

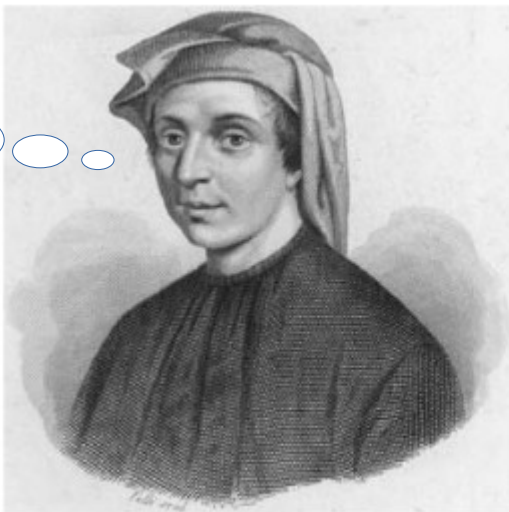
# **Aula 2**

## **Projeto de Algoritmos**

# Introdução

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

O Algoritmo  
certo faz toda a  
diferença!



```
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  $f[n]$ 
```

Leonardo of Pisa (Fibonacci)  
1170–1250

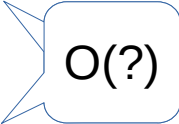
# Divide and Conquer

- 1) Quebra o problema em subproblemas
- 2) Resolve os subproblemas usando o próprio algoritmo que está sendo definido
- 3) Combina as soluções das instâncias menores para produzir uma solução da instância original

# Busca Binária

- Encontrar um número numa longa lista de números

- Lista não ordenada → **Percorrer a lista toda**



$O(?)$

- Lista ordenada → Se percorrer a lista toda?

**Podemos fazer melhor?**

# Busca Binária

- Algoritmo recursivo:

BB-R ( $A, p, r, x$ )

1 se  $p = r-1$

2 devolva  $r$  e pare

3  $q := \lfloor (p+r)/2 \rfloor$

4 se  $A[q] < x$

5 devolva BB-R ( $A, q, r, x$ )

6 senão devolva BB-R ( $A, p, q, x$ )

Divide

Conquista

Combina?

# Busca Binária

- Exemplo

0	1	2	3	4	5	6	7	8	9
-8	-5	1	4	14	21	23	54	67	90

4 Elemento procurado

0	1	2	3	4	5	6	7	8	9
-8	-5	1	4	14	21	23	54	67	90

0	1	2	3	4	5	6	7	8	9
-8	-5	1	4	14	21	23	54	67	90

0	1	2	3	4	5	6	7	8	9
-8	-5	1	4	14	21	23	54	67	90

0	1	2	3	4	5	6	7	8	9
-8	-5	1	4	14	21	23	54	67	90

# Busca Binária

- Análise de complexidade

BB-R ( $A, p, r, x$ )

```
1  se  $p = r-1$ 
2    devolva  $r$  e pare
3   $q := \lfloor (p+r)/2 \rfloor$ 
4  se  $A[q] < x$ 
5    devolva BB-R ( $A, q, r, x$ )
6  senão devolva BB-R ( $A, p, q, x$ )
```

1

1

1

1

$T(\lfloor n/2 \rfloor)$

$T(\lfloor n/2 \rfloor - 1)$

Logaritmando

Logaritmo

$$\text{Log}_a b = x \Leftrightarrow a^x = b$$

Base

$$T(n) = O(\lg n)$$

# Mergesort

- Clássico problema de ordenar um vetor  $A[1..n]$ 
  - Mergesort usa a estratégia da divisão e conquista

MERGESORT ( $A, p, r$ )

Divide

1 se  $p < r$

2  $q := \lfloor (p+r)/2 \rfloor$

3 MERGESORT ( $A, p, q$ )

Conquista

4 MERGESORT ( $A, q+1, r$ )

5 INTERCALA ( $A, p, q, r$ )

Combina



# Mergesort

```
INTERCALA (A, p, q, r)
  6  para i := p até q
  7      B[i] := A[i]
  8  para j := q+1 até r
  9      B[r+q+1-j] := A[j]
 10  i := p
 11  j := r
 12  para k := p até r
 13      se B[i] ≤ B[j]
 14          A[k] := B[i]
 15          i := i+1
 16      senão A[k] := B[j]
 17          j := j-1
```

```
MERGESORT (A, p, r)
```

```
1  se p < r
2      q := ⌊(p+r)/2⌋
3      MERGESORT (A, p, q)
4      MERGESORT (A, q+1, r)
5      INTERCALA (A, p, q, r)
```

