Recorrência do Tempo do Merge-Sort

Seja $T(n)$ o tempo do **Merge-Sort** para um vetor de tamanho n.

$$T(n) \;=\; T\left(\left\lceil \frac{n}{2} \right\rceil\right) \;+\; T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) \;+\; \Theta(n)$$

Para obter a ordem assintótica de $T(n)$, podemos aproximar para

$$T(n) \;=\; 2 \cdot T\left(\frac{n}{2}\right) \;+\; n.$$

Por exemplo, tomando n como potência de 2 ($n = 2^k$).

Recorrências comuns em Divisão e Conquista

Inteiros positivos a e b e função positiva $f(n)$.

Algoritmo-DC(n)

se $(n \leq 1)$ **então retorne**
para $i \leftarrow 1$ **até** a **faça**:
    Algoritmo-DC($\lfloor n/b \rfloor$)
para $j \leftarrow 1$ **até** $f(n)$ **faça**:
    imprime "*"

Tempo de Algoritmo-DC(n)

$$T(n) \; = \; a \cdot T\left(\left\lfloor \frac{n}{b} \right\rfloor\right) \; + \; f(n)$$

Recorrências comuns em Divisão e Conquista

Inteiros $a \geq 1$, $b > 1$ e função positiva $f(n)$. $n = b^k$ (potência de b).

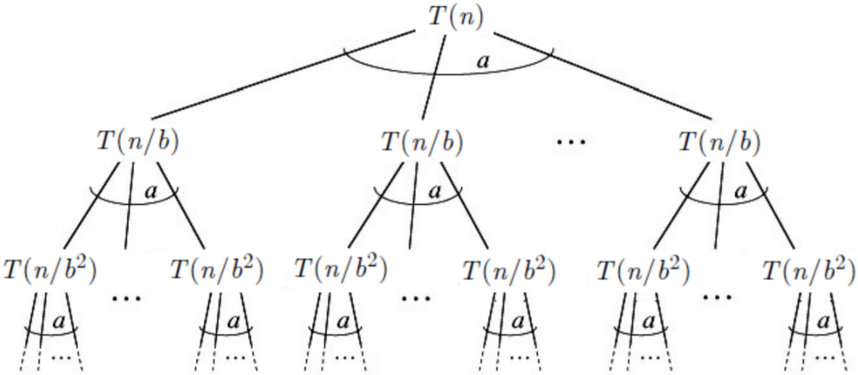
Algoritmo-DC(n)

```
se (n ≤ 1) então retorne
para i ← 1 até a faça:
      Algoritmo-DC(⌊n/b⌋)
para j ← 1 até f(n) faça:
      imprime "*"
```

Tempo de Algoritmo-DC(n)

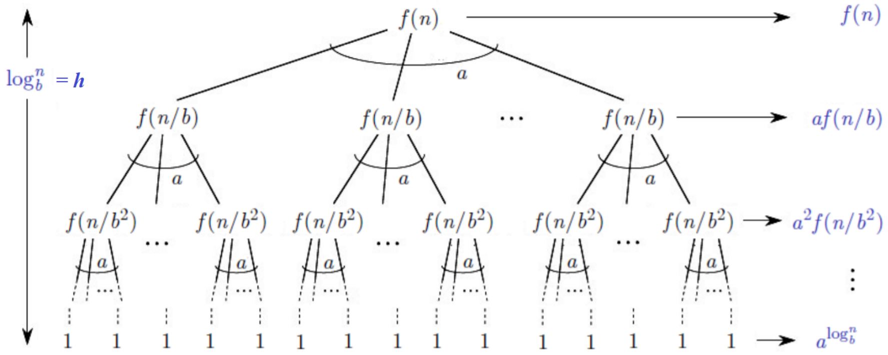
$$T(n) \; = \; a \cdot T\left(\frac{n}{b}\right) \; + \; f(n)$$

Árvore de recursão: $T(n) \; = \; a \cdot T(n/b) \; + \; f(n)$



Altura da árvore: Folha $n/b^h \; = \; 1 \; \Longrightarrow h = \log_b n$
Níveis de recursão: 0 a $h-1$. Nível das folhas: h.

