Fontes principais

1. Cormem T. H.; Leiserson C. E.; Rivest R.: Stein C.. *Introduction to Algorithms*, 3^a edição, MIT Press, 2009
2. Análise de algoritmo - IME/USP (prof. Paulo Feofiloff) `http://www.ime.usp.br/~pf/analise_de_algoritmos`

Notação “Chão” e “Teto”

Notação "Chão" e "Teto"

Definição 1. *O **chão** (ou piso) de um número não negativo x é o único natural i tal que $i \leq x < i + 1$. O chão de x é denotado por $\lfloor x \rfloor$.*

Definição 2. *O **teto** de x é o único número j natural tal que que $j - 1 < x \leq j$. O teto de x é denotado por $\lceil x \rceil$*

Propriedade: $\lceil \frac{x}{2} \rceil + \lfloor \frac{x}{2} \rfloor = x$

Divisão e conquista

Divisão e conquista

Dada uma instância do problema, dividí-lo em instâncias menores, resolvê-los recursivamente e, a partir de suas soluções, obter a solução da instância original.

Ordenação por intercalação

Ordenação por intercalação

Divide-se a sequência em duas partes, ordena-se cada uma delas recursivamente e intercala-se as duas sequências ordenadas.

Ordenação por intercalação

Merge-Sort(A, p, r)	custo

```
se p < r
    então q ← ⌊(p+r)/2⌋
        Merge-Sort(A, p, q)
        Merge-Sort(A, q+1, r)
        Merge(A, p, q, r)
```

Ordenação por intercalação

Merge-Sort(A, p, r)	custo
se $p < r$	$\Theta(1)$
então $q \leftarrow \lfloor \frac{p+r}{2} \rfloor$	$\Theta(1)$
Merge-Sort(A, p, q)	$T(\lceil \frac{n}{2} \rceil)$
Merge-Sort($A, q+1, r$)	$T(\lfloor \frac{n}{2} \rfloor)$
Merge(A, p, q, r)	$\Theta(n)$

Ordenação por intercalação

$$T(n) = \begin{cases} \Theta(1) & \text{, se } n = 1 \\ T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + \Theta(n) + \Theta(2) & \text{, se } n > 1 \end{cases}$$

Resolver recorrências é importante para obter fórmulas fechadas para a complexidade de algoritmos recursivos.

Métodos para resolver a recorrência

Métodos para resolver a recorrência

- **Substituição**: Arriscamos um "palpite" para um limite e então usamos indução matemática para provar que nosso "palpite" estava certo.

- **Árvore de recursão**: Convertemos recorrência em uma árvore e usamos a técnica para limitar somatórios com intuito de resolver a recorrência.

- **Método mestre**: Para recorrências da forma $T(n) = aT(n/b) + f(n)$, com $a \geq 1$ e $b > 1$.

Lembre-se

A meta não é determinar as soluções exatas de uma recorrência, mas encontrar uma solução $g(n)$ tal que $T(n) = \Theta(g(n))$

Método da substituição

Método da substituição

Vamos resolver a recorrência que expressa o tempo de execução do algoritmo Merge-Sort. Primeiro vamos simplificar a recorrência:

$$T(n) = \begin{cases} 1 & \text{, se } n = 1 \\ T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + n & \text{, se } n > 1 \end{cases}$$

Método da substituição

$$T(n) = \begin{cases} 1 & \text{, se } n = 1 \\ T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + n & \text{, se } n > 1 \end{cases}$$

Palpite: $T(n) = O(n \cdot \log n)$. Ou seja, $T(n) \leq c \cdot n \cdot \log n$, para alguma constante c quando $n \geq n_0$.

Método da substituição

$$T(n) = \begin{cases} 1 & \text{, se } n = 1 \\ T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + n & \text{, se } n > 1 \end{cases}$$

Prova por indução

Base: $T(2) = T(1) + T(1) + 2 = 4 \leq 3 \cdot 2 \cdot \log 2 = 3 \cdot 2 \cdot 1 = 6$.

Método da substituição

$$T(n) = \begin{cases} 1 & \text{, se } n = 1 \\ T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + n & \text{, se } n > 1 \end{cases}$$

Prova por indução

Base: $T(2) = T(1) + T(1) + 2 = 4 \leq 3 \cdot 2 \cdot \log 2 = 3 \cdot 2 \cdot 1 = 6$.

Hipótese: $T(n) \leq c \cdot n \cdot \log n$, para todo $m < n$

Método da substituição

$$T(n) = \begin{cases} 1 & \text{, se } n = 1 \\ T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + n & \text{, se } n > 1 \end{cases}$$

Prova por indução

Base: $T(2) = T(1) + T(1) + 2 = 4 \leq 3 \cdot 2 \cdot \log 2 = 3 \cdot 2 \cdot 1 = 6$.

Hipótese: $T(n) \leq c \cdot n \cdot \log n$, para todo $m < n$

Passo indutivo: $T(n) = T(\underbrace{\lceil \frac{n}{2} \rceil}_{<n}) + T(\underbrace{\lfloor \frac{n}{2} \rfloor}_{<n}) + n$

Método da substituição

Passo indutivo: $T(n) = T(\underbrace{\lceil \frac{n}{2} \rceil}_{<n}) + T(\underbrace{\lfloor \frac{n}{2} \rfloor}_{<n}) + n$

$\leq c\lceil \frac{n}{2} \rceil \log\lceil \frac{n}{2} \rceil + c\lfloor \frac{n}{2} \rfloor \log\lfloor \frac{n}{2} \rfloor + n$

Método da substituição

Passo indutivo: $T(n) = T(\underbrace{\lceil \frac{n}{2} \rceil}_{<n}) + T(\underbrace{\lfloor \frac{n}{2} \rfloor}_{<n}) + n$

$$\leq c\lceil \tfrac{n}{2} \rceil \log \lceil \tfrac{n}{2} \rceil + c\lfloor \tfrac{n}{2} \rfloor \log \lfloor \tfrac{n}{2} \rfloor + n$$
$$\leq c\lceil \tfrac{n}{2} \rceil \log n + c\lfloor \tfrac{n}{2} \rfloor (\log n - \log 2) + n$$

