Introdução

Prólogo

- Duas idéias mudaram o mundo
  1) Livros
  2) Algoritmos

Johann Gutenberg
1398–1468

Fibonacci

- Mais famosa sequência de números
  - Biologia, demografia, arte, arquitetura, música, etc...

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots,$$

$$F_n = \begin{cases} F_{n-1} + F_{n-2} & \text{if } n > 1 \\ 1 & \text{if } n = 1 \\ 0 & \text{if } n = 0. \end{cases}$$

$$F_n \approx 2^{0.694n}$$

Leonardo of Pisa (Fibonacci)
1170–1250

Algoritmo 1

- Pseudocódigo

```
function fib1(n)
if n = 0: return 0
if n = 1: return 1
return fib1(n - 1) + fib1(n - 2)
```

  - É correto?
  - Quanto tempo demora?
  - Podemos fazer melhor?

$T(n) \longrightarrow$ Passos computacionais

$T(n) \geq F_n$.

$T(200) \geq F_{200} \geq 2^{138}$

Algoritmo 1

Processador de 10GHz

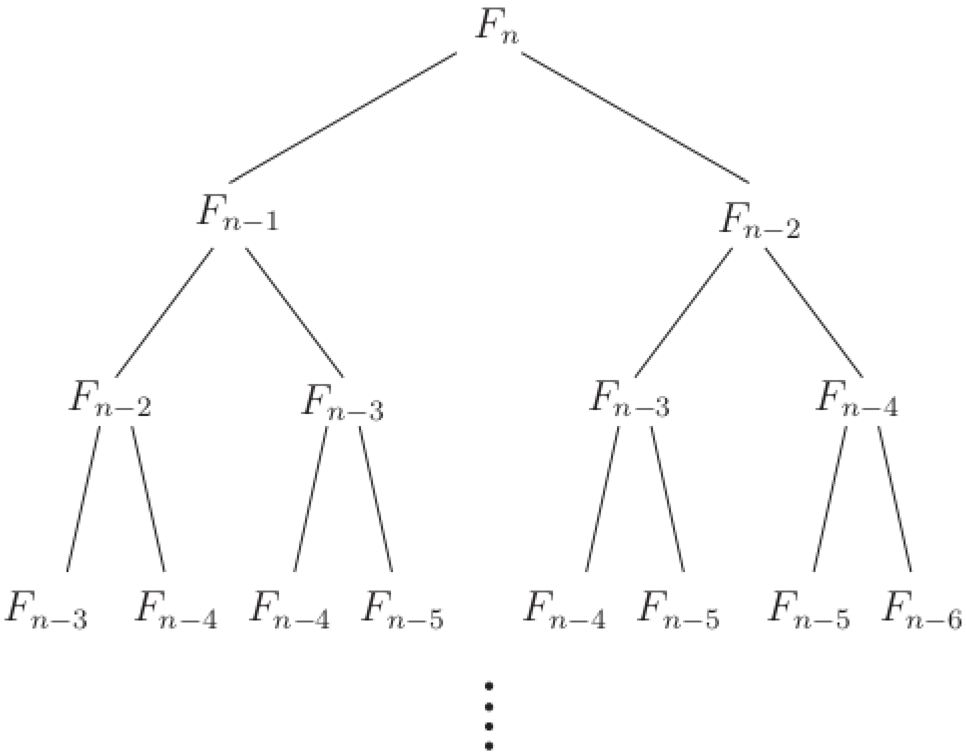
- $2^{10} \sim= 10^3 \longrightarrow 2^{138} \sim= 10^{41,4} \longrightarrow 10^{31}s \longrightarrow$ 3 milênios

	10	100	10^3	10^4	10^5	10^6
$\log_2 n$	3	6	9	13	16	19
n	10	100	1000	10^4	10^5	10^6
$n \log_2 n$	30	664	9965	10^5	10^6	10^7
n^2	100	10^4	10^6	10^8	10^{10}	10^{12}
n^3	10^3	10^6	10^9	10^{12}	10^{15}	10^{18}
2^n	10^3	10^{30}	10^{300}	10^{300}	10^{3000}	10^{300000}

1 ano = $365 \times 24 \times 60 \times 60 \approx 3 \times 10^7$ segundos
1 século $\approx 3 \times 10^9$ segundos
1 milénio $\approx 3 \times 10^{10}$ segundos

Algoritmo 1

- Por quê?



Algoritmo 2

- Pseudocódigo

```
function fib2(n)
if n = 0: return 0
create an array f[0...n]
f[0] = 0, f[1] = 1
for i = 2...n:
   f[i] = f[i-1] + f[i-2]
return f[n]
```

Loop executado (n-1) vezes

O Algoritmo certo faz toda a diferença!

- Podemos fazer melhor?

