DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

Estruturas de Dados

Teorema Mestre

Professores: Anisio Lacerda
Wagner Meira Jr.

Divisão e Conquista

- Vimos na última aula que nem sempre é trivial resolver uma relação de recorrência
- Alguns algoritmos exibem comportamentos semelhantes, como por exemplo:
  - Fazer *a* chamadas recursivas para instâncias de tamanho *n/b*.
  - faz um processamento de custo *f(n)* com as respostas obtidas e retorna uma solução.
- Nesses casos a relação de recorrência será:

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

Divisão e Conquista

- Chamamos esse tipo de algoritmo de **divisão e conquista**.
  - O passo em que fazemos a divisão em subproblemas é a **divisão**.
  - O processamento com as respostas dos subproblemas para gerar a resposta do nosso subproblema atual é a **conquista**.
- Um algoritmo que representa bem essa ideia é o *Merge Sort*.
- Falaremos sobre ele em algumas aulas, então vamos utilizar outro exemplo.

Exemplo:

```
int power(int x, unsigned int y)
{
   if (y == 0)
      return 1;
   else if (y % 2 == 0)
      return power(x, y/2) * power(x, y/2);
   else
      return x * power(x,y/2) * power(x,y/2);
}
```

$$T(n) = 2T\left(\frac{n}{2}\right) + O(1)$$

Teorema Mestre

- Na última aula, utilizamos árvore de recursão para encontrar a ordem de complexidade de relações de recorrência semelhantes.
- O Teorema Mestre é uma “receita de bolo” para relações que satisfazem:

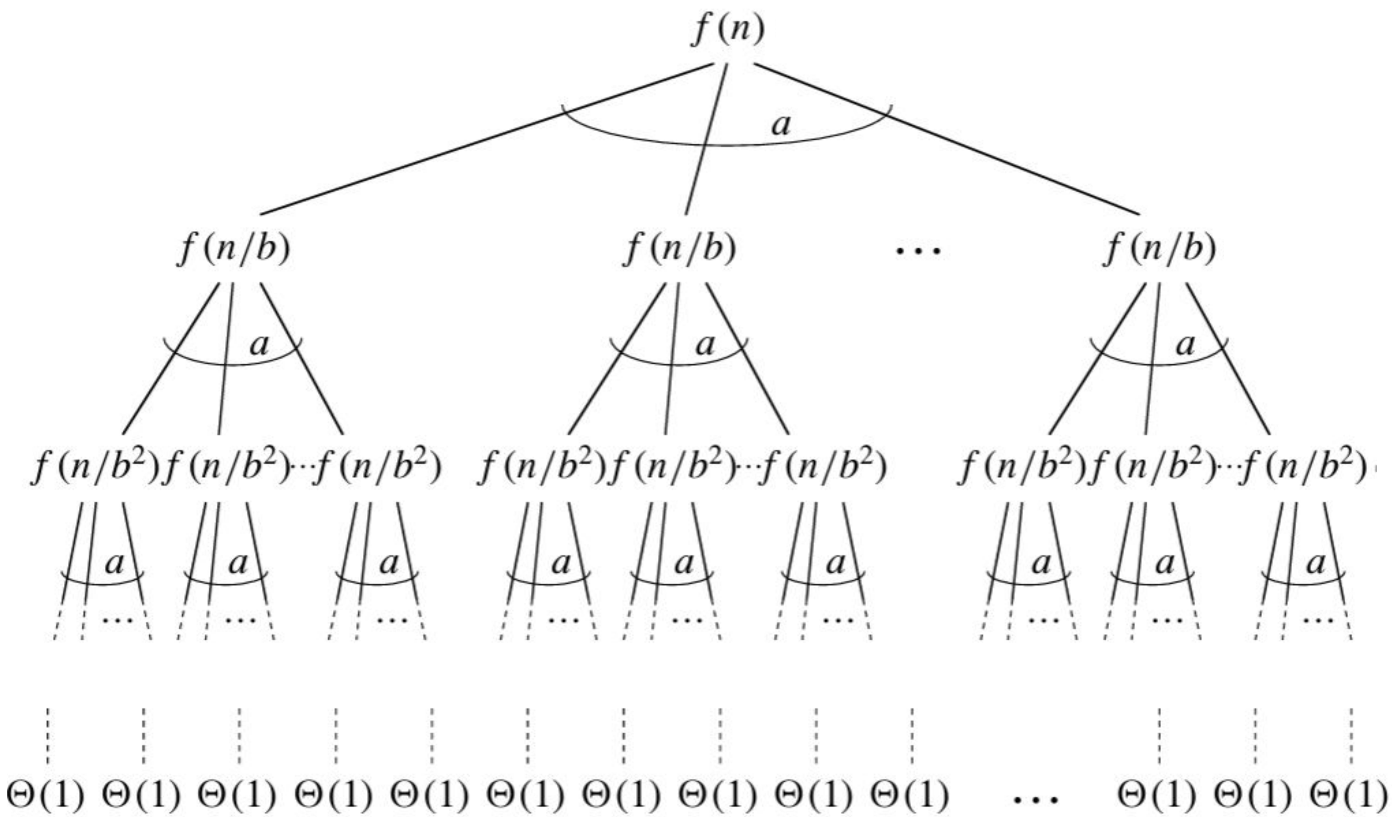
$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

