Sumário

- Hipótese “facilitadoras”
- Métodos para solucionar recorrências
  - Método de Substituição
  - Método da Árvore de recursão
  - Método mestre

Hipóteses "facilitadoras"

- Ignoramos os arredondamentos para cima ou para baixo. Assim, valores inteiros são assumidos para as variáveis.
- Casos base da recursão podem ser ignorados. Assume-se que T(n) tenha valor constante e baixo quando n é baixo.
- Por exemplo, no caso do algoritmo MERGE-SORT temos:

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

- Todavia, podemos considerar apenas:

$$T(n) = 2T(n/2) + \Theta(n)$$

Métoda da Substituição

- **Passo1**: Suponha uma possível solução.
- **Passo 2**: Prove por indução matemática que a solução proposta é válida.

Métoda da Substituição

Exemplo1: $T(n) = 2T(\lfloor n/2 \rfloor) + n$

Passo 1: Suponha que $T(n) = O(n \lg n)$ **(Pq?)**

Passo 2: Provar por indução que

$T(n) \leq cn \lg n$ para algum c>0.

Método de Substituição

- ✓ Por hipótese de indução, para n<k, temos

  $T(\mathrm{n}) \leq cn \lg n$

- ✓ Vamos provar para n =k com $T(k) = 2T(\lfloor k/2 \rfloor) + k$
- ✓ Como k/2 < k, por indução, temos $T(k/2) \leq c\,(k/2) \lg (k/2)$
- ✓ Dada a natureza não decrescente das funções envolvidas, ignoramos a função piso $\lfloor k/2 \rfloor$. Logo,

$$T(k) = 2T(n/2) + k \leq 2c\,(k/2) \lg (k/2) + k$$

$$T(k) \leq c\,k \lg k - c\,k \lg 2 + k$$

$$T(k) \leq c\,k \lg k - c\,k + k$$

$$T(k) \leq c\,k \lg k - (c-1)k$$

$$T(k) \leq c\,k \lg k, \quad para\ c \geq 1$$

Método da Substituição

Exemplo 2: $T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1$

Passo1: Suponha que $T(n) = O(n)$

Passo2: Provar por indução que $T(n) \leq cn$ para $c > 0$

Dem:

✓Provando para n = k, temos que verificar $T(k) \leq ck$

✓$T(k) = T(\lfloor k/2 \rfloor) + T(\lceil k/2 \rceil) + 1 \leq ck/2 + ck/2 + 1$

✓$T(k) \leq ck+1 \not\Rightarrow T(k) \leq ck$ PROBLEMA!!

Métoda da Substituição

Passo 1: Suponha que $T(n) = O(n)$ (MANTIDO)

Passo 2: Provar por indução que $T(n) \leq cn-d$ para $c,d>0$

Dem: Provando para n = k, temos que verificar

$T(k) \leq ck-d$

$T(k) = T(\lfloor k/2 \rfloor) + T(\lceil k/2 \rceil) + 1 \leq ck/2-d+ck/2-d+1$

$T(k) \leq ck-2d+1$

$T(k) \leq ck-d$, para $d \geq 1$ (Pq????)