Método da substituição

Passo indutivo: $T(n) = T(\underbrace{\lceil \frac{n}{2} \rceil}_{<n}) + T(\underbrace{\lfloor \frac{n}{2} \rfloor}_{<n}) + n$

$$
\begin{aligned}
&\le c\lceil \tfrac{n}{2} \rceil \log \lceil \tfrac{n}{2} \rceil + c\lfloor \tfrac{n}{2} \rfloor \log \lfloor \tfrac{n}{2} \rfloor + n \\
&\le c\lceil \tfrac{n}{2} \rceil \log n + c\lfloor \tfrac{n}{2} \rfloor (\log n - \log 2) + n \\
&\le c\lceil \tfrac{n}{2} \rceil \log n + c\lfloor \tfrac{n}{2} \rfloor (\log n - 1) + n
\end{aligned}
$$

Método da substituição

Passo indutivo: $T(n) = T(\underbrace{\lceil \frac{n}{2} \rceil}_{<n}) + T(\underbrace{\lfloor \frac{n}{2} \rfloor}_{<n}) + n$

$$\begin{aligned}
&\leq c\lceil \tfrac{n}{2} \rceil \log\lceil \tfrac{n}{2} \rceil + c\lfloor \tfrac{n}{2} \rfloor \log\lfloor \tfrac{n}{2} \rfloor + n \\
&\leq c\lceil \tfrac{n}{2} \rceil \log n + c\lfloor \tfrac{n}{2} \rfloor (\log n - \log 2) + n \\
&\leq c\lceil \tfrac{n}{2} \rceil \log n + c\lfloor \tfrac{n}{2} \rfloor (\log n - 1) + n \\
&= c(\lceil \tfrac{n}{2} \rceil + \lfloor \tfrac{n}{2} \rfloor) \log n - c\lfloor \tfrac{n}{2} \rfloor + n
\end{aligned}$$

Método da substituição

Passo indutivo: $T(n) = T(\underbrace{\lceil \frac{n}{2} \rceil}_{<n}) + T(\underbrace{\lfloor \frac{n}{2} \rfloor}_{<n}) + n$

$$\begin{aligned}
&\leq c\lceil \tfrac{n}{2} \rceil \log \lceil \tfrac{n}{2} \rceil + c\lfloor \tfrac{n}{2} \rfloor \log \lfloor \tfrac{n}{2} \rfloor + n \\
&\leq c\lceil \tfrac{n}{2} \rceil \log n + c\lfloor \tfrac{n}{2} \rfloor (\log n - \log 2) + n \\
&\leq c\lceil \tfrac{n}{2} \rceil \log n + c\lfloor \tfrac{n}{2} \rfloor (\log n - 1) + n \\
&= c(\lceil \tfrac{n}{2} \rceil + \lfloor \tfrac{n}{2} \rfloor) \log n - c\lfloor \tfrac{n}{2} \rfloor + n \\
&= cn \log n - c\lfloor \tfrac{n}{2} \rfloor + n
\end{aligned}$$

Método da substituição

Passo indutivo: $T(n) = T(\underbrace{\lceil \frac{n}{2} \rceil}_{<n}) + T(\underbrace{\lfloor \frac{n}{2} \rfloor}_{<n}) + n$

$$\begin{aligned}
&\le c\lceil \tfrac{n}{2} \rceil \log \lceil \tfrac{n}{2} \rceil + c\lfloor \tfrac{n}{2} \rfloor \log \lfloor \tfrac{n}{2} \rfloor + n \\
&\le c\lceil \tfrac{n}{2} \rceil \log n + c\lfloor \tfrac{n}{2} \rfloor (\log n - \log 2) + n \\
&\le c\lceil \tfrac{n}{2} \rceil \log n + c\lfloor \tfrac{n}{2} \rfloor (\log n - 1) + n \\
&= c(\lceil \tfrac{n}{2} \rceil + \lfloor \tfrac{n}{2} \rfloor) \log n - c\lfloor \tfrac{n}{2} \rfloor + n \\
&= cn \log n - c\lfloor \tfrac{n}{2} \rfloor + n \\
&\le cn \log n
\end{aligned}$$

A inequação $n - c\lfloor \frac{n}{2} \rfloor \le cn \log n$ é válida para $c \ge 3$, $n \ge 2$

Resolver a recorrência

$$T(n) = \begin{cases} 1 & \text{, se } n = 1 \\ T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + 1 & \text{, se } n > 1 \end{cases}$$

Resolver a recorrência

$$T(n) = \begin{cases} 1 & \text{, se } n = 1 \\ T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + 1 & \text{, se } n > 1 \end{cases}$$

Palpite: $T(n) = O(n)$

Resolver a recorrência

$$T(n) = \begin{cases} 1 & \text{, se } n = 1 \\ T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + 1 & \text{, se } n > 1 \end{cases}$$

Palpite: $T(n) = O(n)$

Base: Okay! $T(2) = T(1) + T(1) + 1 = 3 = O(2)$

Resolver a recorrência

$$T(n) = \begin{cases} 1 & \text{, se } n = 1 \\ T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + 1 & \text{, se } n > 1 \end{cases}$$

Palpite: $T(n) = O(n)$

Base: Okay! $T(2) = T(1) + T(1) + 1 = 3 = O(2)$

Hipótese: $T(n) \leq cn$, para todo $m < n$.

Resolver a recorrência

$$T(n) = \begin{cases} 1 & \text{, se } n = 1 \\ T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + 1 & \text{, se } n > 1 \end{cases}$$

Palpite: $T(n) = O(n)$

Base: Okay! $T(2) = T(1) + T(1) + 1 = 3 = O(2)$

Hipótese: $T(n) \leq cn$, para todo $m < n$.

Passo indutivo:

$$T(n) = T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + 1$$

Resolver a recorrência

$$T(n) = \begin{cases} 1 & \text{, se } n = 1 \\ T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + 1 & \text{, se } n > 1 \end{cases}$$

Palpite: $T(n) = O(n)$

Base: Okay! $T(2) = T(1) + T(1) + 1 = 3 = O(2)$

Hipótese: $T(n) \leq cn$, para todo $m < n$.

Passo indutivo:

$$T(n) = T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + 1$$

$$\leq c\lceil \frac{n}{2} \rceil + c\lfloor \frac{n}{2} \rfloor + 1$$

Resolver a recorrência

$$T(n) = \begin{cases} 1 & \text{, se } n = 1 \\ T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + 1 & \text{, se } n > 1 \end{cases}$$

Palpite: $T(n) = O(n)$

Base: Okay! $T(2) = T(1) + T(1) + 1 = 3 = O(2)$

Hipótese: $T(n) \leq cn$, para todo $m < n$.