- Onde $a \geq 1$ e $b > 1$ são constantes e $f(n)$ é uma função assintoticamente positiva.

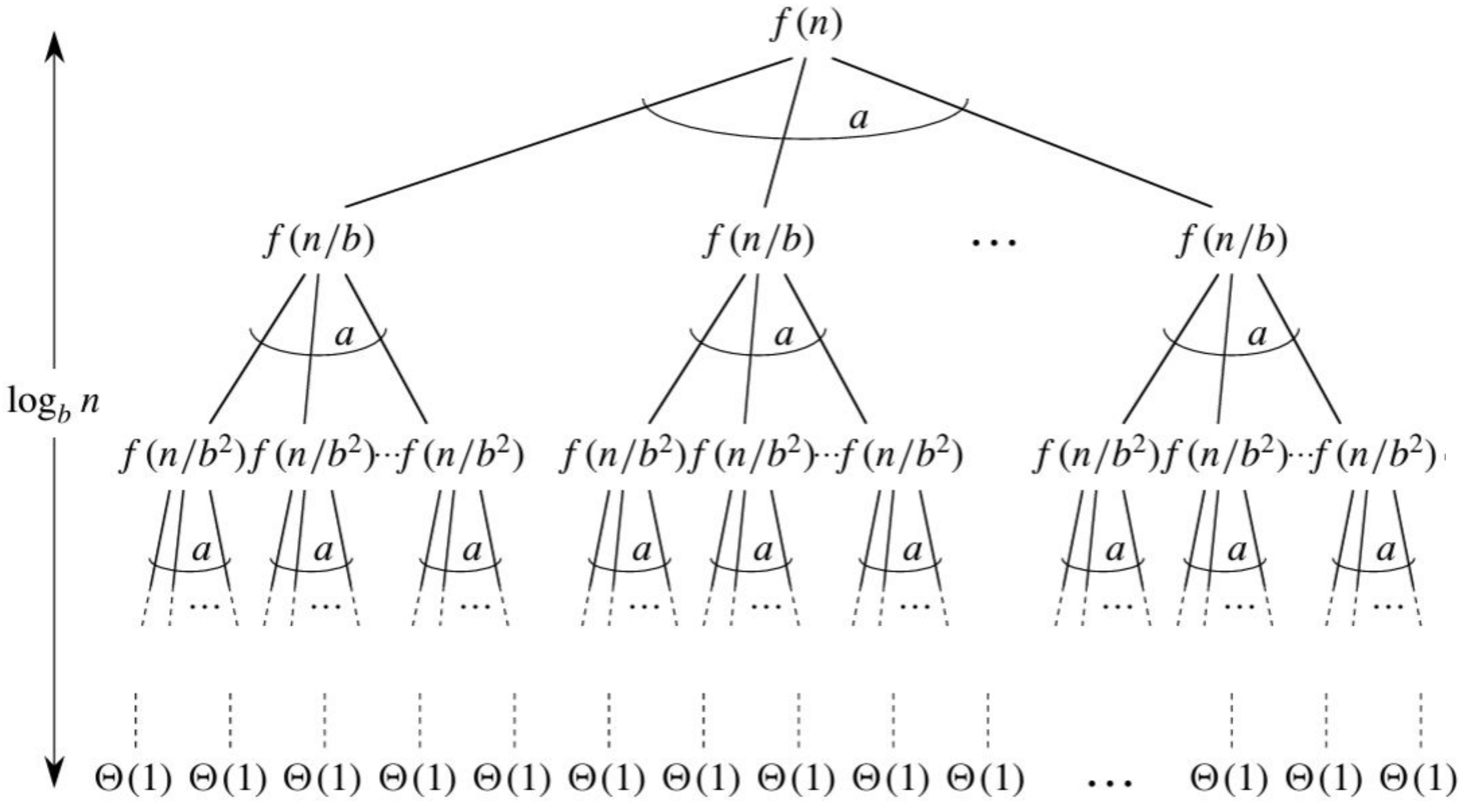
Teorema Mestre - Intuição

- A demonstração do Teorema Mestre não faz parte do escopo da disciplina
- A intuição não é muito diferente do que fizemos com a árvore de recursão.
- Expandir a árvore e somar os custos independentes à recursão em cada nível da árvore.

