# Fundamentos de Programação

## Aula 05 - Noções de Complexidade

**Profª. Elisa de Cássia Silva Rodrigues**

# Sumário

- Introdução.
- Complexidade assintótica.
- Notação assintótica.
- Classes de complexidade assintótica.
- Atividade prática.
- Atividade para casa.

# Introdução

- Complexidade Assintótica
  - Análise de tempo feita sobre entrada com tamanhos grandes.
  - Mede o crescimento do tempo de execução conforme o tamanho da entrada $n$ aumenta indefinidamente.
- Notação Assintótica
  - Representa o comportamento assintótico de funções.
  - Relaciona funções de complexidade de tempo de dois algoritmos.
  - O domínio das funções é o conjunto dos naturais $\mathcal{N} = \{0, 1, 2, ...\}$.
    - **Ex:** $f(n) = n^2 + 2$, onde $n \in \mathcal{N}$.

**Definição**

Uma função $g(n)$ domina assintoticamente uma função $f(n)$ se existem duas constantes positivas $c$ e $n_0$ tais que $|f(n)| \leq c \cdot |g(n)|$, para todo $n \geq n_0$.

- **Exemplo:** $f(n) = n$ e $g(n) = -n^2$.
  - Para $c = 1$ e $n_0 = 1$, temos $|n| \leq 1 \cdot |-n^2|$, para todo $n \geq 1$.
  - Portanto, $g(n)$ domina assintoticamente $f(n)$.

| $n$ | $\|n\| \leq \cdot \|-n^2\|$ |
|---|---|
| 1 | $1 \leq 1$ |
| 2 | $2 \leq 4$ |
| 3 | $3 \leq 9$ |
| 4 | $4 \leq 16$ |

Será que $f(n)$ domina assintoticamente $g(n)$?

# Notação Assintótica

**Notação $\mathcal{O}$**

Uma função $f(n)$ é $\mathcal{O}(g(n))$ se existem duas constantes $c$ e $n_0$ tais que $f(n) \leq c \cdot g(n)$, para todo $n \geq n_0$. (Outras representações: $f(n) = \mathcal{O}(g(n))$ ou $f(n) \in \mathcal{O}(g(n))$).

- Utilizada para dar um limite assintótico superior sobre uma função.
- Graficamente, para todos os valores $n \geq n_0$ (à direita de $n_0$), o valor de $f(n)$ está em ou abaixo de $g(n)$.

- **Exemplo:** $f(n) = (n+1)^2$ e $g(n) = n^2$.
  - Para $c = 3$ e $n_0 = 2$, temos $|(n+1)^2| \leq 3 \cdot |n^2|$, para todo $n \geq 2$.
  - Portanto, $f(n)$ é $\mathcal{O}(n^2)$.

**Notação $\Omega$**

Uma função $f(n)$ é $\Omega(g(n))$ se existem duas constantes $c$ e $n_0$ tais que $c \cdot g(n) \leq f(n)$, para todo $n \geq n_0$. (Outras representações: $f(n) = \Omega(g(n))$ ou $f(n) \in \Omega(g(n))$).

- Utilizada para dar um limite assintótico inferior sobre uma função.
- Graficamente, para todos os valores $n \geq n_0$ (à direita de $n_0$), o valor de $f(n)$ está em ou acima de $g(n)$.

- **Exemplo:** $f(n) = 2n^3 + 3n^2 + n$ e $g(n) = n^3$.
  - Para $c = 1$ e $n_0 = 1$, temos $|2n^3 + 3n^2 + n| \leq 1 \cdot |n^3|$, para todo $n \geq 1$.
  - Portanto, $f(n)$ é $\Omega(n^3)$.

# Notação Assintótica

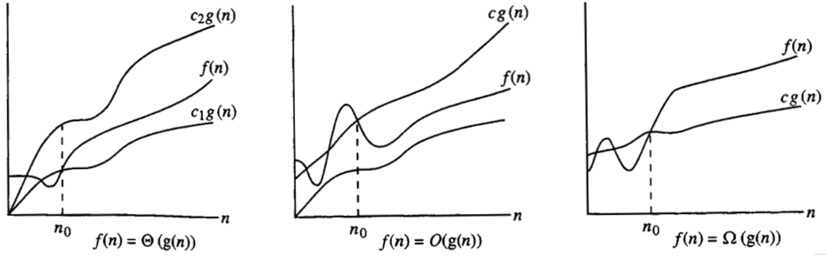
**Notação $\Theta$**

Uma função $f(n)$ é $\Theta(g(n))$ se existem três constantes $c_1$, $c_2$ e $n_0$ tais que $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$, para todo $n \geq n_0$. (Outras representações: $f(n) = \Theta(g(n))$ ou $f(n) \in \Theta(g(n))$).