Árvore de Recursão

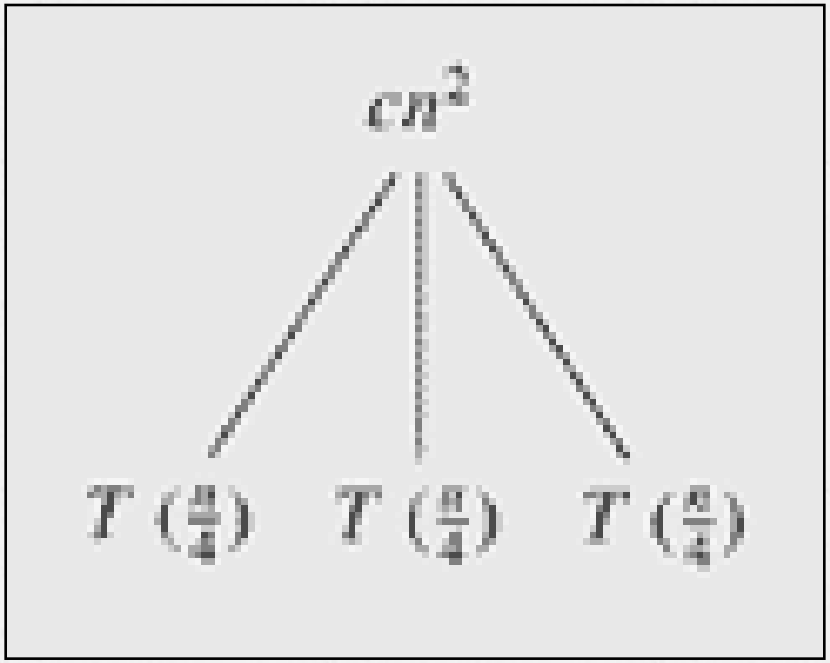
- Trata-se de uma representação visual da hierarquia das chamadas recursivas.
- Cada nó representa o custo de um subproblema.
- Auxilia na definição da função utilizada no Passo 1 do Método de Substituição (MS).

Árvore de Recursão

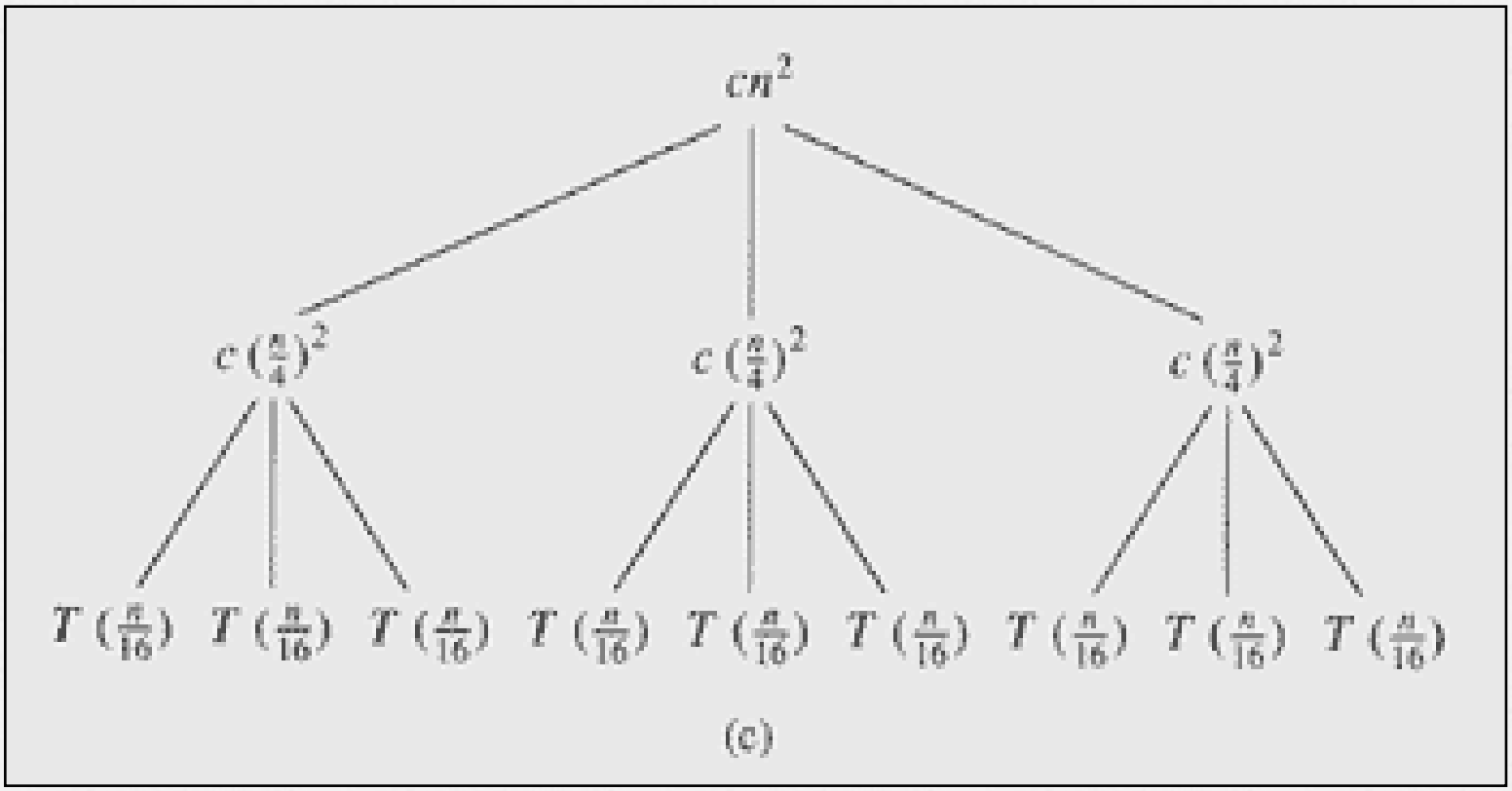
Exemplo1: $T(n) = 3T(\lfloor n/4 \rfloor) + \Theta(n^2)$