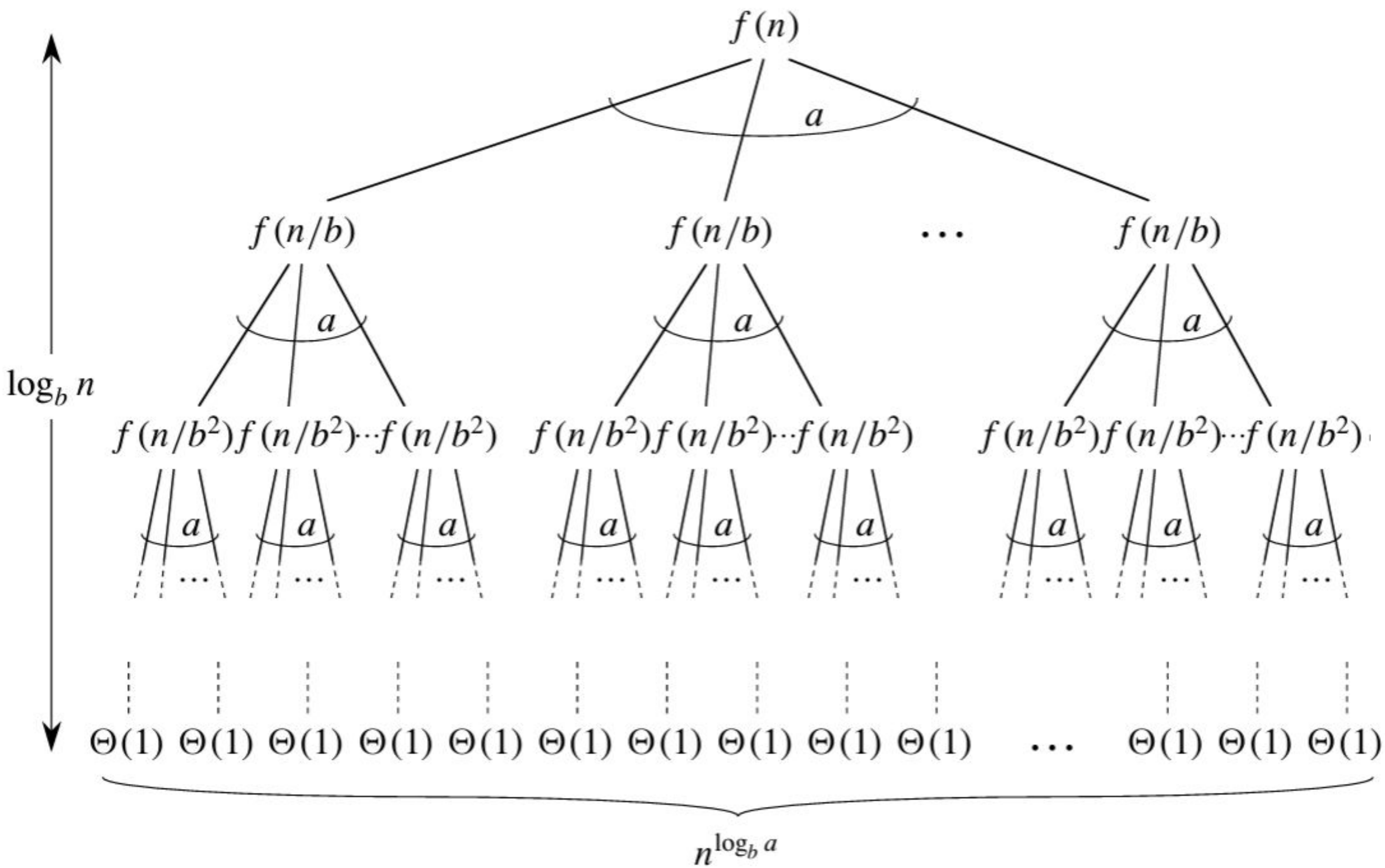
Teorema Mestre - Intuição



