

Merge Sort (A, p, r)

Se $p < r$

$$q \leftarrow \left\lfloor \frac{p+r}{2} \right\rfloor$$

Merge Sort (A, p, q)

Merge Sort ($A, q+1, r$)

Intercala (A, p, q, r)

Como descobrir o tempo a partir das instruções do algoritmo?

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

Algoritmos recursivos:

De um modo geral, seu tempo de execução é adunado em função da própria tempo para entradas menores

↳ Relação de recorrência

* Recorrência de merge % é suficiente para adunar o tempo?

* O que a recorrência diz sobre o tempo?

Redução de relação de recorrência

Relação de recorrência:

Equação (ou desigualdade) que descreve uma função em termos de valores menores de seus inputs.

Ex Fibonacci:

$$F(0) = 1$$

$$F(1) = 1$$

$$F(k) = F(k-1) + F(k-2)$$

Fatorial: $F(1) = 1$
 $F(k) = k \cdot F(k-1)$

Resolver uma relação de recorrência:

- Determinar a função correspondente a ela, dependendo apenas do valor de n .

↳ Pode ser idemboolo exatamente em alguns casos:

$$\text{Ex: } f(0) = 1$$

$$f(k) = 3^k$$

$$f(k) = 3 \cdot f(k-1), \forall k > 0$$

base:

$$f(0) = 3^0 = 1$$

passo:

$$f(k) = 3 \cdot 3^{k-1} = 3^k$$

hip. $< k$

↳ Para algoritmos, idemboolos os limites assintóticos

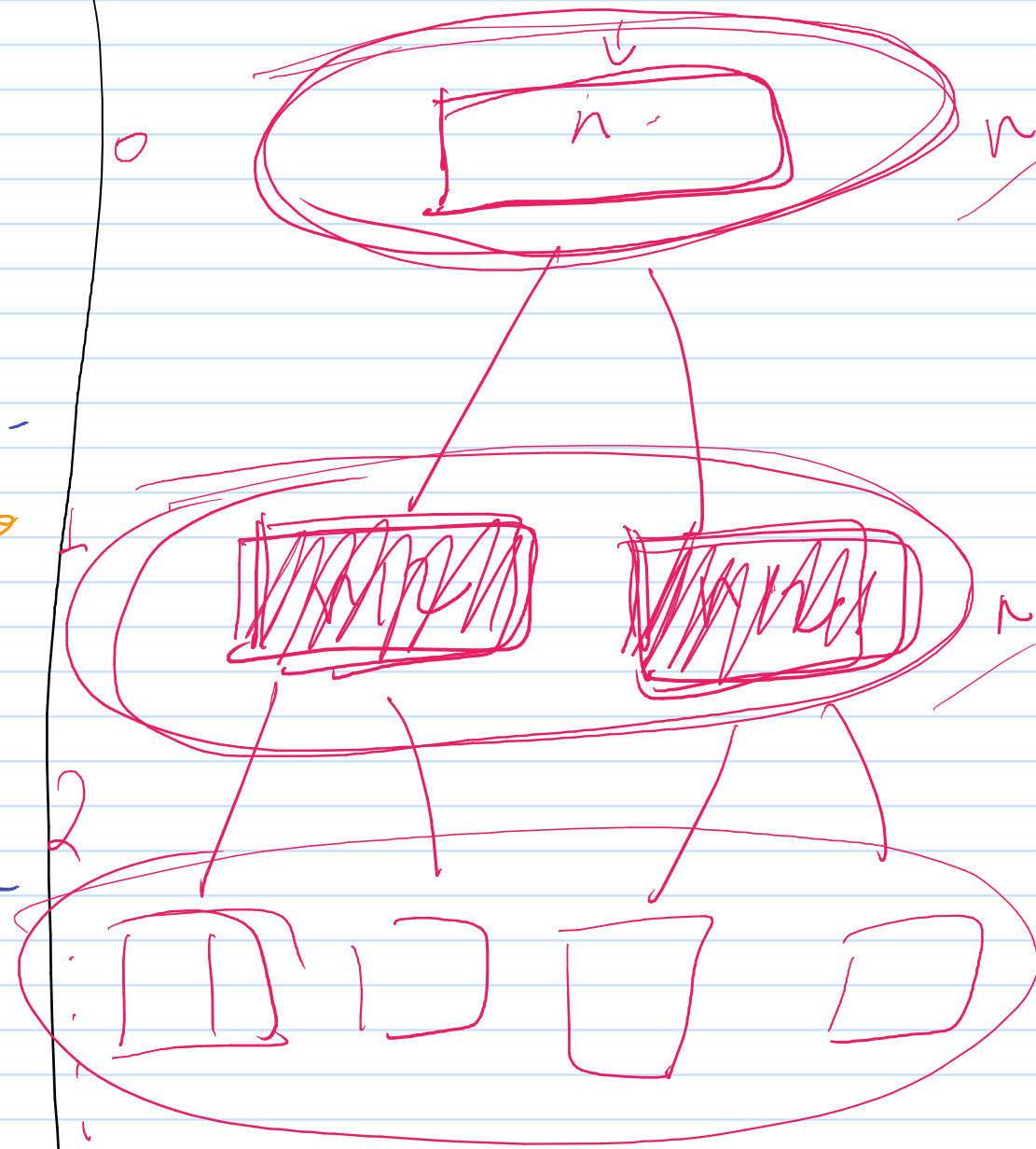
Relações de recorrência: Como resolver?