Passo indutivo:

$$T(n) = T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + 1$$

$$\leq c\lceil \frac{n}{2} \rceil + c\lfloor \frac{n}{2} \rfloor + 1 = cn + 1$$

Resolver a recorrência

$$T(n) = \begin{cases} 1 & \text{, se } n = 1 \\ T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + 1 & \text{, se } n > 1 \end{cases}$$

Palpite: $T(n) = O(n)$

Base: Okay! $T(2) = T(1) + T(1) + 1 = 3 = O(2)$

Hipótese: $T(n) \leq cn$, para todo $m < n$.

Passo indutivo:

$T(n) = T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + 1$

$\leq c\lceil \frac{n}{2} \rceil + c\lfloor \frac{n}{2} \rfloor + 1 = cn + 1$ (?)

Solução para (?)

Substituir a hipótese por $T(n) \leq cn - b$, com $b > 0$. $T(n) \leq cn - b$ permite as escolhas $c = 2$, $b = 1$ e $n_0 = 1$

Método da árvore de recursão

Método da árvore de recursão

Esquema geral:

- Simplificar a fórmula de recorrência.
- Itere a recorrência visualmente como uma árvore.
- Indique em cada nó o custo correspondente às chamadas recursivas.
- Calcular a cada nível da árvore o custo que não corresponde às chamadas recursivas
- A solução da recorrência é a soma de todos os custos em cada nível.

Método da árvore de recursão

Importante:

Depois de encontrar a solução deve-se aplicar o método de substituição!

Método da árvore de recursão

Resolver a recorrência a seguir para uma constante $c > 0$:

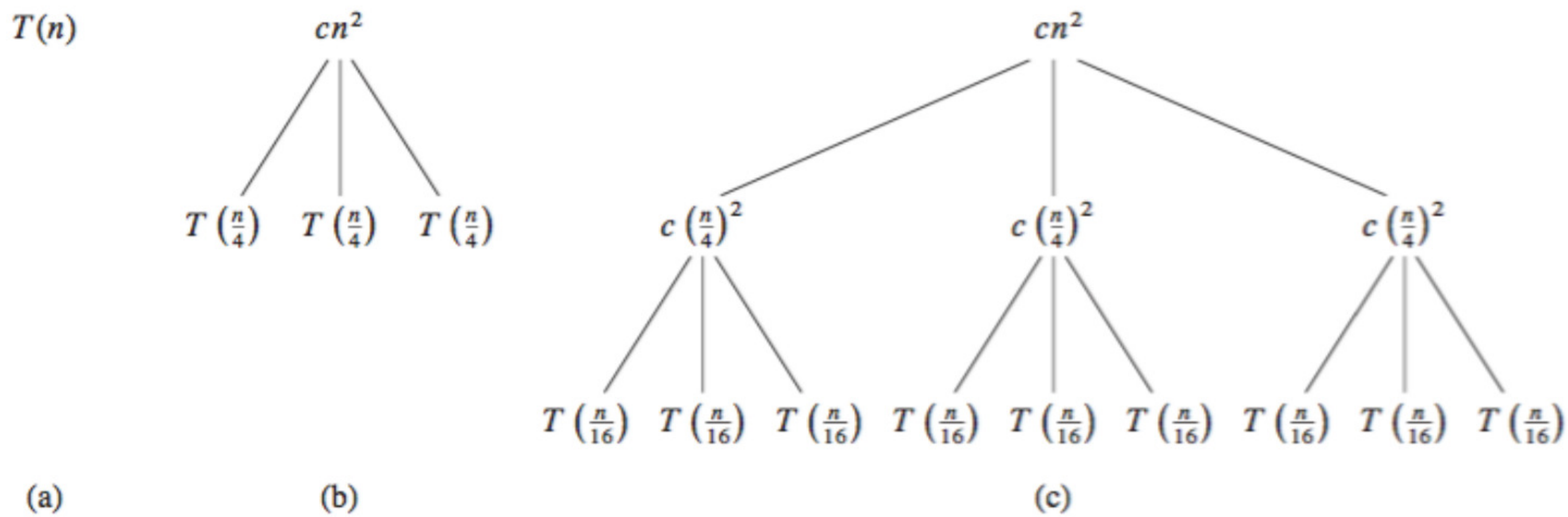
$$T(n) = \begin{cases} \Theta(1) & \text{, se } n \leq 3 \\ 3T(\lfloor \frac{n}{4} \rfloor) + \Theta(n^2) & \text{, se } n \geq 4 \end{cases}$$