- Hipóteses “facilitadoras”:
  - A função piso (assim como a teto) não costuma ter relevância quando resolvendo recorrências.
  - Por conveniência, vamos assumir que n é uma potência do número 4. Assim, todo subproblema terá um tamanho inteiro.

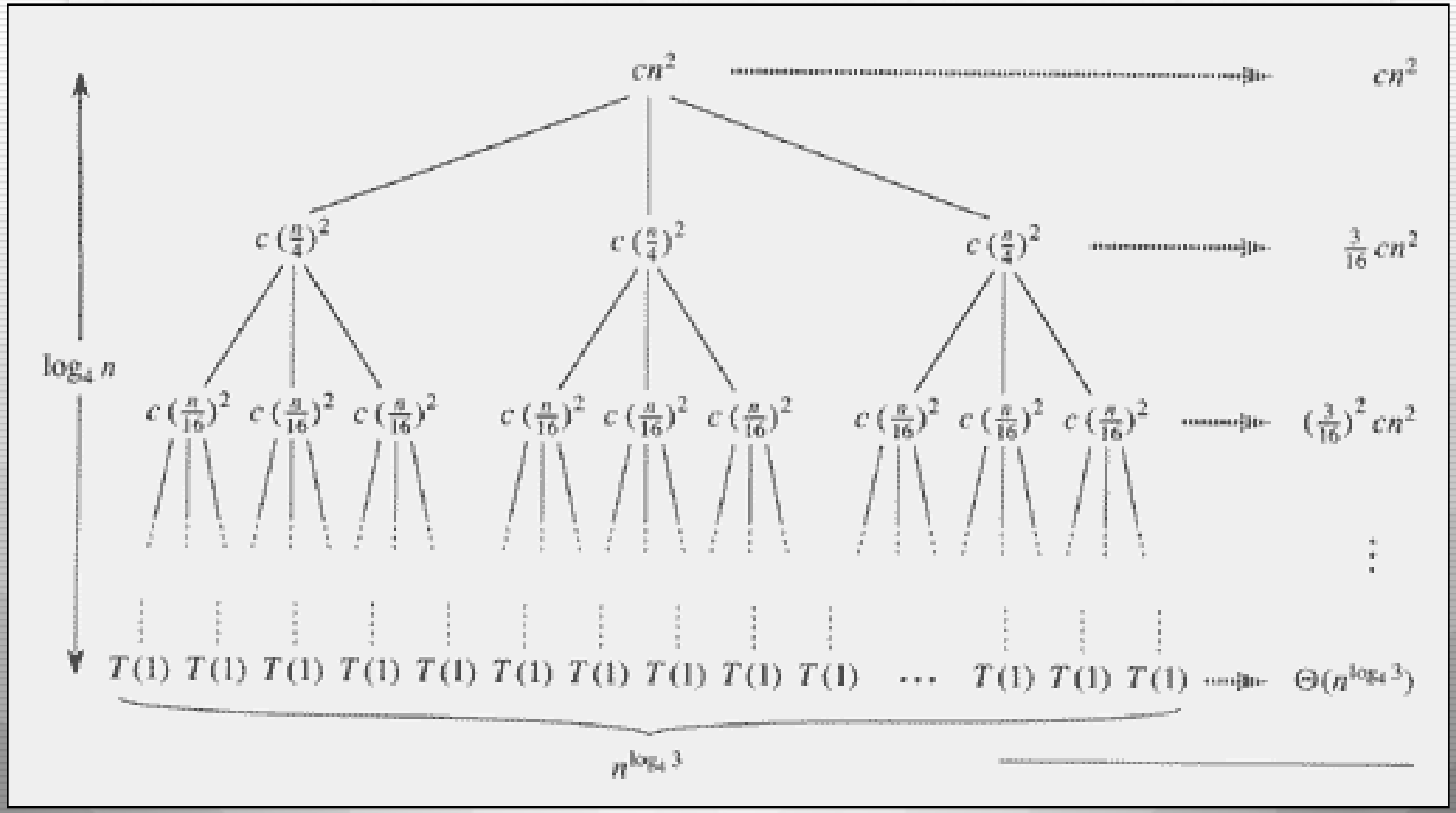
Árvore de Recursão



Árvore de Recursão



Árvore de Recursão



Árvore de Recursão

- O tamanho dos subproblemas decrescem a um fator 4 cada vez que descemos um nível na árvore.
- O tamanho de um subproblema para um nó na profundidade i é $n/4^i$.
- Assim, $n/4^i = 1 \Rightarrow i=\log_4 n$, ou seja, a árvore terá $\log_4 n + 1$ níveis.

Árvore de Recursão

- No nível i, o número de nós será 3^i.
- O custo associado a cada nó será $c(n/4^i)^2$.
- Logo, o custo no nível i será

$$3^i c(n/4^i)^2 = (3/16)^i cn^2.$$

Árvore de Recursão

- No nível $i=\log_4 n$, temos $3^{\log_4 n}$ nós.
- Assumindo custo constante T(1) **(Pq???).**
- O custo total no último nível será

$$3^{\log_4 n} T(1) = \Theta(3^{\log_4 n})$$

Árvore de Recursão

- O custo total considerando todos os níveis será:

$$T(n) = \sum_{i=0}^{\log_4 n - 1} (3/16)^i cn^2 + \Theta(n^{\log_4 3})$$

$$T(n) \leq \sum_{i=0}^{\infty} (3/16)^i cn^2 + \Theta(n^{\log_4 3})$$

$$T(n) \leq (1/(1-3/16))cn^2 + \Theta(n^{\log_4 3})$$

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

$$T(n) = O(n^2)$$

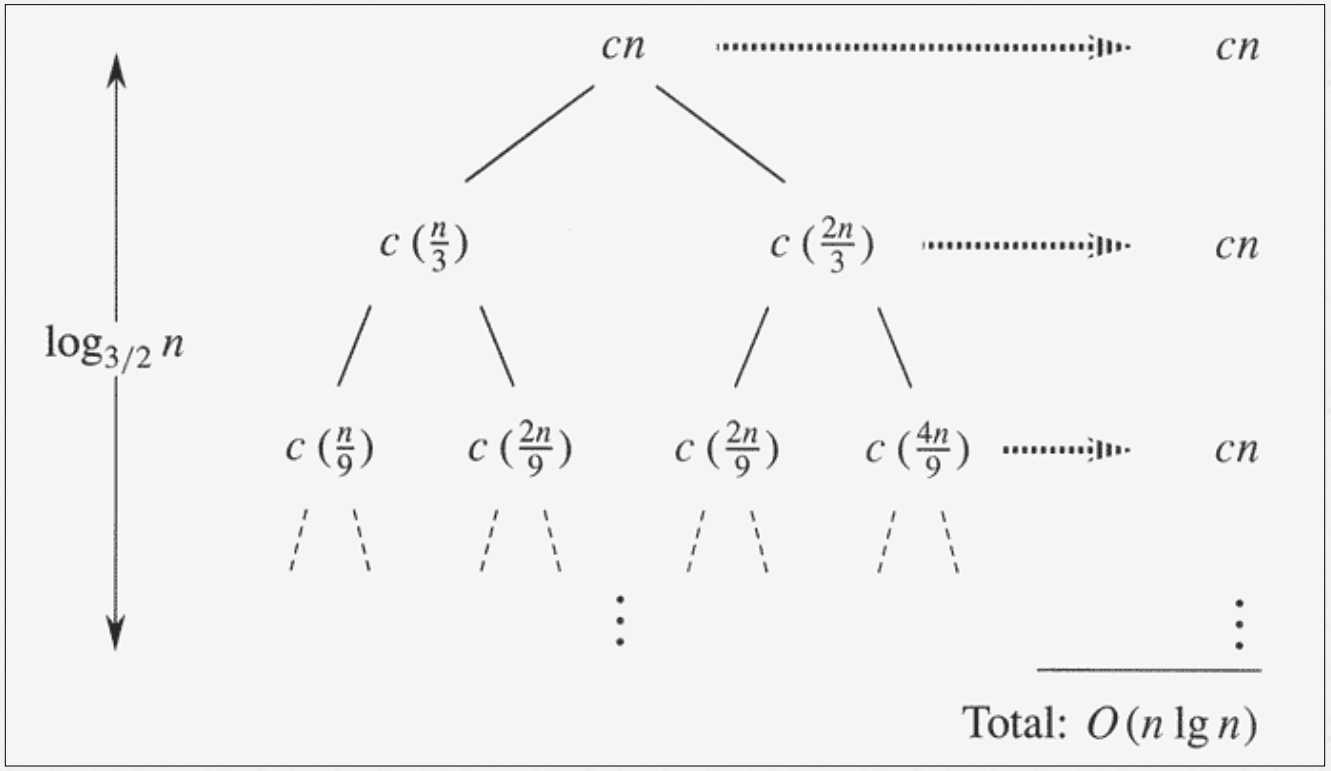
!!!NÃO TÃO RÁPIDO!!!

Árvore de Recursão

- Via árvore de recursão, temos uma suposição para o Passo 1 do Método de Substituição: $T(n) \leq dn^2$.
- Agora, vamos para o Passo 2:
- A H.I. ocorre para k < n, para n=k:
- $T(k) = 3T(\lfloor n/4 \rfloor) + \Theta(n^2) \leq 3T(\lfloor n/4 \rfloor) + cn^2$
- $\leq 3d(n/4)^2 + cn^2 \leq (3/16)dn^2 + cn^2$
- $\leq [(3/16)d + c]n^2$
- $\leq dn^2$ para $d \geq (16/13)c$ **(Pq???)**

Árvore de Recursão

Exemplo 2: $T(n) = T(n/3) + T(2n/3) + O(n)$



Árvore de Recursão

- Novamente, vamos desconsiderar funções piso e teto.
- O maior caminho do nó raiz até um nó folha passa pelos nós n, (2/3)n, $(2/3)^2$n,…,$(2/3)^k$n,…, 1.
- Logo, $(2/3)^k n=1 \Rightarrow k = \log_{3/2} n$ (altura!!)
