Análise de Algoritmos: O Método Mestre

Na postagem passada, foi apresentado o conceito de **relação de recorrência**. Esse conceito é fundamental para que possamos realizar a análise de algoritmos recursivos. Nesta postagem, veremos o método mais simples para a análise de relações de recorrência: o **Método Mestre** (ou Teorema Mestre). Utilizá-lo deveria ser sempre a sua **primeira opção**, devido à sua simplicidade.

O Método Mestre só pode ser aplicado para recorrência da seguinte forma:

$$T(n) = aT(n/b) + \underbrace{f(n)}_{\text{função assintoticamente positiva}} \qquad \begin{array}{l} a \geq 1 \\ b \geq 1 \end{array}$$

O método clássico, como apresentado no livro Algoritmos: Teoria e Prática, é apresentado abaixo:

1. Se $f(n) = O(n^{\log_b a - \epsilon})$ para alguma constante $\epsilon > 0$, então $T(n) = \Theta(n^{\log_b a})$.
2. Se $f(n) = \Theta(n^{\log_b a})$, então $T(n) = \Theta(n^{\log_b a} \lg n)$.
3. Se $f(n) = \Omega(n^{\log_b a + \epsilon})$ para alguma constante $\epsilon > 0$, e se $af(n/b) \leq cf(n)$ para alguma constante $c < 1$ e para todo n suficientemente grande, então $T(n) = \Theta(f(n))$. ■

Uma versão simplificada do Método Mestre é possível para recorrências das forma abaixo:

$$T(n) = aT(n/b) + \Theta(n^d) \qquad \begin{array}{l} a \geq 1 \\ b \geq 1 \\ d \geq 0 \end{array}$$

Com isso, temos as seguintes regras:

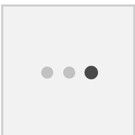
1. Se $a=b^d$, então $T(n)=\Theta(n^d \cdot \log n)$
2. Se $a<b^d$, então $T(n)=\Theta(n^d)$
3. Se $a>b^d$, então $T(n)=\Theta(n^{\log_a b})$

Vamos utilizar a versão simplificada para analisar a **busca binária** e o **Merge Sort**.
A relação de recorrência da **busca binária** é:

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

Dado isso, temos que **a = 1**, **b = 2**, e **d = 0**. Então, temos que **b^d = 2^0**, resultando em 1. Com isso, entramos no caso 1, e T(n) = **θ**(n^0*logn), resultando em **T(n) = θ(logn)**, como esperado.

A relação de recorrência do Merge Sort é:



Dado isso, temos que **a = 2**, **b = 2**, e **d = 1**. Então, temos que **b^d = 2^1**, resultando em 2. Com isso, entramos no caso 1, e T(n) = **θ**(n^1*logn), resultando em **T(n) = θ(nlogn)**, como esperado.

Apesar de sua simplicidade, infelizmente, o Método Mestre só pode ser utilizado para algoritmos do paradigma **Divisão-e-Conquista** (tema de outra série de postagens) e apenas para os casos de os subproblemas terem o mesmo tamanho. A sua generalização é denominada Método Akra-Bazzi.

Sumário

Nesta postagem, você aprendeu sobre:

- Utilização do **Método Mestre** para realizar a análise de algoritmos de recursão do paradigma Divisão-e-Conquista com subproblemas do mesmo tamanho.
- O Método Mestre clássico, apresentado no livro Algoritmos: Teoria e Prática; e o Método Mestre simplificado.