Análise Assintótica

- Notação O
  - O define um limite superior para a função, por um fator constante.
  - g(n) = O(f(n))
  - Lê-se:
    - g(n) é de ordem no máximo f(n)
    - f(n) domina assintoticamente g(n)
    - (f(n) é um limite assintótico superior para g(n))
  - Formalmente:
    - g(n) = O(f(n)), c > 0 e n0 | 0 <= g(n) <= c.f(n), ∀ n >= n0

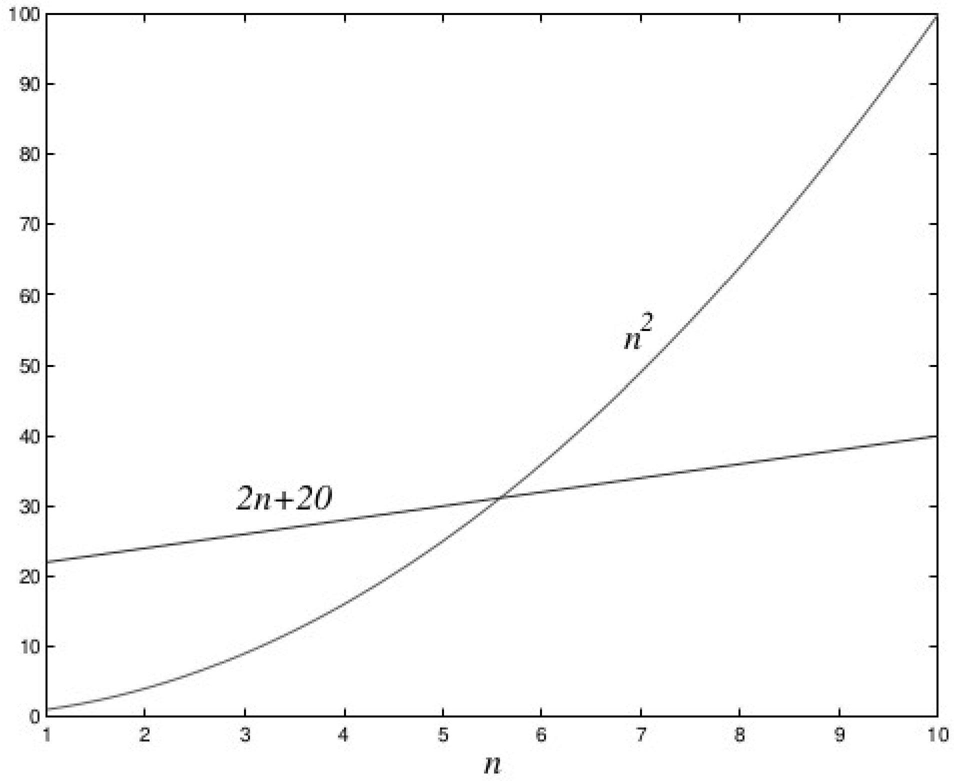
Análise Assintótica

- Exemplo:

$$f_1(n) = n^2$$
$$f_2(n) = 2n + 20$$

$$f_2 = O(f_1),$$

$$\frac{f_2(n)}{f_1(n)} = \frac{2n+20}{n^2} \leq 22$$



Análise Assintótica

- Notação Ω
  - Ω define um limite inferior para a função, por um fator constante.
  - $g(n) = \Omega(f(n))$
  - Formalmente:
    - $g(n) = \Omega(f(n))$, $c > 0$ e $n0 \mid 0 <= c.f(n) <= g(n)$, $\forall\ n >= n0$

Análise Assintótica

- Notação Θ
  - A notação Θ limita a função por fatores constantes
  - g(n) = Θ(f(n)) se existirem constantes positivas c1 e c2 e n0 tais que para n >= n0, o valor de g(n) está sempre entre c1.f(n) e c2.f(n) inclusive.
  - Formalmente:

$$g(n) = \Theta(f(n)),\ c1 > 0 \text{ e } c2 > 0 \text{ e } n0 \mid$$

$$0 <= c1.f(n) <= g(n) <= c2.f(n),\ \forall\ n >= n0$$

Exercícios

- Indique se f(n) = O(g(n)), ou se f(n) = Ω(g(n)), ou se f(n) = Θ(g(n))

	$f(n)$	$g(n)$
(a)	$n - 100$	$n - 200$
(b)	$n^{1/2}$	$n^{2/3}$
(c)	$100n + \log n$	$n + (\log n)^2$
(d)	$n \log n$	$10n \log 10n$
(e)	$\log 2n$	$\log 3n$
(f)	$10 \log n$	$\log(n^2)$
(g)	$n^{1.01}$	$n \log^2 n$
(h)	$n^2 / \log n$	$n(\log n)^2$
(i)	$n^{0.1}$	$(\log n)^{10}$
(j)	$(\log n)^{\log n}$	$n / \log n$
(k)	$\sqrt{n}$	$(\log n)^3$
(l)	$n^{1/2}$	$5^{\log_2 n}$
(m)	$n2^n$	3^n

$$f = \Omega(g) \text{ means } g = O(f)$$
$$f = \Theta(g) \text{ means } f = O(g) \text{ and } f = \Omega(g).$$