Árvore de recursão: $T(n) = a \cdot T(n/b) + f(n)$



$$T(n) = a^h + \sum_{i=0}^{h-1} a^i \cdot f(n/b^i)$$

Resolução de recorrências: $T(n) = a \cdot T(n/b) + n^c$

Em geral: $f(n) = n^c$ para alguma constante c fixa.

$$T(n) = a^h + \sum_{i=0}^{h-1} a^i \cdot f(n/b^i)$$

$$T(n) = a^{\log_b n} + \sum_{i=0}^{h-1} a^i \cdot \left(\frac{n}{b^i}\right)^c$$

$$T(n) = n^{\log_b a} + \sum_{i=0}^{h-1} \left(\frac{a}{b^c}\right)^i \cdot n^c$$

$$T(n) = n^{\log_b a} + n^c \cdot \sum_{i=0}^{h-1} \left(\frac{a}{b^c}\right)^i$$

Resolução de recorrências: $T(n) = a \cdot T(n/b) \ + \ n^c$

$$T(n) \ = \ n^{\log_b a} \ + \ n^c \ \cdot \ \sum_{i=0}^{h-1} \left(\frac{a}{b^c}\right)^i, \ com\ b > 1\ e\ h = \log_b n$$

- Se $\log_b a \ = \ c \ \Rightarrow a = b^c \ \Rightarrow$ P.G. constante:

$$T(n) \ = \ n^{\log_b a} \ + \ (\log_b n) \cdot n^c \ \Rightarrow \ T(n) = \Theta(n^c \cdot \log n)$$

- Se $\log_b a \ < \ c \ \Rightarrow a < b^c \ \Rightarrow$ P.G. decrescente:

$$T(n) \leq n^{\log_b a} + n^c \cdot \sum_{i=0}^{\infty} \left(\frac{a}{b^c}\right)^i \leq n^c + n^c \cdot \left(\frac{1}{1 - a/b^c}\right) \ \Rightarrow \ T(n) = \Theta(n^c)$$

- Se $\log_b a \ > \ c \ \Rightarrow a > b^c \ \Rightarrow$ P.G. crescente:

$$T(n) \ = \ \ldots \text{(muitas continhas, fica pro próximo slide)}$$

Resolução de recorrências: $T(n) = a \cdot T(n/b) \; + \; n^c$

$$T(n) \;=\; n^{\log_b a} \;+\; n^c \;\cdot\; \sum_{i=0}^{h-1} \left(\frac{a}{b^c}\right)^i, \; com \; a > b^c \; e \; h = \log_b n$$

$$T(n) = n^{\log_b a} + n^c \cdot \frac{(a/b^c)^h - 1}{(a/b^c) - 1} \;=\; n^{\log_b a} + \frac{b^c \cdot n^c}{a - b^c} \cdot \left(\left(\frac{a}{b^c}\right)^h - 1\right)$$

$$T(n) = \; n^{\log_b a} + \frac{b^c \cdot \cancel{n^c}}{a - b^c} \cdot \left(\frac{a^h - b^{hc}}{\cancel{b^{hc}}}\right) \;=\; n^{\log_b a} + \frac{b^c}{a - b^c} \cdot \left(n^{\log_b a} - n^c\right)$$

$$T(n) = \; n^{\log_b a} \cdot \left(\frac{a}{a - b^c}\right) - n^c \left(\frac{b^c}{a - b^c}\right) \;\Rightarrow\; T(n) = \Theta(n^{\log_b a})$$

Resolução de recorrências: $T(n) = a \cdot T(n/b) \; + \; n^c$

TEOREMA MESTRE para Recorrências comuns de Divisão e Conquista

$$T(n) \;=\; n^{\log_b a} \;+\; n^c \;\cdot\; \sum_{i=0}^{h-1} \left(\frac{a}{b^c}\right)^i, \; com \; b > 1 \; e \; h = \log_b n$$

- Se $\log_b a \;=\; c \;\Rightarrow\; a = b^c \;\Rightarrow$ P.G. constante:

$$T(n) \;=\; n^{\log_b a} \;+\; (\log_b n) \cdot n^c \;\Rightarrow\; T(n) = \Theta(n^c \cdot \log n)$$

