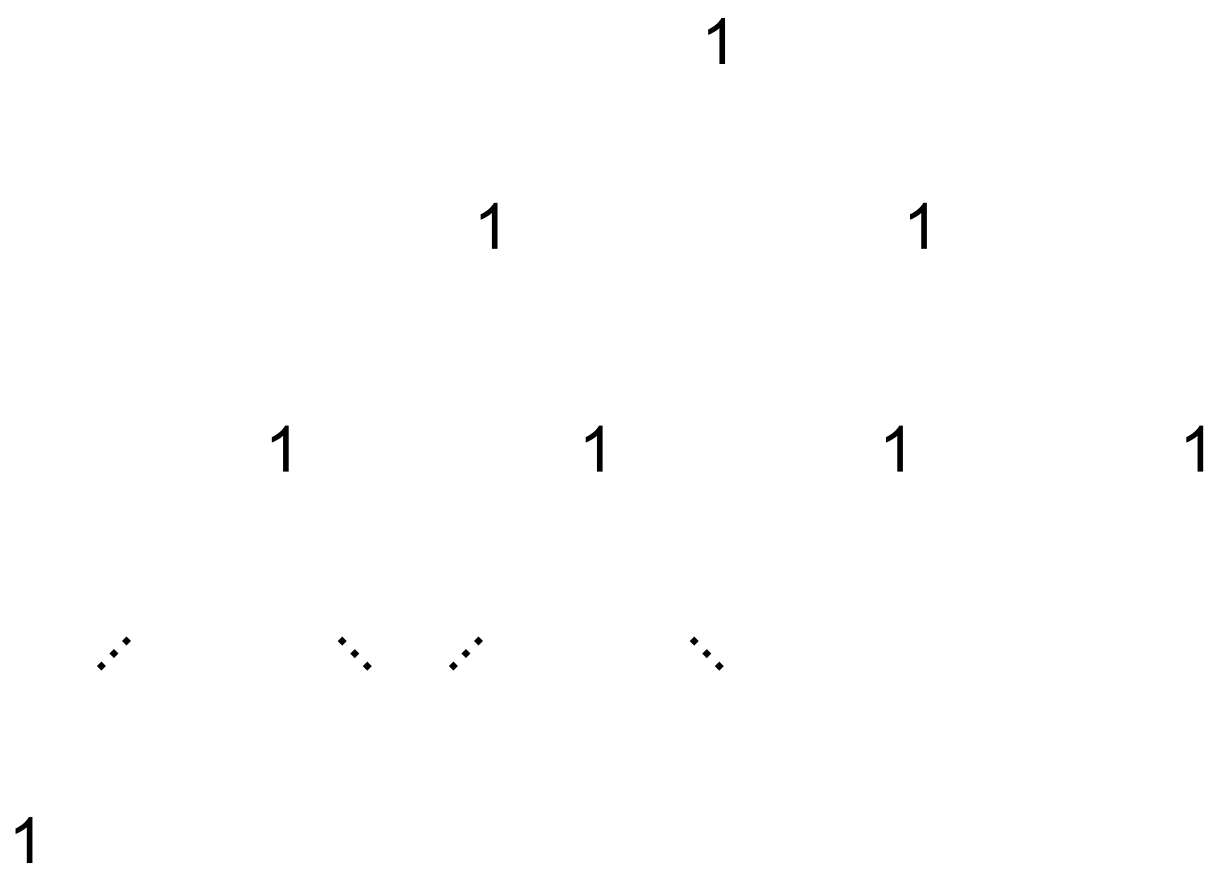
1

T(n-1) T(n-2)