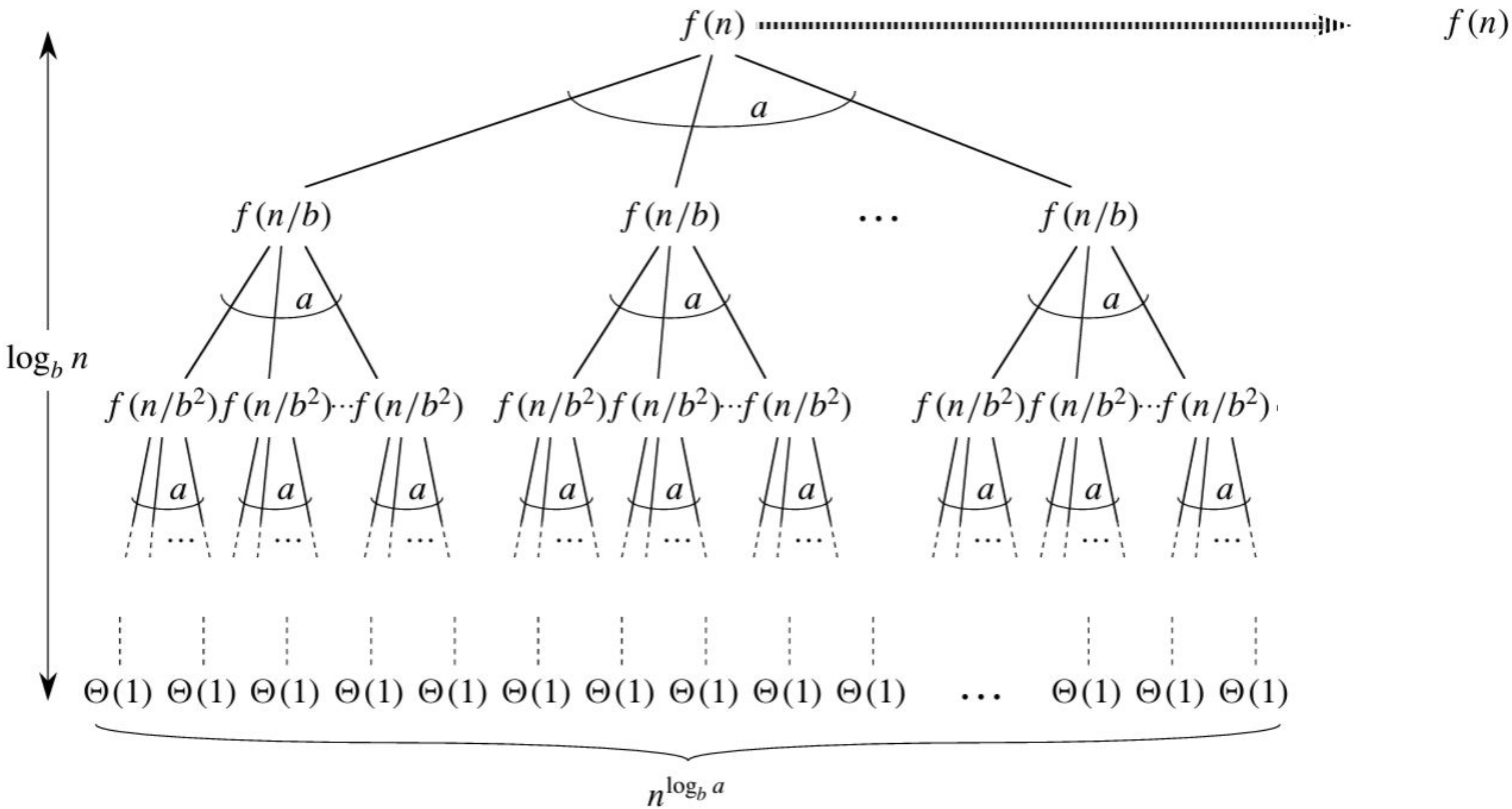
Teorema Mestre - Intuição