- Nem todo nível da árvore contribui com um custo cn.

Árvore de Recursão

- Considerando o custo nos nós folhas em uma árvore binária completa com altura $\log_{3/2} n$ (altura!!), teremos:

$$2^{\log_{3/2} n} = n^{\log_{3/2} 2}$$ folhas.

- Desde que o custo de cada folha é constante, o custo total no último nível poderia ser $\Theta(n^{\log_{3/2} 2})$

- Como $\log_{3/2} 2 > 1$, tal custo será $\omega(n \lg n)$

Árvore de Recursão

- Custo total superestimado:

$$T(n) = \sum_{i=0}^{\log_{3/2} n - 1} cn + \Theta(n^{\log_{3/2} 2})$$

- Podemos chegar em:

$$T(n) = O(n \lg n) + \omega(n \lg n)$$

- Vamos supor:

$$T(n) = O(n \lg n)$$

Árvore de Recorrência

- Podemos assumir

$$T(n) = T(n/3) + T(2n/3) + cn$$

- Supomos no Passo 1 do M.S.:

$$T(n) = O(n \lg n)$$

- Provaremos no Passo 2 do M.S.:

$$T(n) \leq dn \lg n \text{ para } d>0$$

Árvore de Recorrência

- Novamente, vamos ignorar o caso base
- Pela H.I. temos que para $n < k$ a desigualdade $T(n) \leq dn \lg n$ é válida.

