

## O que é algoritmo?

- É uma sequência de passos computacionais que transformam uma entrada em uma saída com o objetivo de resolver um problema computacional.
- Conjunto de Regras e Procedimentos Lógicos perfeitamente definidos que levam à solução de um problema.

## Problemas NP-completos

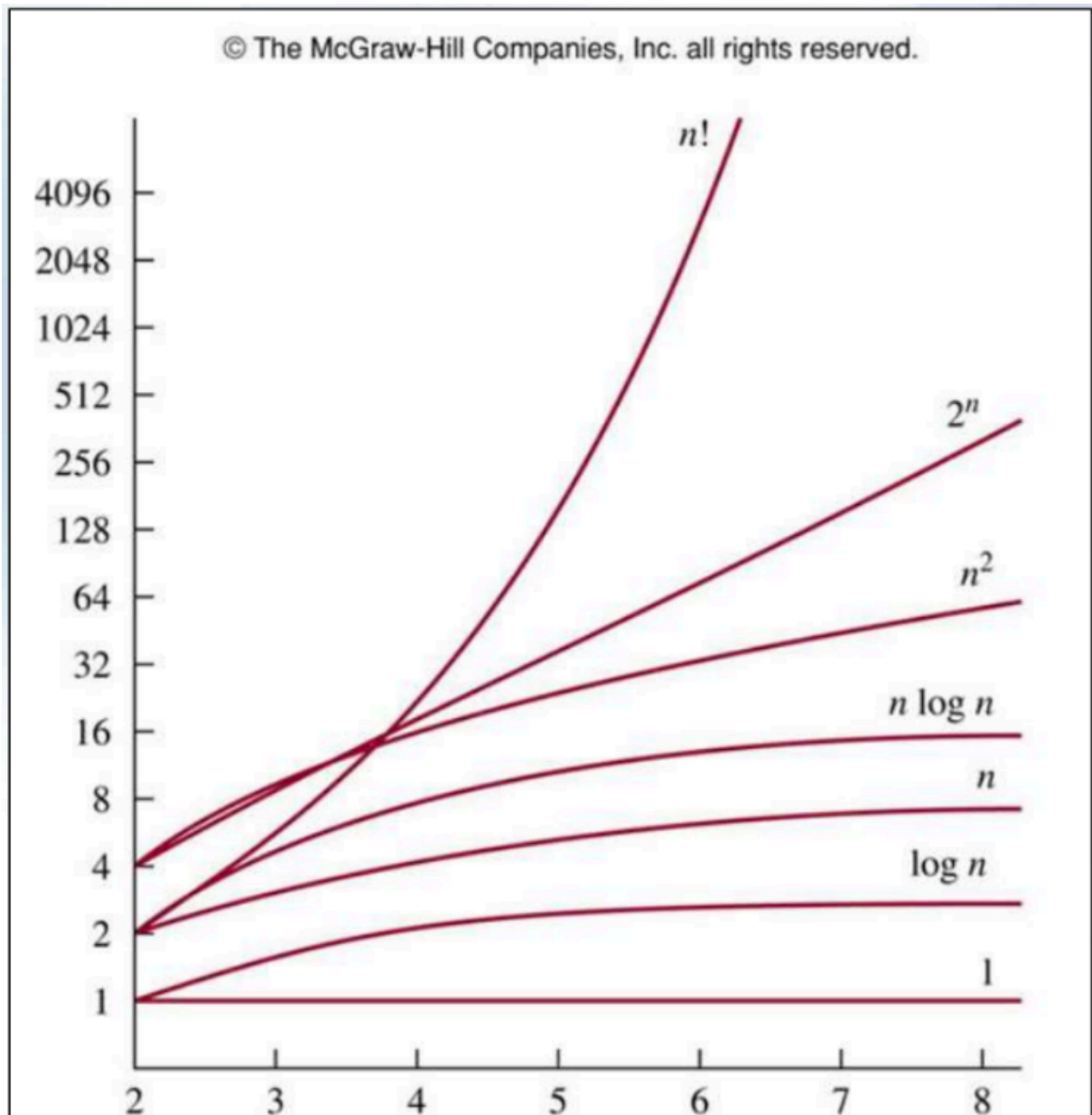
- Não se sabe se existem algoritmos eficientes para eles.
- Se existir a solução para um deles. Existe solução para todos eles.

## Modelo Computacional Genérico - Arquitetura de Von Neumann

- 1 processador
- Memória RAM
- Operações sequenciais
- Todas as instituições têm mesmo custo

## Análise de Algoritmo

- Analisar os recursos que um algoritmo necessitará (Tempo e Memória)
- Sempre é mais importante estudar o Pior Caso e às vezes o Caso Médio.



- **Lower Bound ou Limitante Inferior (LB(P)):** Complexidade Mínima para resolver um problema (Big-Ω).
- **Upper Bound ou Limitante Superior (UB(P)):** Complexidade do melhor algoritmo para resolver o problema (Big-O).
- Um problema é computacionalmente resolvido se **UB(P)** pertence ao domínio de **LB(P)**.