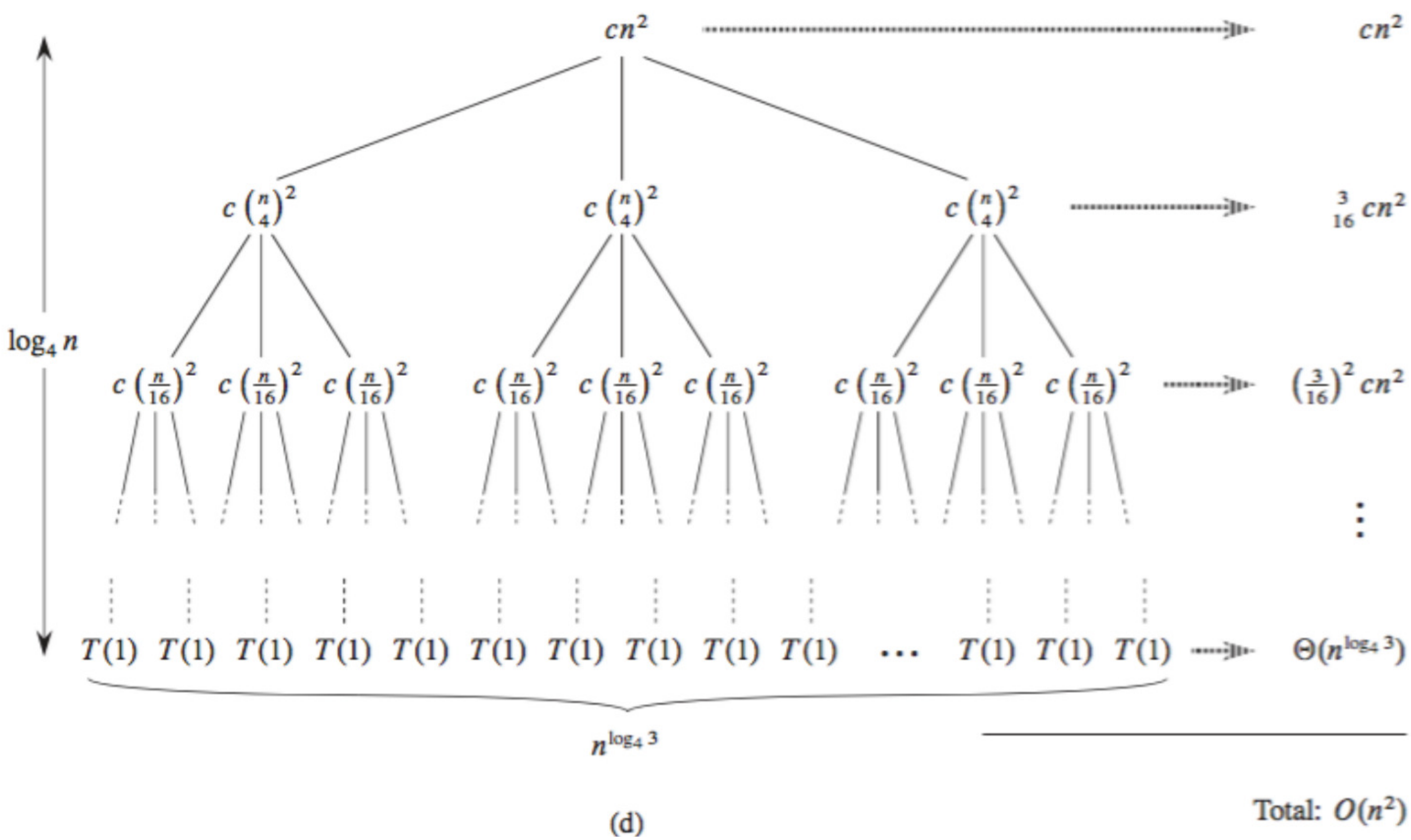
Método da árvore de recursão

Simplifique a recorrência:

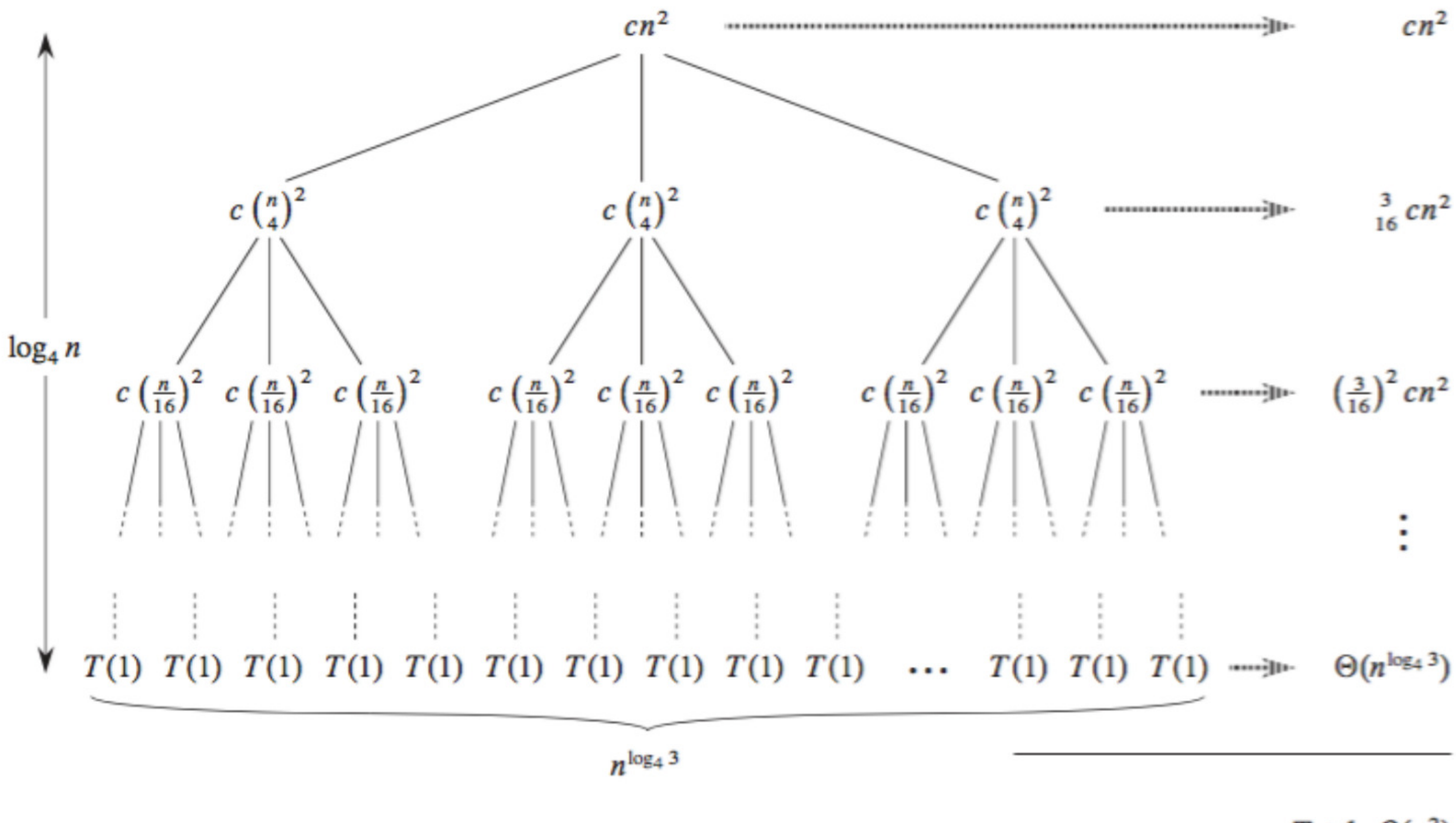
$$T(n) = \begin{cases} \Theta(1) & \text{, se } n \leq 3 \\ 3T(\frac{n}{4}) + cn^2 & \text{, se } n \geq 4^i \text{ (conveniência)} \end{cases}$$

$$T(n) = \begin{cases} \Theta(1) & \text{, se } n \leq 3 \\ 3T(\frac{n}{4}) + cn^2 & \text{, se } n \geq 4^i \end{cases}$$