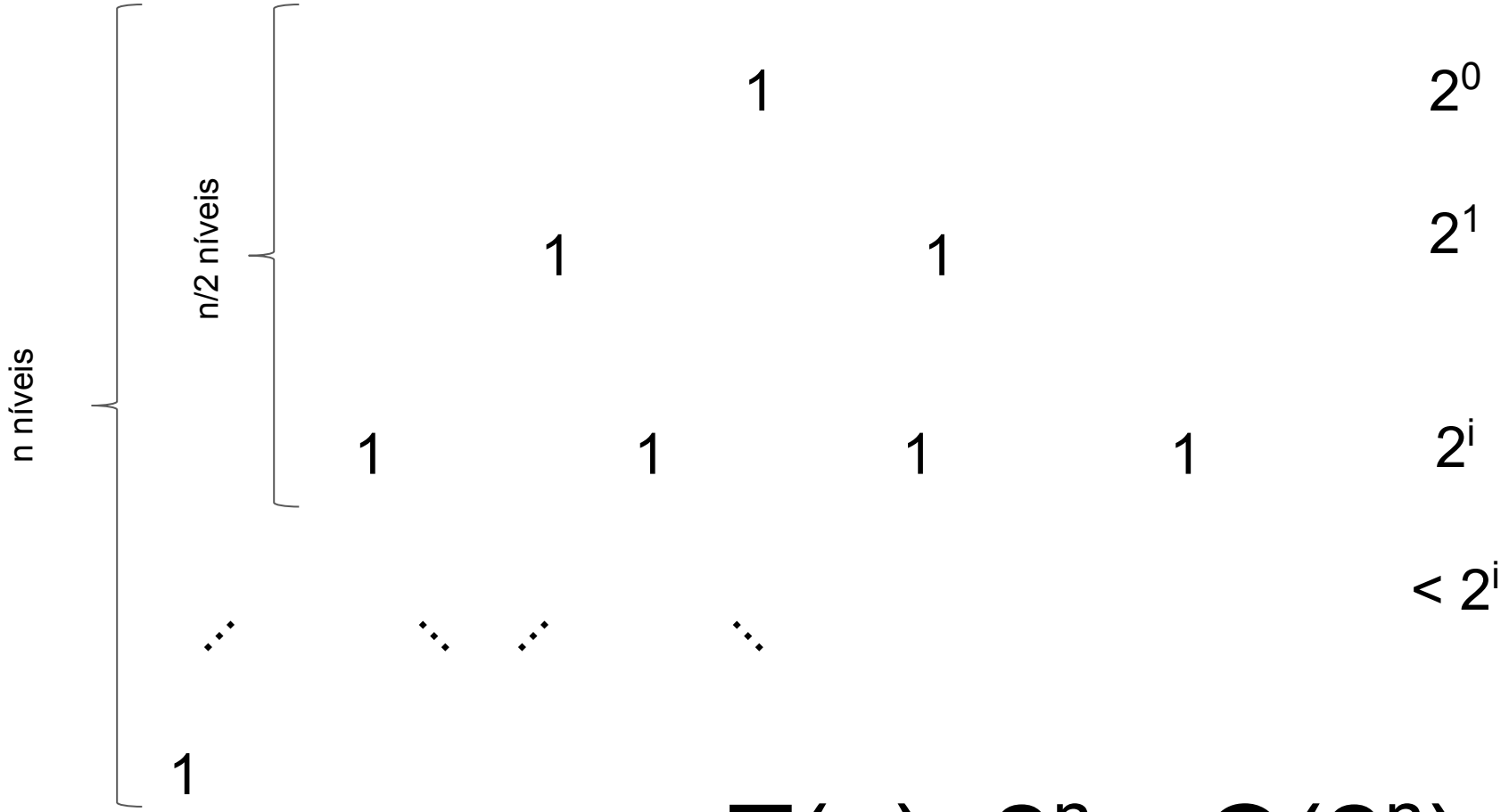
Árvore Recursiva para Fib(n)



Árvore Recursiva para Fib(n)



Árvore Recursiva para Fib(n)



$$T(n) < 2^n \in O(2^n)$$

Análise Imprecisa do Algoritmo Recursivo para Fib(n)

- $T(n) \in O(2^n)$ porém não é preciso
- Existe $g(n) < 2^n$ tal que $T(n) \in O(g(n))$

Análise Precisa do Algoritmo Recursivo para Fib(n)

- Fib(n) = Fib(n-1) + Fib(n-2) é uma *recorrência linear* no formato

$$x_n = A_1 x_{n-1} + A_2 x_{n-2} + \ldots + A_k x_{n-k}, \text{ onde A é um constante}$$

Análise Precisa do Algoritmo Recursivo para Fib(n)

- Fib(n) = Fib(n-1) + Fib(n-2) é uma *recorrência linear* no formato
- Pode ser traduzido em uma equação linear de raízes x e x'

$$\text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \rightarrow k=2 \text{ termos}$$

$$x^k = x^{k-1} + x^{k-2}$$

$$x^2 = x^1 + x^0$$

$$x^2 = x^1 + 1 \rightarrow x'=(1+5^{1/2})/2 \text{ e } x'=(1-5^{1/2})/2$$

Análise Precisa do Algoritmo Recursivo para Fib(n)

- Fib(n) = Fib(n-1) + Fib(n-2) é uma *recorrência linear*
- Pode ser traduzido em uma equação linear de raízes x e x'
- Pode ser re-escrito como Fib(n) = $(x)^n + (x')^n$

$$Fib(n) = ((1+5^{1/2})/2)^n + ((1-5^{1/2})/2)^n$$

$$Fib(n) = ((1+5^{1/2})/2)^n + ((1-5^{1/2})/2)^n$$

$$T(n) \simeq Fib \in O((1+5^{1/2})/2)^n + ((1-5^{1/2})/2)^n)$$

$$T(n) \simeq Fib \in O((1+5^{1/2})/2)^n = O(\boldsymbol{\varphi}^n)$$

Algoritmo Iterativo para Fib(n)

```
1.  função Fib(n)
2.        n1 = 1
3.        n2 = 1
4.        n3 = 1
5.
6.        para i = 3, 4, …, n
7.              n3 =  n1 + n2
8.              n1 = n2
9.              n2 = n3
10.
11.        retorne n3
```

Análise do Algoritmo Iterativo para Fib(n)

```
1.  função Fib(n)
2.      n1 = 0
3.      n2 = 1
4.      n3 = 1
5.
6.      para i = 3, 4, …, n
7.          n3 =  n1 + n2
8.          n1 = n2
9.          n2 = n3
10.
11.     retorne n3
```

3 instruções

4n instruções para n > 2

1 instrução

$$T(n) = 4n + 4 \in \Theta(n)$$