TEOREMA MESTRE E EQUAÇÕES DE RECORRÊNCIA

LISTA DE EXERCÍCIOS

O **Teorema Mestre** é uma ferramenta usada para resolver recorrências da forma:

Forma geral: $\boldsymbol{T(n) = aT\left(\frac{n}{b}\right) + f(n)}$

onde: **Caso Particular:** $\boldsymbol{T(n) = a \cdot T\left(\frac{n}{b}\right) + \Theta\left(n^k\right)}$

- **a** ≥ 1 é o número de subproblemas;
- **b** > 1 é o fator de divisão do tamanho do problema;
- **f(n)** é o custo do trabalho fora das chamadas recursivas.
  - **Caso 1: se a > b^k**, então T(n) = $\boldsymbol{\Theta(n^{\log_b a})}$
  - **Caso 2: se a = b^k**, então T(n) = $\boldsymbol{\Theta\left(n^k \log n\right)}$
  - **Caso 3: se a < b^k, então T(n) =** $\boldsymbol{\Theta\left(n^k\right)}$

Nota: Essa forma **só funciona nesse caso específico** em que $\boldsymbol{f(n) = n^k}$. Mas **não é o Teorema Mestre completo**, apenas um **caso particular**.

Critério de comparação é entre $\boldsymbol{f(n)}$ e $n^{\log_b a}$

Caso	**Condição sobre f(n)**	**Solução T(n)**	**Situação**
1	**$f(n) = O(n^{\log_b a - \varepsilon})$**	**$\Theta(n^{\log_b a})$**	**Recursão domina**
2	**$f(n) = \Theta(n^{\log_b a} . \log^k n)$**	**$\Theta(n^{\log_b a} . \log^{k+1} n)$**	**Empate**
3	**$f(n) = \Omega(n^{\log_b a + \varepsilon})$, reg.**	**$\Theta(f(n))$**	**Termo externo domina**

Exercício 1 – Aplicação direta do Teorema Mestre

Resolva a recorrência:

$$T(n) = 2T(\frac{n}{2}) + n$$

Exercício 2 – Comparação com caso limite

Determine a complexidade assintótica de:

$$T(n) = 2T(\frac{n}{2}) + n^2$$

Exercício 3 – Caso logarítmico

Resolva a recorrência:

$$T(n) = 2T(\frac{n}{2}) + n\log\, n$$

Exercício 4 – Custo constante adicional

Resolva a recorrência:

$$T(n) = 3T(\frac{n}{2}) + n$$

Exercício 5 – Divisão desigual

Determine a complexidade de:

$$T(n) = T(\frac{2n}{3}) + 1$$

Exercício 6 – Recorrência sem Teorema Mestre direto (não se encaixa)

Resolva (ou explique por que não se aplica o Teorema Mestre):

$$T(n) = T(n-1) + 1$$

Exercício 7 – Análise por árvore de recursão

Construa a **árvore de recursão** e determine a complexidade assintótica de:

$$T(n) = T(\frac{n}{2}) + n$$

Exercício 8 – Custo polinomial e múltiplas subchamadas

Determine $T(n)$para:

$$T(n) = 4T(\frac{n}{2}) + n^2$$

Exercício 9 – Recorrência dominada pelo custo externo

Resolva:

$$T(n) = 8T(\frac{n}{2}) + n^3$$

Exercício 10 – Desafio

Classifique a complexidade (**sem necessariamente resolver completamente**) de:

$$T(n) = 7T(\frac{n}{3}) + n^2$$

GABARITO / SOLUÇÕES

Exercício	Resolução / Resultado
1	$a = 2, b = 2, f(n) = n$. $n^{\log_b a} = n^{\log_2 2} = n$. $f(n) = \Theta(n)$⇒ **Caso 2** → $T(n) = \Theta(n\log\, n)$.
2	$a = 2, b = 2, f(n) = n^2$. $n^{\log_2 2} = n$. Como $f(n) = \Omega(n^{1+\epsilon})$com $\epsilon = 1$⇒ **Caso 3** → $T(n) = \Theta(n^2)$.
3	$a = 2, b = 2, f(n) = n\log\, n$. $n^{\log_2 2} = n$. $f(n) = \Theta(n\log\, n)$⇒ **limite do Caso 2** → $T(n) = \Theta(n\log^2 n)$.
4	$a = 3, b = 2, f(n) = n$. $n^{\log_2 3} \approx n^{1.585}$. Como $f(n)$cresce mais lento ⇒ **Caso 1** → $T(n) = \Theta(n^{\log_2 3})$.
5	Forma não padrão (não divide igualmente), mas o custo decresce geometricamente. Aproximação: $T(n) = O(\log\, n)$.
6	Não se aplica o Teorema Mestre (divisão não é por constante). Solução direta: $T(n) = O(n)$.
7	Árvore de recursão: Nível 0: n Nível 1: n/2 Nível 2: n/4 ... até $\log_2 n$ níveis. Soma total: $n + n/2 + n/4 + \ldots \approx 2n = O(n)$+ log n níveis → **O(n)**.
8	$a = 4, b = 2, f(n) = n^2$. $n^{\log_2 4} = n^2$. $f(n) = \Theta(n^2)$⇒ **Caso 2** → $T(n) = \Theta(n^2\log\, n)$.
9	$a = 8, b = 2, f(n) = n^3$. $n^{\log_2 8} = n^3$. Igual ⇒ **Caso 2** → $T(n) = \Theta(n^3\log\, n)$.

Exercício Resolução / Resultado	
10	$a = 7, b = 3, f(n) = n^2$. $n^{\log_3 7} \approx n^{1.77}$. Como $f(n)$cresce mais → **Caso 3** → $T(n) = \Theta(n^2)$.

Observações para o aluno

- O Teorema Mestre **não se aplica** quando:
  - ✓ o tamanho do subproblema **não é fração constante de n** (ex: $T(n-1)$);
  - ✓ ou quando **a divisão não é uniforme**.
- Sempre compare o crescimento de $f(n)$com $n^{\log_b a}$para determinar qual caso usar:

1. $f(n)$**menor** → **Caso 1**
2. $f(n)$**igual** → **Caso 2**
3. $f(n)$**maior** → **Caso 3**