

# Análise de Algoritmos $\leftrightarrow$ ideia p/ resolver um problema.

$\xrightarrow{\text{tempo}}$

Uma entrada de um tamanho específico ( $n$ )

↳ Encontro qual tempo (qtd de instruções de altíssimo nível) necessário para resolver algo de tamanho  $n$ .

→ Funções de complexidade.

$\begin{matrix} \downarrow & \downarrow & \downarrow \\ 2, & 3, & 4 \end{matrix}$

Alg 1. Maior

Entrada: um vetor  $V$  com  $n$  elementos **pior caso**

**Melhor**

- |   |                              |                              |
|---|------------------------------|------------------------------|
| - 1. maior $\leftarrow V[1]$                          | $\rightarrow 1$              | $\rightarrow 1$              |
| - 2. for $i \leftarrow 2$ to <u><math>n</math></u> do | $\rightarrow n \times 1$     | $\rightarrow n \times 1$     |
| - 3.   if maior $< V[i]$ then                         | $\rightarrow (n-1) \times 1$ | $\rightarrow (n-1) \times 1$ |
| 4.    ↳ maior $\leftarrow V[i]$                       | $\rightarrow (n-1) \times 1$ | $\rightarrow 0$              |
| 5. return maior                                       | $\rightarrow 1$              | $\rightarrow 1$              |

$$\begin{aligned} &= 1 + n + 2(n-1) \cdot 1 + 1 \\ &= 2 + n + 2n - 2 \\ &= 3n \end{aligned}$$

$$\begin{aligned} &= 1 + n + (n-1) + 1 \\ &= 2n + 1 \end{aligned}$$

$$f(n) = 2n + 1$$

$$f(n) = \underline{\underline{cn + c}}$$

$$f(n) = \boxed{3n}$$

complexidade

$$f(n) = n$$

constante tecnológica ( $n$  depende de  $n$ )

$$\boxed{f(n) = \cancel{c} \cdot n} \quad \text{função de complexidade}$$

Perspectivas de complexidade (melhor caso, pior caso, caso médio).  
 $\begin{matrix} \times & \downarrow & \checkmark \\ & + \text{tempo} & \end{matrix}$

6

7

8

## Alg 2. Busca sequencial

Entrada: um vetor  $V$  "de tamanho"  $n$ , e uma chave " $k$ "

1. for  $i \leftarrow 1$  to  $n$  do

2. | if  $V[i] = k$  then

3. | | return  $i$

4. return  $-1$

Melhor

Pior Caso

$\Delta 1$

$\Delta (n+1) \times 1$

$\Delta 1$

$\Delta n \times 1$

$\Delta 1$

$f(n) = 3$

$\Delta 1$

$f(n) = c$

$f(n) = 2n + 2$

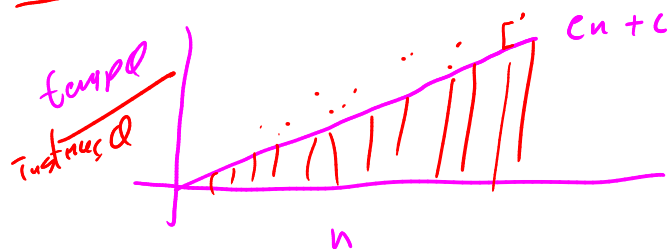
$f(n) = cn + c$

## Ordens Assintóticas / comparar funções

função	números	
$O(\text{micron})$	$\leq$	
$\Theta(\text{theta})$	$=$	
$\Omega(\text{omega})$	$\geq$	

Complexidade da busca sequencial é  $O(cn + c)$   
 $\leq cn + c$

Complexidade da busca sequencial é  $\Omega(c)$   
 $\geq c$



Quando  $O(\cdot)$  é igual  $\Omega(\cdot)$  são sensivelmente

diferentes (diferença é uma constante independente de  $n$ ), então use a ordem  $\Theta$

Alg 1 possui complexidade de tempo  $\Theta(cn) \leftrightarrow \Theta(n)$ , mas também poderia dizer que é  $\Omega(n)$  e  $O(n)$ .

# Escalas de Complexidade:

menor tempo

maior tempo

$c, \log n, n, n \log n, n^2, n^3, \dots, 2^n, 3^n, n!, n^n$

Problema X

Alg a

$\Theta(n^2)$

Alg b

$O(n \log n)$

Alg c

~~$\Omega(n)$~~

n conhece os demais casos

## Cap 2. Representações Computacionais (implementar um grafo)

Matriz de adjacência

	1	2	3	4	5
1	0	0	1	0	0
2	0	0	0	1	1
3	1	0	0	1	0
4	0	1	1	0	1
5	0	1	0	1	0

Lista de adjacência