Árvore de Recorrência

- Pela H.I, temos:

$$T(k/3) \leq d\ k/3 \lg k/3$$

$$T(2k/3) \leq d\ 2k/3 \lg 2k/3$$

- A relação de recorrência é:

$$T(k) = T(k/3) + T(2k/3) + ck$$

$$T(k) \leq [d\ (k/3) \lg (k/3)] + [d\ (2k/3) \lg (2k/3)] + ck$$

Árvore de Recorrência

$$T(k) \leq [d\,(k/3)\lg k - d\,(k/3)\lg 3] + [d\,(2k/3)\lg k - d\,(2k/3)\lg(3/2)] + ck$$

$$T(k) \leq dk \lg k - d[(k/3)\lg 3 + (2k/3)\lg(3/2)] + ck$$

$$T(k) \leq dk \lg k - dk[\lg 3 - 2/3] + ck$$

$$T(k) \leq dk \lg k \text{ para } d \geq c/(\lg 3 - (2/3))$$

Método Mestre

- Fornece uma receita para resolver recorrências que devem estar na forma:

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

com $a \geq 1$ e $b > 1$ sendo constantes e $f(n)$ uma função assintóticamente positiva.

Método Mestre

Teorema: Seja T(n)=aT(n/b)+f(n), com a>0 e b>0 constantes e f(n) assintoticamente positiva. Temos que:

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

Método Mestre

Exemplo 1: $T(n)=9T(n/3) + n$

Temos, $a=9$, $b=3$, $f(n)=n$. Pelo teorema anterior, $n^{\log_b(a)}=n^{\log_3(9)}=\Theta(n^2)$.

Como $f(n)=O(n^{\log_3(9-\epsilon)})$, onde $\epsilon=1$, podemos aplicar o caso 1 do teorema mestre, concluindo que $T(n)=\Theta(n^2)$

Método Mestre

Exemplo 2: $T(n)=T(2n/3) + 1$

Temos, $a=1$, $b=3/2$, $f(n)=1$ com $n^{\log_b(a)}=n^{\log_{3/2}(1)}=n^0=1$.

O caso 2 se aplica aqui, desde que $f(n)=\Theta(1)$, então a solução para a relação de recorrência será $T(n)=\Theta(\lg n)$

Método Mestre

Exemplo 3: $T(n)=3T(n/4) + n\lg n$

Temos, $a=3$, $b=4$, $f(n)=n\lg n$ com $n^{\log_b(a)}=n^{\log_4(3)}=O(n^{0.793})$. Desde que $f(n)=\Omega(n^{\log_4(3+\epsilon)})$, onde $\epsilon \approx 0.2$, o caso 3 se aplica se as condições de regularidade ocorrem.

Para n suficientemente grande, temos

$af(n/b)= 3(n/4)\lg(n/4) \leq (3/4)n\lg n = cf(n)$ para $c=3/4$.

Logo, pelo caso 3, a solução para a relação de recorrência será $T(n)=\Theta(n\lg n)$

Método Mestre

Exemplo 4: $T(n)=2T(n/2) + n\lg n$

Temos, $a=2$, $b=2$, $f(n)=n\lg n$ com $n^{\log_b(a)}=n$.

A função $f(n)=n\lg n$ é assintoticamente maior que $n^{\log_b(a)}=n$, mas não é polinomialmente maior:

$$f(n)/n^{\log_b(a)} = (n\lg n)/n = \lg n$$

onde $\lg n$ é assintoticamente menor que n^{ϵ} para qualquer constante positiva ϵ. Logo, o caso 3 não se aplica.