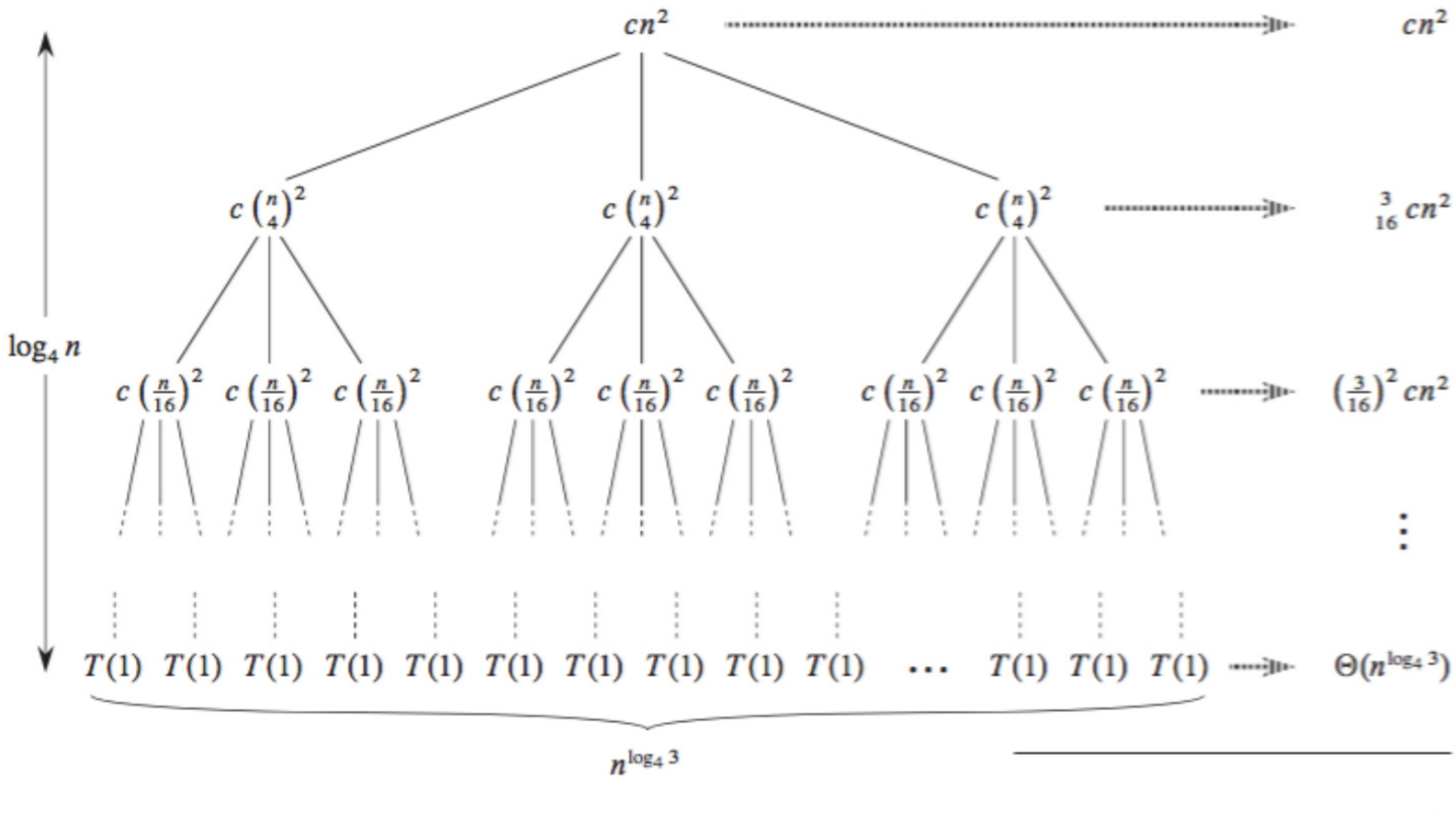


A altura da árvore de recursão é: $\log_4 n$



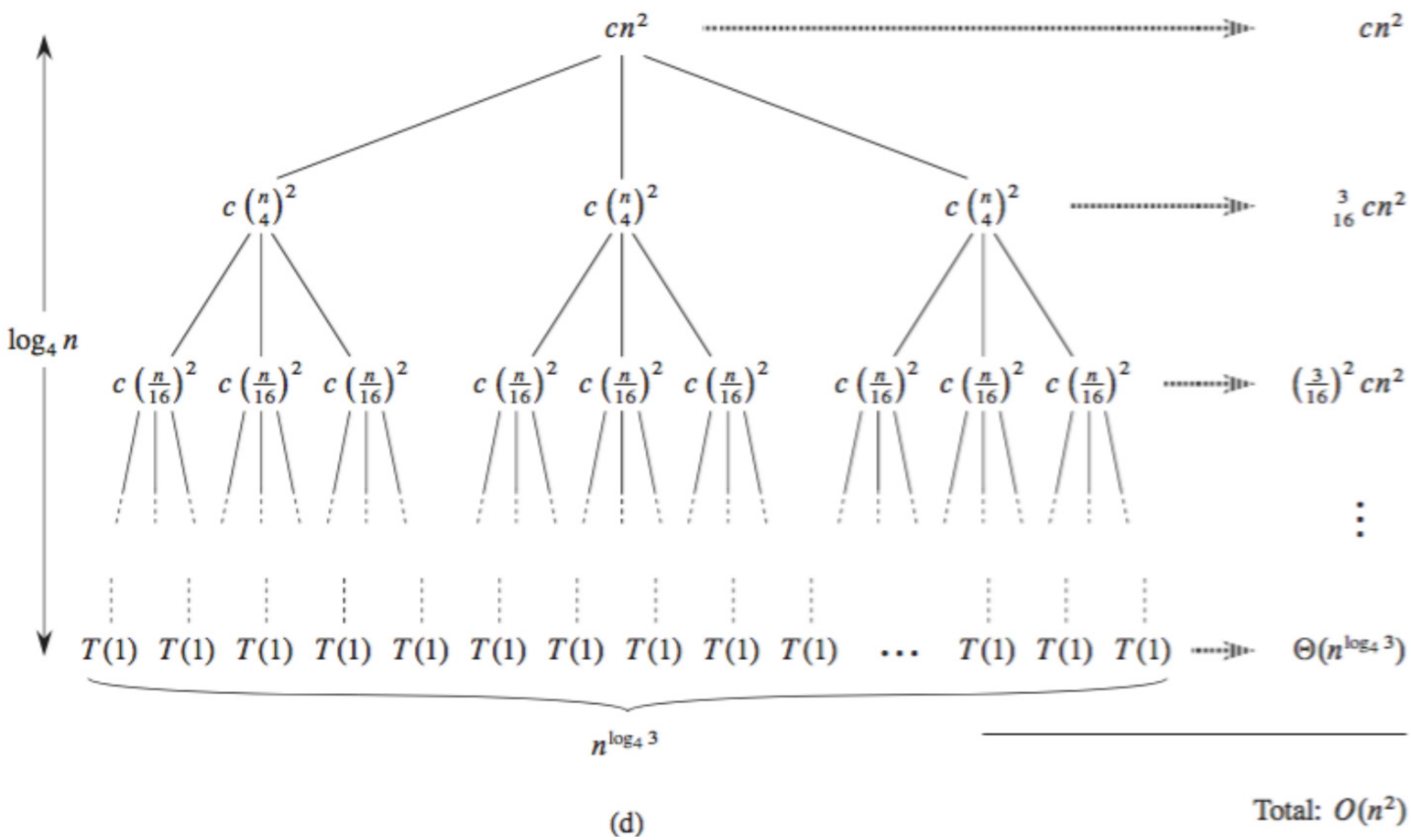
(d)

