Recorrências e soluções de recorrências 2

Exercício 1 — Método Mestre

1) $T(n) = 16T(n/2) + n^2$
Temos $a = 16$ e $b = 2$, logo:

$$n^{\log_b a} = n^{\log_2 16} = n^4.$$

Além disso, $f(n) = n^2 = O(n^{4-\varepsilon})$, com $\varepsilon = 2$.
Pelo caso 1 do Teorema Mestre:

$$T(n) = \Theta(n^4).$$

—

2) $T(n) = 25T(n/5) + n$
Temos $a = 25$ e $b = 5$, logo:

$$n^{\log_5 25} = n^2.$$

Além disso, $f(n) = n = O(n^{2-\varepsilon})$.
Pelo caso 1:

$$T(n) = \Theta(n^2).$$

—

3) $T(n) = 5T(n/5) + n$
Temos $a = 5$ e $b = 5$, logo:

$$n^{\log_5 5} = n.$$

Além disso, $f(n) = n = \Theta(n^{\log_b a})$.
Pelo caso 2:

$$T(n) = \Theta(n \log n).$$

—

4) $T(n) = 9T(n/3) + n^2$
Temos $a = 9$ e $b = 3$, logo:

$$n^{\log_3 9} = n^2.$$

Aqui $f(n) = n^2 = \Theta(n^{\log_b a})$.
Pelo caso 2:

$$T(n) = \Theta(n^2 \log n).$$

—
5) $T(n) = 4T(n/2) + 1$
Temos $a = 4$ e $b = 2$, logo:

$$n^{\log_2 4} = n^2.$$

Além disso, $f(n) = 1 = O(n^{2-\varepsilon})$.
Pelo caso 1:

$$T(n) = \Theta(n^2).$$

—
6) $T(n) = 5T(n/3) + n^3$
Temos $a = 5$ e $b = 3$, logo:

$$n^{\log_3 5} \approx n^{1.4649}.$$

Além disso, $f(n) = n^3 = \Omega(n^{\log_b a+\varepsilon})$.
Verifiquemos a condição de regularidade:

$$af(n/b) = 5\left(\frac{n}{3}\right)^3 = \frac{5}{27}n^3 \leq cn^3,$$

com $c = \frac{5}{27} < 1$.
Pelo caso 3:

$$T(n) = \Theta(n^3).$$

Exercício 2 — Método de Expansão

1) $T(n) = T(n/3) + 1$
Expandindo:

$$T(n) = T(n/3) + 1 = T(n/3^2) + 2 = \cdots = T(n/3^k) + k.$$

Com $n/3^k = \Theta(1) \Rightarrow k = \Theta(\log n)$.
Portanto:

$$T(n) = \Theta(\log n).$$

—
2) $T(n) = 3T(n/3) + n$
No nível i há 3^i subproblemas de tamanho $n/3^i$.
O custo de cada nível é:

$$3^i \cdot \frac{n}{3^i} = n.$$

Número de níveis: $\log_3 n$.
Portanto:

$$T(n) = n \log n.$$

Exercício 3 — Método da Árvore de Recursão

1) $T(n) = 3T(n/3) + n$

Cada chamada gera 3 subproblemas de tamanho $n/3$. No nível i, há 3^i subproblemas, cada um de custo $n/3^i$. O custo total do nível é:

$$3^i \cdot \frac{n}{3^i} = n.$$

A altura da árvore é $\log_3 n$.

Portanto:

$$T(n) = n \log_3 n + O(n) = \Theta(n \log n).$$

—
2) $T(n) = 4T(n/4) + n$

Cada chamada gera 4 subproblemas de tamanho $n/4$. No nível i, há 4^i subproblemas, cada um de custo $n/4^i$. O custo total do nível é:

$$4^i \cdot \frac{n}{4^i} = n.$$

A altura da árvore é $\log_4 n$.

Portanto:

$$T(n) = n \log_4 n + O(n) = \Theta(n \log n).$$