- Utilizada para dar um limite assintótico firme sobre uma função.
- Graficamente, para todos os valores $n \geq n_0$ (à direita de $n_0$), o valor de $f(n)$ está sobre ou acima de $c_1 \cdot g(n)$ e sobre ou abaixo de $c_2 \cdot g(n)$.

- **Exemplo:** $f(n) = (1/2)n^2 - 3n$ e $g(n) = n^2$.
  - Para $c_1 = 1/8$, $c_2 = 1/2$ e $n_0 = 8$, para todo $n \geq 8$, temos $(1/8) \cdot |n^2| \leq |(1/2)n^2 - 3n| \leq (1/2) \cdot |n^2|$.
  - Portanto, $f(n)$ é $\Theta(n^2)$.

# Notação Assintótica



# Notação Assintótica

## Notação $o$

- Limite superior não assintoticamente restrito sobre uma função.
- $f(n)$ torna-se insignificante em relação a $g(n)$ conforme $n$ se aproxima de $\infty$.
- **Exemplo:** $2n = o(n^2)$ e $2n^2 \neq o(n^2)$, mas $2n^2 = \mathcal{O}(n^2)$.

## Notação $\omega$

- Limite inferior não assintoticamente restrito sobre uma função.
- $f(n)$ torna-se muito grande em relação a $g(n)$ conforme $n$ se aproxima de $\infty$.
- **Exemplo:** $n^2/2 = \omega(n)$ e $n^2/2 \neq \omega(n^2)$, mas $n^2/2 = \Omega(n^2)$.

# Classes de Complexidade Assintótica

- **O(1):** ordem constante.
  - Instruções são executadas um número fixo de vezes.
  - Não depende do tamanho dos dados de entrada.
- **O($logn$):** ordem logarítmica.
  - Resolve um problema transformando-o em problemas menores.
- **O($n$):** ordem linear.
  - Operações são realizadas sobre cada um dos elementos de entrada.
- **O($n \cdot logn$):** ordem log linear.
  - Algoritmos que trabalham com particionamento dos dados.
  - Transformam um problema em vários problemas.
  - Cada problema é resolvido de forma independente e depois unidos.

# Classes de Complexidade Assintótica

- **O($n^2$):** ordem quadrática.
  - Ocorre quando os dados são processados aos pares.
  - Caracterizado pela presença de dois comandos de repetição aninhadas.
- **O($n^3$):** ordem cúbica.
  - Caracterizado pela presença de três estruturas de repetição aninhadas.
- **O($2^n$):** ordem exponencial.
  - Ocorre quando se usa uma solução de força bruta.
  - Não são úteis do ponto de vista prático.
- **O($n!$):** ordem fatorial.
  - Possuem comportamento muito pior que o exponencial.

# Atividade Prática

- Pesquise na internet 1 exemplo de algoritmo para cada classe:
  - **$O(1)$:** ordem constante.
  - **$O(log n)$:** ordem logarítmica.
  - **$O(n)$:** ordem linear.
  - **$O(n \cdot log n)$:** ordem log linear.
  - **$O(n^2)$:** ordem quadrática.
  - **$O(n^3)$:** ordem cúbica.
  - **$O(2^n)$:** ordem exponencial.
  - **$O(n!)$:** ordem fatorial.

# Atividade para Casa...

- Leia o seguinte tópico do livro texto (Cormen, 2012):
  - **Seção 3.2 - Notações padrão e funções comuns.**

**Sugestão de exercícios:**

- **Capítulo 3 - Crescimento de funções.**

**Observação**

Não precisa entregar!

# Próxima aula...

- Métodos de ordenação interna.
  - Problema da ordenação.
  - Ordenação por inserção.

**Sugestão de leitura:**

- **Capítulo 1 - A função dos algoritmos na computação.**
- **Sessão 2.1 - Ordenação por inserção.**

# Bibliografia


1. CORMEN, Thomas H. et al. ***Algoritmos: Teoria e Prática***. 2ª ed. 2012.
2. ASCÊNCIO, A. F. G.; ARAÚJO, Graziela S. ***Estrutura de Dados: Algoritmos, Análise de Complexidade, Implementações em Java e C/C++***. 2010.


**Obrigada pela atenção!**