3^i é número de nós na profundidade $i = 0, 1, \cdots, (\log_4 n - 1)$



(d)

Cada nó na profundidade i tem custo $c(n/4^i)^2$, então o custo total é $3^i c(n/4^i)^2 = (3/16)^i cn^2$



No último nível, cada nó contribui com custo $T(1)$. Nesse nível existem $3^{\log_4 n} = n^{\log_4 3}$ nós.

Método da árvore de recursão

$$T(n) = \begin{cases} \Theta(1) & \text{, se } n \leq 3 \\ 3T(\frac{n}{4}) + cn^2 & \text{, se } n \geq 4^i \text{ (conveniência)} \end{cases}$$

Então o custo total para a recorrência é:

$$T(n) = cn^2 + \frac{3}{16}cn^2 + \left(\frac{3}{16}\right)^2 cn^2 + \cdots + \left(\frac{3}{16}\right)^{\log_4 n - 1} cn^2 + \Theta(n^{\log_4 3})$$

Método da árvore de recursão

$$T(n) = \begin{cases} \Theta(1) & \text{, se } n \leq 3 \\ 3T(\frac{n}{4}) + cn^2 & \text{, se } n \geq 4^i \text{ (conveniência)} \end{cases}$$

Então o custo total para a recorrência é:

$$\begin{aligned} T(n) &= cn^2 + \tfrac{3}{16}cn^2 + \left(\tfrac{3}{16}\right)^2 cn^2 + \cdots + \left(\tfrac{3}{16}\right)^{\log_4 n - 1} cn^2 + \Theta(n^{\log_4 3}) \\ &= \sum_{i=0}^{\log_4 n - 1} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3}) \end{aligned}$$

Método da árvore de recursão

$$T(n) = \begin{cases} \Theta(1) & \text{, se } n \leq 3 \\ 3T(\frac{n}{4}) + cn^2 & \text{, se } n \geq 4^i \text{ (conveniência)} \end{cases}$$

Então o custo total para a recorrência é:

$$\begin{aligned} T(n) &= cn^2 + \tfrac{3}{16}cn^2 + \left(\tfrac{3}{16}\right)^2 cn^2 + \cdots + \left(\tfrac{3}{16}\right)^{\log_4 n - 1} cn^2 + \Theta(n^{\log_4 3}) \\ &= \sum_{i=0}^{\log_4 n - 1} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3}) \\ &\leq cn^2 \sum_{i=0}^{\infty} \left(\frac{3}{16}\right)^i + \Theta(n^{\log_4 3}) \end{aligned}$$

Método da árvore de recursão

Série geométrica:

$$\sum_{i=0}^{\infty} r^i = \frac{1}{1-r}, \text{ se } |r| < 1,$$

Método da árvore de recursão

$$T(n) = \begin{cases} \Theta(1) & \text{, se } n \leq 3 \\ 3T(\frac{n}{4}) + cn^2 & \text{, se } n \geq 4^i \text{ (conveniência)} \end{cases}$$

Então o custo total para a recorrência é:

$$\begin{aligned} T(n) &= cn^2 + \frac{3}{16}cn^2 + \left(\frac{3}{16}\right)^2 cn^2 + \cdots + \left(\frac{3}{16}\right)^{\log_4 n - 1} cn^2 + \Theta(n^{\log_4 3}) \\ &= \sum_{i=0}^{\log_4 n - 1} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3}) \\ &\leq cn^2 \sum_{i=0}^{\infty} \left(\frac{3}{16}\right)^i + \Theta(n^{\log_4 3}) \\ &= \frac{1}{1 - \frac{3}{16}} cn^2 + \Theta(n^{\log_4 3}) \end{aligned}$$

Método da árvore de recursão

$$T(n) = \begin{cases} \Theta(1) & \text{, se } n \leq 3 \\ 3T(\frac{n}{4}) + cn^2 & \text{, se } n \geq 4^i \text{ (conveniência)} \end{cases}$$

Então o custo total para a recorrência é:

$$\begin{aligned} T(n) &= cn^2 + \frac{3}{16}cn^2 + \left(\frac{3}{16}\right)^2 cn^2 + \cdots + \left(\frac{3}{16}\right)^{\log_4 n - 1} cn^2 + \Theta(n^{\log_4 3}) \\ &= \sum_{i=0}^{\log_4 n - 1} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3}) \\ &\leq cn^2 \sum_{i=0}^{\infty} \left(\frac{3}{16}\right)^i + \Theta(n^{\log_4 3}) \\ &= \frac{16}{13}cn^2 + \Theta(n^{\log_4 3}) \end{aligned}$$

Método da árvore de recursão

$$T(n) = \begin{cases} \Theta(1) & \text{, se } n \leq 3 \\ 3T(\frac{n}{4}) + cn^2 & \text{, se } n \geq 4^i \text{ (conveniência)} \end{cases}$$

Então o custo total para a recorrência é:

$$T(n) = \frac{16}{13}cn^2 + \Theta(n^{\log_4 3})$$

Como $n^{\log_4 3} < n^2$, $\frac{16}{13}cn^2$ é “dominante” e $T(n) = O(n^2)$

De volta ao método da substituição

Agora, vamos verificar a fórmula de recorrência anterior usando o método da substituição.

$$T(n) = 3T\left(\left\lfloor \frac{n}{4} \right\rfloor\right) + \Theta(n^2)$$

Queremos mostrar que $T(n) \leq dn^2$, para alguma constante $d > 0$.