1) Método da árvore de recorrência

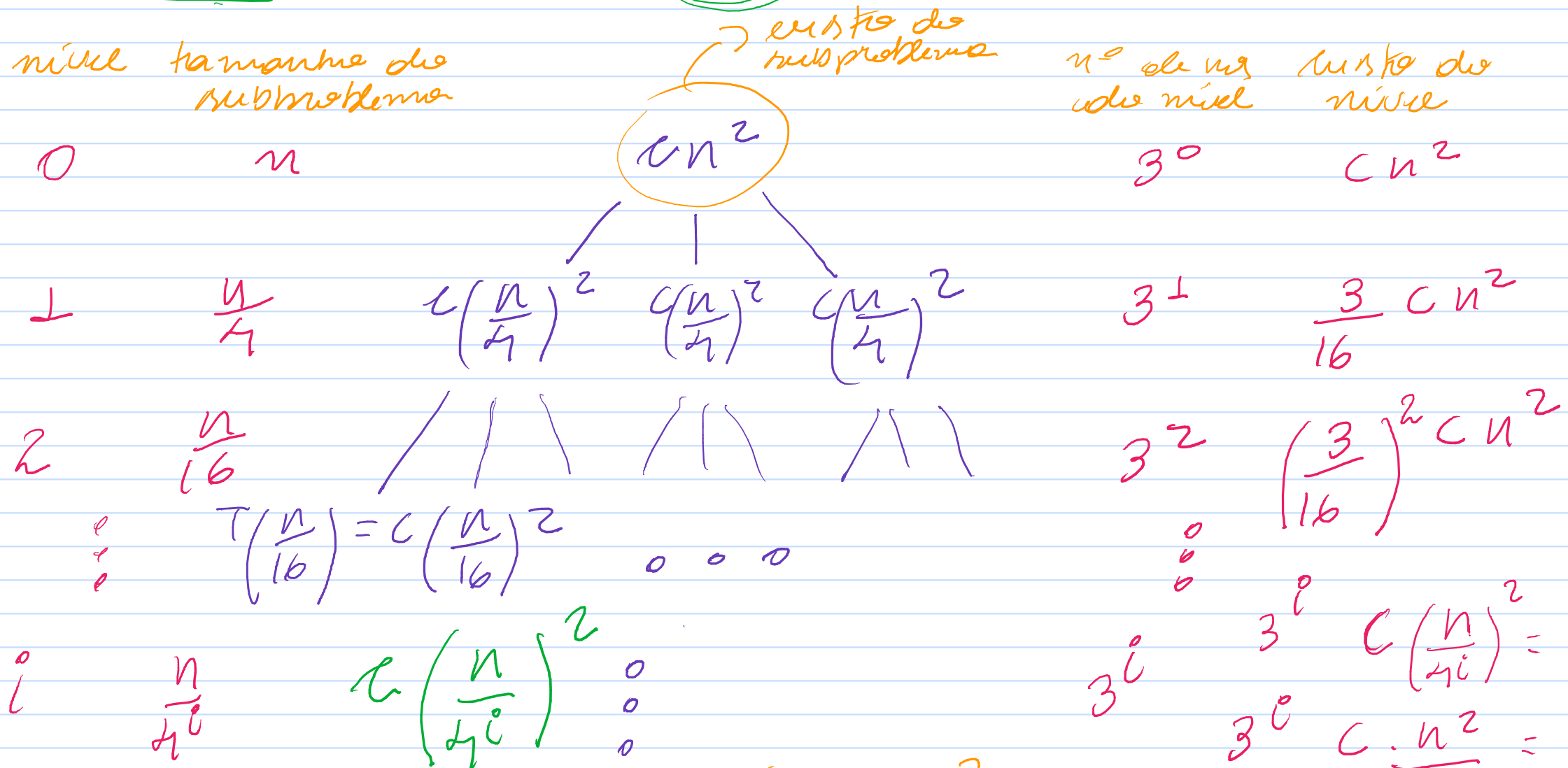
↳ Cada nó representa o custo de um subproblema (valor tamanho do subproblema no nível i)