# Insertion Sort

Exemplo Ilustrativo:

i	0	1	2	3	4	5
Vet[i]	60	30	40	50	90	80
	S1			S2		

Primeira Iteração:

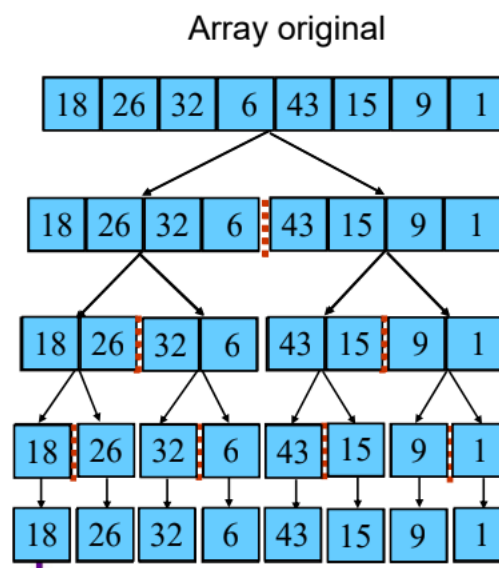
i	0	1	2	3	4	5
Vet[i]	60	<u>30</u>	40	50	90	80
	S1			S2		

i	0	1	2	3	4	5
Vet[i]	30	60	<u>40</u>	50	90	80
	S1			S2		

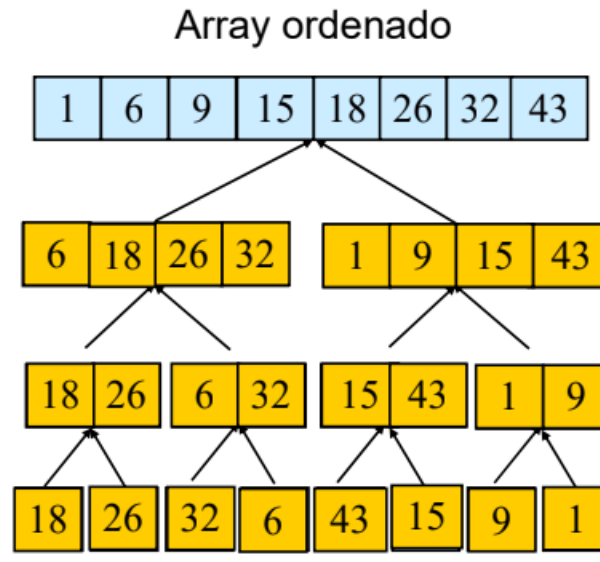
COMPLEXIDADE:  $N^2$

## Divisão e Conquista:

- **Divisão:** É dividir o problema até que cheguemos em uma solução trivial (ex: Array de tamanho 1).

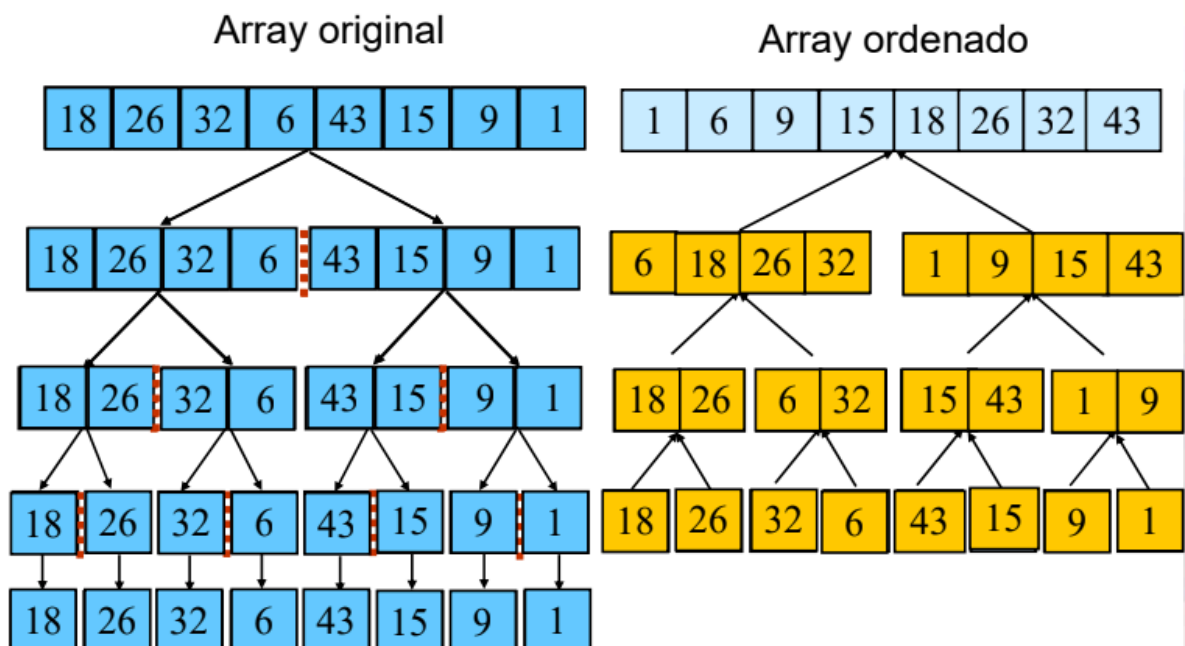


- **Conquista:** Solução do problema nestes arrays menores
- **Combinação (Opcional):** Reunião dos Arrays (agora resolvidos) para um único array novamente.