Pela hipótese de indução, nós temos que $T(\lfloor \frac{n}{4} \rfloor) = d\lfloor \frac{n}{4} \rfloor^2$

De volta ao método da substituição

Então usando a mesma constante $c > 0$ como antes, nós temos:

$$T(n) \leq 3T(\lfloor \tfrac{n}{4} \rfloor) + cn^2$$

De volta ao método da substituição

Então usando a mesma constante $c > 0$ como antes, nós temos:

$$
\begin{aligned}
T(n) &\leq 3T(\lfloor \tfrac{n}{4} \rfloor) + cn^2 \\
&\leq 3d\lfloor \tfrac{n}{4} \rfloor^2 + cn^2
\end{aligned}
$$

De volta ao método da substituição

Então usando a mesma constante $c > 0$ como antes, nós temos:

$$\begin{aligned} T(n) &\leq 3T(\lfloor \tfrac{n}{4} \rfloor) + cn^2 \\ &\leq 3d\lfloor \tfrac{n}{4} \rfloor^2 + cn^2 \\ &\leq 3d(\tfrac{n}{4})^2 + cn^2 \end{aligned}$$

De volta ao método da substituição

Então usando a mesma constante $c > 0$ como antes, nós temos:

$$
\begin{aligned}
T(n) &\leq 3T(\lfloor \tfrac{n}{4} \rfloor) + cn^2 \\
&\leq 3d\lfloor \tfrac{n}{4} \rfloor^2 + cn^2 \\
&\leq 3d(\tfrac{n}{4})^2 + cn^2 \\
&\leq \tfrac{3}{16}dn^2 + cn^2
\end{aligned}
$$

De volta ao método da substituição

Então usando a mesma constante $c > 0$ como antes, nós temos:

$$
\begin{aligned}
T(n) &\leq 3T(\lfloor \tfrac{n}{4} \rfloor) + cn^2 \\
&\leq 3d\lfloor \tfrac{n}{4} \rfloor^2 + cn^2 \\
&\leq 3d(\tfrac{n}{4})^2 + cn^2 \\
&\leq \tfrac{3}{16}dn^2 + cn^2 \\
&\leq dn^2 \text{ (quando } c \leq (13/16)d\text{, i.e. } d \geq (16/13)c)
\end{aligned}
$$

Método Mestre

Método Mestre

Recorrências de algoritmos que requerem chamadas recursivas para resolver problemas de tamanho (n/b) com custo $f(n)$ em passos de divisão e conquista podem ser escritas (com constantes $a \geq 1$ e $b > 1$) da seguinte forma:

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

Método Mestre

Assumindo que a base indutiva tem custo constante, temos um método geral para descrever solução de algumas destas recorrências.

A recorrência anterior também inclui casos $\lceil n/b \rceil$ e $\lfloor n/b \rfloor$.

Teorema Mestre

Sejam $a \geq 1$ e $b > 1$ constantes, $f(n)$ uma função e $T(n)$ definida para inteiros não-negativos como a recorrência.

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

Teorema Mestre

Então, $T(n)$ tem os seguintes limites assintóticos

(1) Se $f(n) = O(n^{\log_b a - \varepsilon})$ para alguma constante $\varepsilon > 0$, então $T(n) = \Theta(n^{\log_b a})$

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

(3) Se $f(n) = \Omega(n^{\log_b a + \varepsilon})$ para alguma constante $\varepsilon > 0$, e se $af(n/b) \leq cf(n)$ para alguma constante $c < 1$ e todos n suficientemente grandes, então $T(n) = \Theta(f(n))$

Exemplos (a)

$$T(n) = \begin{cases} \Theta(1) & \text{, se } n = 1 \\ 4T(\lfloor \frac{n}{2} \rfloor) + n & \text{, se } n > 1 \end{cases}$$

Exemplos (a)

$$T(n) = \begin{cases} \Theta(1) & \text{, se } n = 1 \\ 4T(\lfloor \frac{n}{2} \rfloor) + n & \text{, se } n > 1 \end{cases}$$

$a = 4$, $b = 2$, $f(n) = n$

Exemplos (a)

$$T(n) = \begin{cases} \Theta(1) & \text{, se } n = 1 \\ 4T(\lfloor \frac{n}{2} \rfloor) + n & \text{, se } n > 1 \end{cases}$$

$a = 4,\ b = 2,\ f(n) = n$

$n^{\log_b a} = n^{\log_2 4} = n^2$

Exemplos (a)

$$T(n) = \begin{cases} \Theta(1) & \text{, se } n = 1 \\ 4T(\lfloor \frac{n}{2} \rfloor) + n & \text{, se } n > 1 \end{cases}$$

$a = 4$, $b = 2$, $f(n) = n$

$n^{\log_b a} = n^{\log_2 4} = n^2$

$f(n) = n$ está "por cima" ou "por baixo" de $n^{\log_b a} = n^2$?

Exemplos (a)

$$T(n) = \begin{cases} \Theta(1) & \text{, se } n = 1 \\ 4T(\lfloor \frac{n}{2} \rfloor) + n & \text{, se } n > 1 \end{cases}$$

$a = 4$, $b = 2$, $f(n) = n$

$n^{\log_b a} = n^{\log_2 4} = n^2$

$f(n) = n$ está “por cima” ou “por baixo” de $n^{\log_b a} = n^2$?

$f(n) = n = O(n^{2-\varepsilon})$ para $\varepsilon = 1$ (Caso 1)

Exemplos (a)

$$T(n) = \begin{cases} \Theta(1) & \text{, se } n = 1 \\ 4T(\lfloor \frac{n}{2} \rfloor) + n & \text{, se } n > 1 \end{cases}$$

$a = 4,\ b = 2,\ f(n) = n$

$n^{\log_b a} = n^{\log_2 4} = n^2$

$f(n) = n$ está "por cima" ou "por baixo" de $n^{\log_b a} = n^2$?