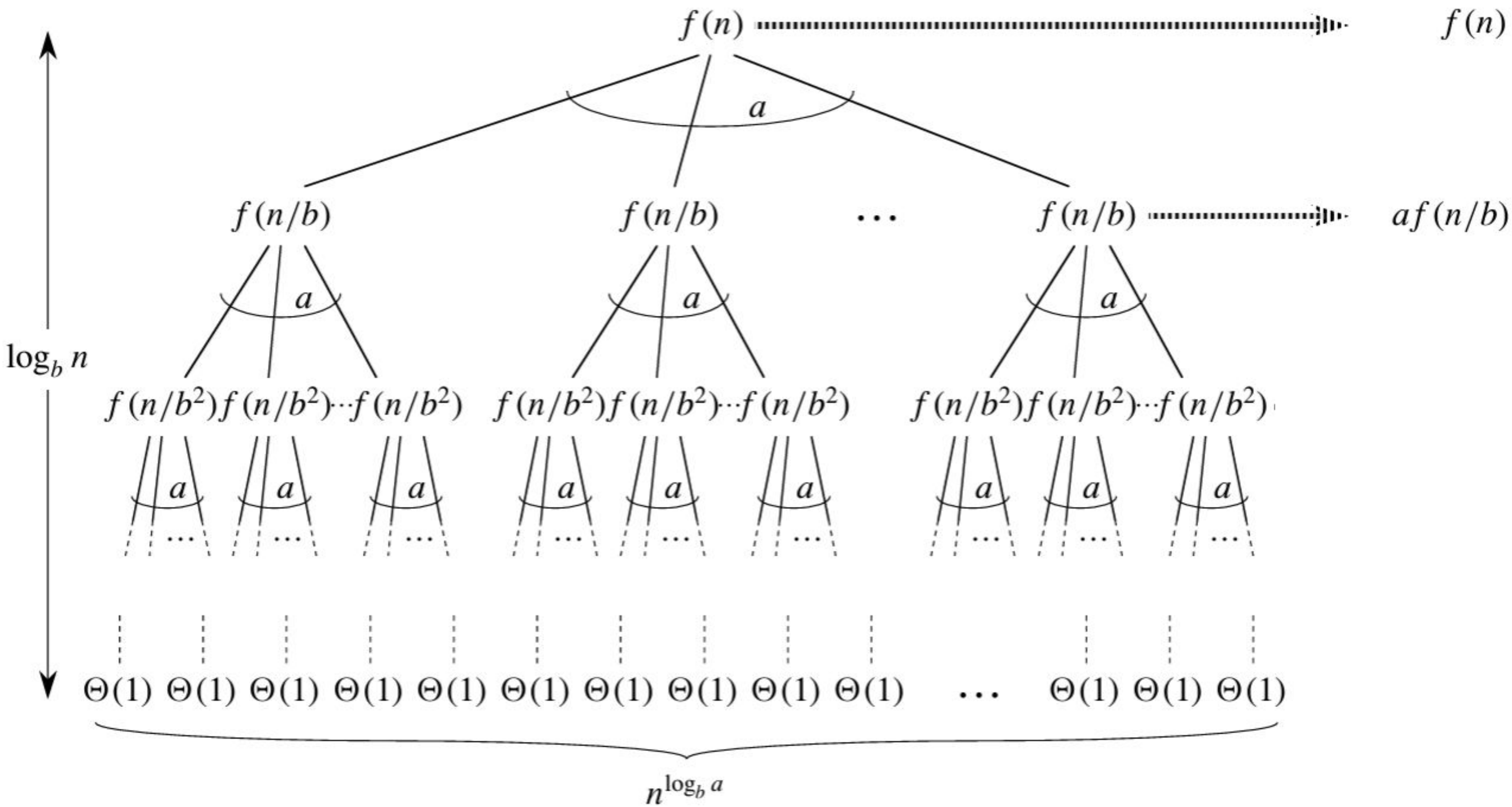
Teorema Mestre - Intuição