- recebe vetores crescentes

$A[p..q]$  e  $A[q+1..r]$

e rearranja o vetor

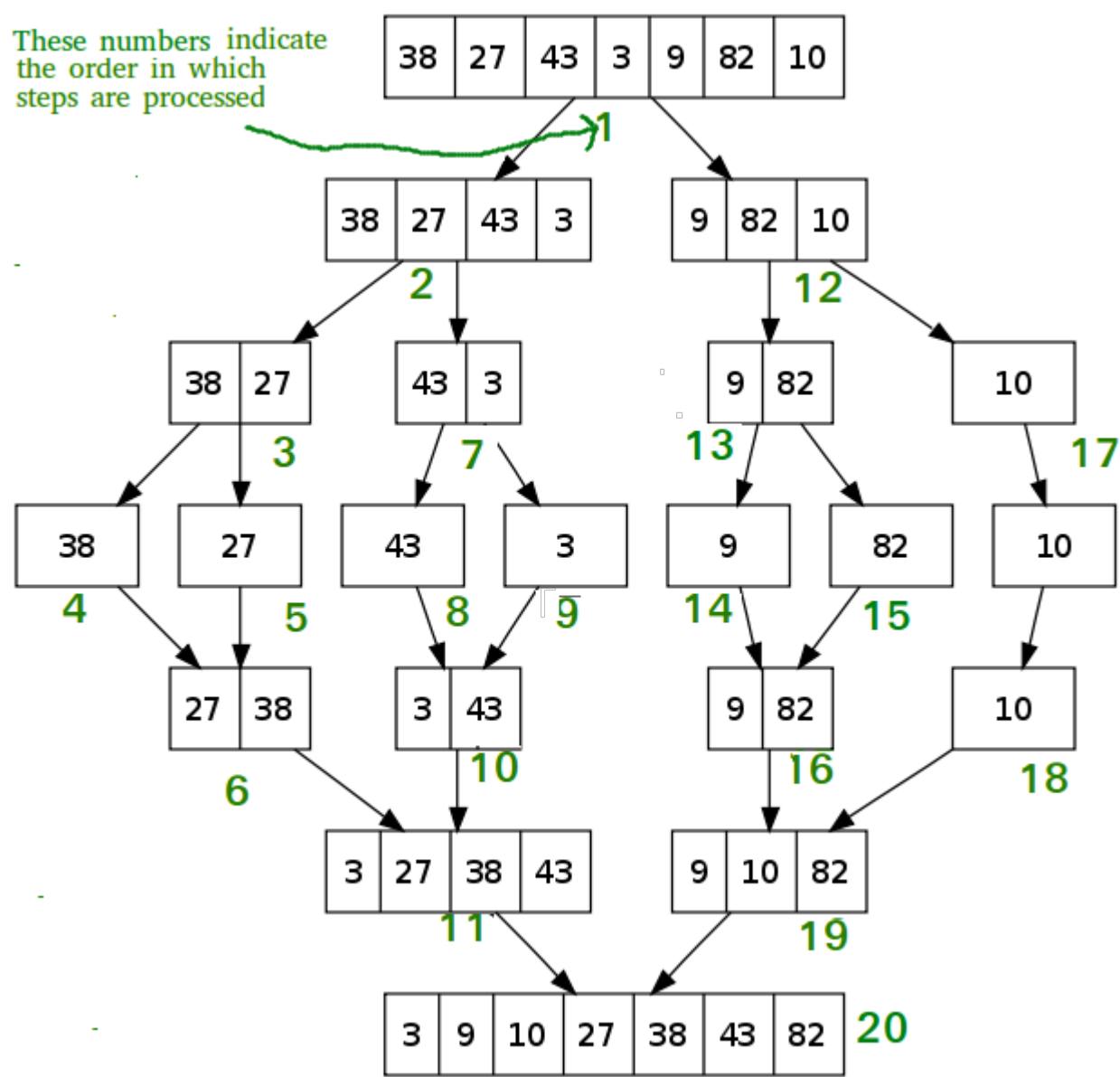
$A[p..q..r]$

de modo que ele fique crescente

- Exemplo:

38	27	43	3	9	82	10
----	----	----	---	---	----	----

These numbers indicate  
the order in which  
steps are processed



# Mergesorte

- Análise de complexidade

MERGESORT ( $A, p, r$ )

1 se  $p < r$

2  $q := \lfloor (p+r)/2 \rfloor$

3 MERGESORT ( $A, p, q$ )

4 MERGESORT ( $A, q+1, r$ )

5 INTERCALA ( $A, p, q, r$ )

1

1

$T(\lfloor n/2 \rfloor)$

$T(\lfloor n/2 \rfloor)$

$6n+5$

$$T(n) = \Theta(n \lg n).$$

$$T(n) = T(\lfloor n/2 \rfloor) + T(\lfloor n/2 \rfloor) + 6n + 7$$

$$\leq 16 \lfloor n/2 \rfloor \lg \lfloor n/2 \rfloor + 16 \lfloor n/2 \rfloor \lg \lfloor n/2 \rfloor + 6n + 7$$

$$\leq 16 (\lfloor n/2 \rfloor + \lfloor n/2 \rfloor) \lg \lfloor n/2 \rfloor + 6n + 7$$

$$\leq 16 n \lg (2n/3) + 6n + 7$$

$$= 16 n (\lg n + \lg (2/3)) + 6n + 7$$

$$< 16 n (\lg n - 0.5) + 6n + 7$$

$$= 16 n \lg n - 8n + 6n + 7$$

$$= 16 n \lg n - 2n + 7$$

$$\leq 16 n \lg n, \text{ CQD.}$$