Lista de Exercícios**: Análise Assintótica**

A **notação O** descreve como o **tempo de execução ou uso de memória cresce** em relação ao **tamanho da entrada (n)**. Ela não **mede** em segundos, mas sim em **ordens de crescimento**.

f**(n) =** O(**g(n)**).

**Notações**:

* **Big-O (O) LS - pior caso. 0 ≤ f(n) ≤** **c ⋅ g(n),**
* **Big-Omega (Ω)** **LI - melhor caso. 0 ≤ c ⋅ g(n) ≤ f(n),**
* **Big-Theta (Θ) LE - caso médio**. **0 ≤ c1 ⋅ g(n) ≤ f(n) ≤** **c2 ⋅ g(n),**

**Definição Formal**: Um algoritmo é O(**f(n)**) se existem **constantes** **c > 0 e n₀ ≥ 0** tais que: **T(n)** ≤ **c ⋅ f(n),** para todo **n ≥ n0**,​ Onde:

* **T(n)** é o tempo do algoritmo.
* **f(n)** é a função de crescimento.

**Exemplo 1**: 2n+1 = O(2n)?

T(n) = 2n+1; O(2n)

T(n) <= c.O(2n); c = 2; n0 =1

**Resposta**: **Verdadeiro**. Para mostrar que 2n+1 = O(2n), nós devemos encontrar **constantes** **c, n0 > 0** tal que 0 ≤ 2n+1 ≤ c · 2n ∀n ≥ n0. Uma vez que 2n+1 ≤ c · 2n ∀n, nós podemos resolver a desigualdade escolhendo c = 2 e n0 = 1.

**Exemplo 2**: 10n = O(n)? c = ? e n0 = ? ; **e para:** 5n + 3; c = ? e n = ?

**Calcule a função de crescimento e o valor de c e no das funções abaixo:**

**Exercício 1. Dada a função:**

T(n) = 3n2 + 5n + 2

Mostre que T(n) = O(n2), encontrando c e n0 tal que

T (n) ≤ c · n2 para todo n ≥ n0.

**Exercício 2. Considere:**

T (n) = 7n log2 n + 20

Mostre que T (n) = O(n log n) e determine c e n0.

**Exercício 3. Seja:**

T (n) = 4n3 + 2n2 − 5n + 10

Encontre c e n0 para comprovar que T (n) = O(n3).

**Exercício 4. Dada a função:**

T (n) = 10 · 2n + n5

Determine c e n0 para mostrar que T (n) = O(2n).

**Exercício 5. Considere:**

T (n) = n2 + 50n + 1000

Determine c e n0 tal que T (n) = O(n2).

**Exercício 6. Dada a função:**

T (n) = 5√n + log n + 20

Mostre que T (n) = O(√n), encontrando c e n0.

**Gabarito**

**Exercício 1.**

T (n) = 3n2 + 5n + 2 ≤ (3 + 5 + 2)n2 = 10n2 para n ≥ 1

c = 10, n0 = 1

**Exercício 2.**

T (n) = 7n log2 n+20 ≤ (7+20) n log2 n = 27n log2 n para n ≥ 2

c = 27, n0 = 2

**Exercício 3.**

T (n) = 4n3+2n2−5n+10 ≤ (4+2+5+10)n3 = 21n3 para n ≥1

c = 21, n0 = 1

**Exercício 4.**

Para n ≥ 23, portanto:

T (n) = 10 · 2n + n5 ≤ (10 + 1)2n = 11 · 2n

c = 11, n0 = 30; pode ser qq valor n0 >= 23

**Exercício 5.**

Para n ≥ 50, 50n + 1000 ≤ n2 + n2 = 2n2

T (n) ≈ O(n2)

c = 3, n0 = 50

**Exercício 6.**

Para n ≥ 400, log n + 20 ≤ √n + √n = 2√n

T (n) ≤ (5 + 2)√n = 7√n

c = 7, n0 = 400