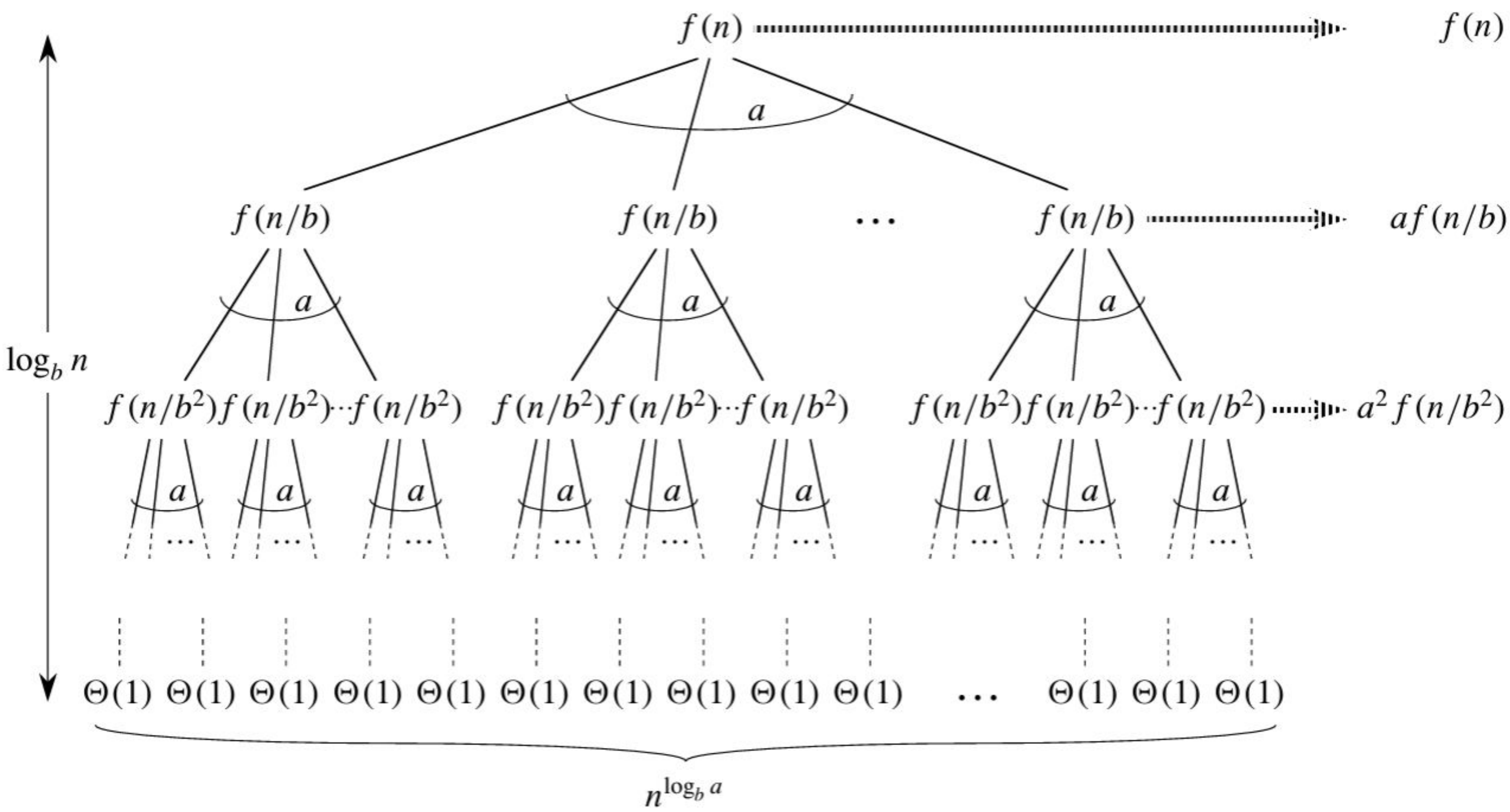
Teorema Mestre - Intuição



Teorema Mestre - Intuição