↳ Somando o custo de cada nó em um nível, obtemos o custo do nível (valor o número de nós em cada nível)

↳ Somando o custo de cada nível, determinamos o custo total (valor a altura da árvore)



$E_n: T(n) = 3T(n/4) + \Theta(n^2)$ (Encontre limite superior)
 $T(n) = 3T(n/4) + cn^2$



Altura da árvore?

$\frac{n}{4^i} = 1 \Rightarrow n = 4^i \Rightarrow \log_4 n = i$
 $3^i \cdot \frac{cn^2}{(4^i)^2} = \frac{3^i}{16^i} cn^2 = \left(\frac{3}{16}\right)^i cn^2$

Ans to total:

$$T(n) = cn^2 + \frac{3}{16}cn^2 + \left(\frac{3}{16}\right)^2 cn^2 + \dots + \left(\frac{3}{16}\right)^{\log_4 n} cn^2$$

$$= cn^2 \sum_{i=0}^{\log_4 n} \left(\frac{3}{16}\right)^i$$

$$< cn^2 \sum_{i=0}^{\infty} \left(\frac{3}{16}\right)^i$$

$$= cn^2 \cdot \frac{1}{1 - (3/16)}$$

$$= \frac{16}{13} cn^2$$

$$= O(n^2)$$

2) Resolução iterativa

Ex: Busca binária

$$T(n) = T(n/2) + 1$$

$$= T(n/4) + 1 + 1 = T\left(\frac{n}{2^2}\right) + 2$$

$$= T(n/8) + 1 + 1 + 1 = T\left(\frac{n}{2^3}\right) + 3$$

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

$$= 1 + \log n = \Theta(\log n)$$

$$\frac{n}{2^i} = 1 \Rightarrow i = \log n$$

$$T(1) = 0$$
$$Ex: T(n) = T(n-1) + \frac{1}{n}$$

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

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

$$= H_n = \ln n + O(1)$$

↳ mínima
harmônica.

3) Método da Substituição

↳ Indução

Ex: Merge

$$T(1) = \Theta(1)$$

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

$$\boxed{T(n) = O(n \log n) \leq C n \log n}$$

base: $n=2$

$$\begin{aligned} T(2) &= 2 \cdot T(1) + 2 = 4 \\ &\leq C \cdot 2 \log 2, \\ C &\geq 2. \end{aligned}$$

hip.: $T(n) \leq C n \log n$
 $\forall n \leq K$

passo: $n=K$

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

$$\begin{aligned} &\leq 2 \left(C \cdot \frac{K}{2} \cdot \log \frac{K}{2} \right) + K \\ &= CK \left(\log K - \log 2 \right) + K \end{aligned}$$

$$= CK \log K - CK + K$$

$$\leq CK \log K \quad \underbrace{-CK + K}_{< 0}$$

Ex 2: Fibonacci

$$T(0) = 1$$

$$T(1) = 1$$

$$T(n) = T(n-1) + T(n-2)$$

$$\boxed{\begin{aligned} T(n) &= O(2^n) \\ &\leq C \cdot 2^n \end{aligned}}$$

base: $n = 2$

$$T(2) = 2$$

$$\leq C \cdot 2^2$$

hipótese: $T(n) \leq C \cdot 2^n$

$$n < k$$

passo: $n = k$

$$T(k) = T(k-1) + T(k-2)$$

$$\leq C \cdot 2^{k-1} + C \cdot 2^{k-2}$$

$$\leq C \cdot 2^{k-1} + C \cdot 2^{k-1}$$

$$= 2 \cdot C \cdot 2^{k-1}$$

$$= C \cdot 2^k$$

4) Teorema Mestre:

Seja $a \geq 1$, $b \geq 1$ constantes e $f(n)$ uma função. Seja

$$T(n) = a T(n/b) + f(n)$$

Então:

1. Se $f(n) = O(n^{\log_b a - \epsilon})$, $T(n) = \Theta(n^{\log_b a})$

2. Se $f(n) = \Theta(n^{\log_b a})$, $T(n) = \Theta(n^{\log_b a} \log n)$

3. Se $f(n) = \Omega(n^{\log_b a + \epsilon})$ e $a f(n/b) \leq c f(n)$,

$$T(n) = \Theta(f(n))$$