$f(n) = n = O(n^{2-\varepsilon})$ para $\varepsilon = 1$ (Caso 1)

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

Exemplos (b)

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

Exemplos (b)

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

$a = 1$, $b = 3/2$, $f(n) = 1$

Exemplos (b)

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

$$a = 1,\ b = 3/2,\ f(n) = 1$$

$$n^{\log_b a} = n^{\log_{3/2} 1} = n^0 = 1$$

Exemplos (b)

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

$a = 1$, $b = 3/2$, $f(n) = 1$

$n^{\log_b a} = n^{\log_{3/2} 1} = n^0 = 1$

$f(n) = 1$ está "por cima" ou "por baixo" de $n^{\log_b a} = 1$?

Exemplos (b)

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

$a = 1$, $b = 3/2$, $f(n) = 1$

$n^{\log_b a} = n^{\log_{3/2} 1} = n^0 = 1$

$f(n) = 1$ está "por cima" ou "por baixo" de $n^{\log_b a} = 1$?

$f(n) = 1 = \Theta(1)$ (Caso 2)

Exemplos (b)

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

$a = 1$, $b = 3/2$, $f(n) = 1$

$n^{\log_b a} = n^{\log_{3/2} 1} = n^0 = 1$

$f(n) = 1$ está “por cima” ou “por baixo” de $n^{\log_b a} = 1$?

$f(n) = 1 = \Theta(1)$ (Caso 2)

$\therefore T(n) = \Theta(1 \cdot \log n)$

Exemplos (c)

$T(n) = 3T(\lfloor \frac{n}{4} \rfloor) + n \log n$, para $n = 4, 8, 16, \cdots$

Exemplos (c)

$T(n) = 3T(\lfloor \frac{n}{4} \rfloor) + n \log n$, para $n = 4, 8, 16, \cdots$

$a = 3$, $b = 4$, $f(n) = n \log n$

Exemplos (c)

$T(n) = 3T(\lfloor \frac{n}{4} \rfloor) + n \log n$, para $n = 4, 8, 16, \cdots$

$a = 3$, $b = 4$, $f(n) = n \log n$

$n^{\log_b a} = n^{\log_4 3} = n^{0.8}$

$f(n) = n \log n$ está "por cima" ou "por baixo" de $n^{\log_b a} = n^{0.8}$?

Exemplos (c)

$T(n) = 3T(\lfloor \frac{n}{4} \rfloor) + n \log n$, para $n = 4, 8, 16, \cdots$

$a = 3$, $b = 4$, $f(n) = n \log n$

$n^{\log_b a} = n^{\log_4 3} = n^{0.8}$

$f(n) = n \log n$ está "por cima" ou "por baixo" de $n^{\log_b a} = n^{0.8}$?

$f(n) = n \log n = \Omega(n^{0.8+\varepsilon})$ para $\varepsilon = 0.2$ (Caso 3)

Exemplos (c)

Ainda temos que verificar a condição de regularidade: $af(n/b) \leq cf(n)$ para alguma constante $c < 1$ e n suficientemente grande.

$a = 3$, $b = 4$, $f(n) = n \log n$

Exemplos (c)

Ainda temos que verificar a condição de regularidade: $af(n/b) \leq cf(n)$ para alguma constante $c < 1$ e n suficientemente grande.

$a = 3$, $b = 4$, $f(n) = n \log n$

$$3f(n/4) =$$

Exemplos (c)

Ainda temos que verificar a condição de regularidade: $af(n/b) \leq cf(n)$ para alguma constante $c < 1$ e n suficientemente grande.

$a = 3$, $b = 4$, $f(n) = n \log n$

$3f(n/4) =$
$3(n/4 \log n/4) =$

Exemplos (c)

Ainda temos que verificar a condição de regularidade: $af(n/b) \leq cf(n)$ para alguma constante $c < 1$ e n suficientemente grande.

$a = 3$, $b = 4$, $f(n) = n \log n$

$$3f(n/4) =$$
$$3(n/4 \log n/4) =$$
$$3n/4(\log n - \log 4) =$$

Exemplos (c)

Ainda temos que verificar a condição de regularidade: $af(n/b) \leq cf(n)$ para alguma constante $c < 1$ e n suficientemente grande.

$a = 3$, $b = 4$, $f(n) = n \log n$

$3f(n/4) =$
$3(n/4 \log n/4) =$
$3n/4(\log n - \log 4) =$
$3/4n \log n - 3/4n \log 2^2 =$

Exemplos (c)

Ainda temos que verificar a condição de regularidade: $af(n/b) \leq cf(n)$ para alguma constante $c < 1$ e n suficientemente grande.

$a = 3$, $b = 4$, $f(n) = n \log n$

$3f(n/4) =$
$3(n/4 \log n/4) =$
$3n/4(\log n - \log 4) =$
$3/4n \log n - 3/4n \log 2^2 =$
$3/4n \log n - 3/2n$

Exemplos (c)

Ainda temos que verificar a condição de regularidade: $af(n/b) \leq cf(n)$ para alguma constante $c < 1$ e n suficientemente grande.

$a = 3$, $b = 4$, $f(n) = n \log n$

$$
\begin{aligned}
&3f(n/4) = \\
&3(n/4 \log n/4) = \\
&3n/4(\log n - \log 4) = \\
&3/4n \log n - 3/4n \log 2^2 = \\
&3/4n \log n - 3/2n \leq \\
&3/4n \log n
\end{aligned}
$$

para todo $n \geq 1$ e $c = 3/4 < 1$.

Exemplos (c)

Ainda temos que verificar a condição de regularidade: $af(n/b) \leq cf(n)$ para alguma constante $c < 1$ e n suficientemente grande.

$a = 3$, $b = 4$, $f(n) = n \log n$

$$
\begin{aligned}
&3f(n/4) = \\
&3(n/4 \log n/4) = \\
&3n/4(\log n - \log 4) = \\
&3/4n \log n - 3/4n \log 2^2 = \\
&3/4n \log n - 3/2n \leq \\
&3/4n \log n
\end{aligned}
$$

para todo $n \geq 1$ e $c = 3/4 < 1$.

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

Lembre-se

O método mestre não resolve todas as recorrências!

Obrigado