## Merge Sort

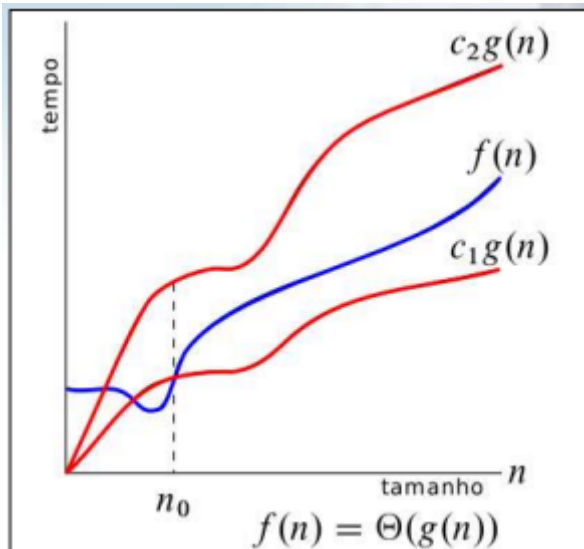
- Algoritmo de Ordenação por método de Divisão e Conquista



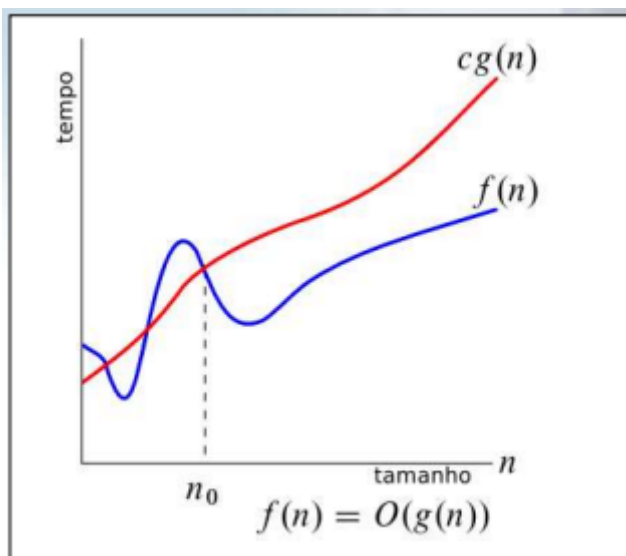
- **COMPLEXIDADE:**  $n \log n$

## Notação Assintótica:

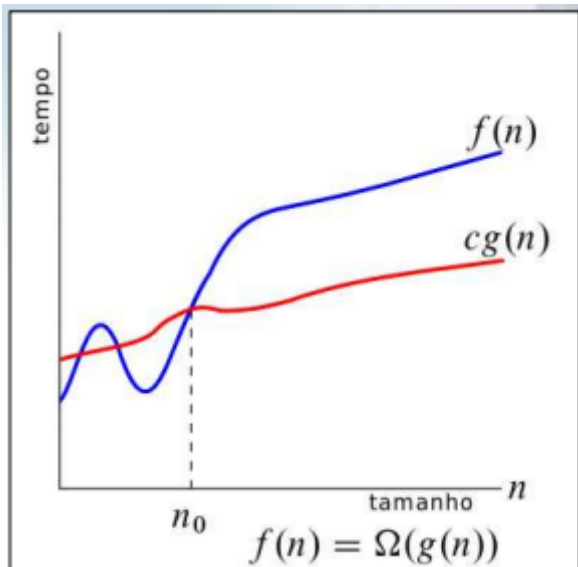
- **CLASSE  $\Theta$  (Big- $\Theta$ ):**  $0 \leq c_1.g(n) \leq f(n) \leq c_2.g(n)$



- **CLASSE  $O$  (Big- $O$ ):**  $0 \leq f(n) \leq c.g(n)$



- **CLASSE  $\Omega$  (Big- $\Omega$ ):**  $0 \leq c.g(n) \leq f(n)$



- CLASSE  $o$  (Big- $o$ ):  $0 \leq f(n) < c \cdot g(n)$
- CLASSE  $\omega$  (Big- $\Omega$ ):  $0 \leq c \cdot g(n) < f(n)$

■ **Analogia** entre duas funções **f** e **g** e dois números **a** e **b**:

- $f(n) = O(g(n))$       equivalente       $a \leq b$
- $f(n) = \Omega(g(n))$       equivalente       $a \geq b$
- $f(n) = \Theta(g(n))$       equivalente       $a = b$
- $f(n) = o(g(n))$       equivalente       $a < b$
- $f(n) = \omega(g(n))$       equivalente       $a > b$