- Se $\log_b a \;<\; c \;\Rightarrow\; a < b^c \;\Rightarrow$ P.G. decrescente:

$$T(n) \leq n^{\log_b a} + n^c \cdot \sum_{i=0}^{\infty} \left(\frac{a}{b^c}\right)^i \leq n^c + n^c \cdot \left(\frac{1}{1 - a/b^c}\right) \;\Rightarrow\; T(n) = \Theta(n^c)$$

- Se $\log_b a \;>\; c \;\Rightarrow\; a > b^c \;\Rightarrow$ P.G. crescente:

$$T(n) = \; n^{\log_b a} \cdot \left(\frac{a}{a - b^c}\right) - n^c \left(\frac{b^c}{a - b^c}\right) \;\Rightarrow\; T(n) = \Theta(n^{\log_b a})$$

Voltando ao Tempo do Merge-Sort

Seja $T(n)$ o tempo do Merge-Sort para um vetor de tamanho n.

$$T(n) \;=\; T\left(\left\lceil \frac{n}{2} \right\rceil\right) \;+\; T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) \;+\; \Theta(n)$$

Para obter a ordem assintótica de $T(n)$, podemos aproximar para

$$T(n) \;=\; 2 \cdot T\left(\frac{n}{2}\right) \;+\; n.$$

Por exemplo, tomando n como potência de 2 ($n = 2^k$).

Teorema Mestre ou Método Mestre:

$a = 2,\ b = 2$ e $c = 1 \Rightarrow \log_b a = 1 = c \Rightarrow$

$\Rightarrow\ T(n) = \Theta(n^c \cdot \log n) \Rightarrow$

$\Rightarrow\ T(n) = \Theta(n \cdot \log n)$

Voltando ao Tempo do Merge-Sort

Seja $T(n)$ o tempo do Merge-Sort para um vetor de tamanho n.

$$T(n) \;=\; T\left(\left\lceil \frac{n}{2} \right\rceil\right) \;+\; T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) \;+\; \Theta(n)$$

$\Rightarrow\;\; T(n) = \Theta(n \cdot \log n)$

Esse é o tempo de pior caso. O tempo de melhor caso não muda quase nada, pois o **Merge-Sort** segue a recursão indiferentemente dos valores do vetor.

Logo, o **Merge-Sort** tem tempo de pior caso e de melhor caso $\Theta(n \log n)$.

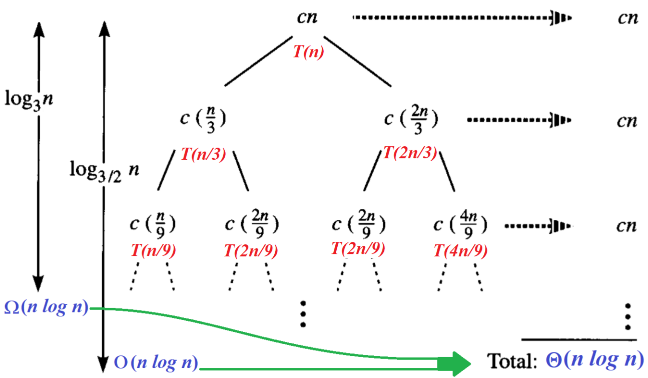
Ao contrário do **Insertion-Sort**, que tem tempo de pior caso $\Theta(n^2)$ e de melhor caso $\Theta(n)$.

Método Geral da Árvore de Recursão

Método geral para Resolução de Recorrências: expandir a recursão.
Exemplo:

$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + \Theta(n).$$

O termo $\Theta(n)$ significa alguma função $f(n) = \Theta(n)$ e pode ser substituído por $c \cdot n$, para alguma constante $c > 0$.



Método da Árvore de Recursão

Método geral para Resolução de Recorrências: expandir a recursão.
Exemplo:

$$T(n) = T\left(\frac{n}{10}\right) + T\left(\frac{9n}{10}\right) + \Theta(n).$$

O termo $\Theta(n)$ significa alguma função $f(n) = \Theta(n)$ e pode ser substituído por $c \cdot n$, para alguma constante $c > 0$.