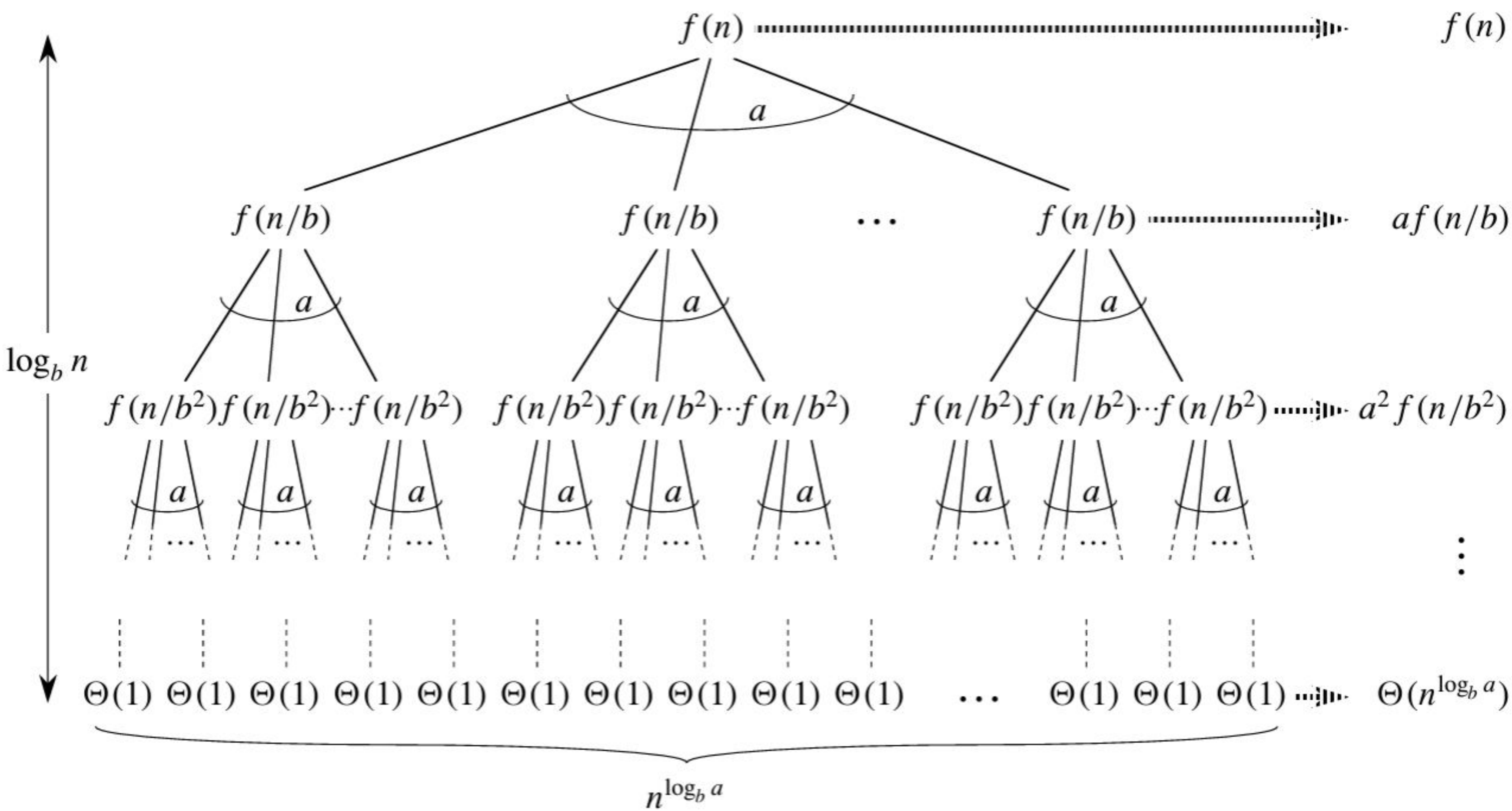


Teorema Mestre - Intuição

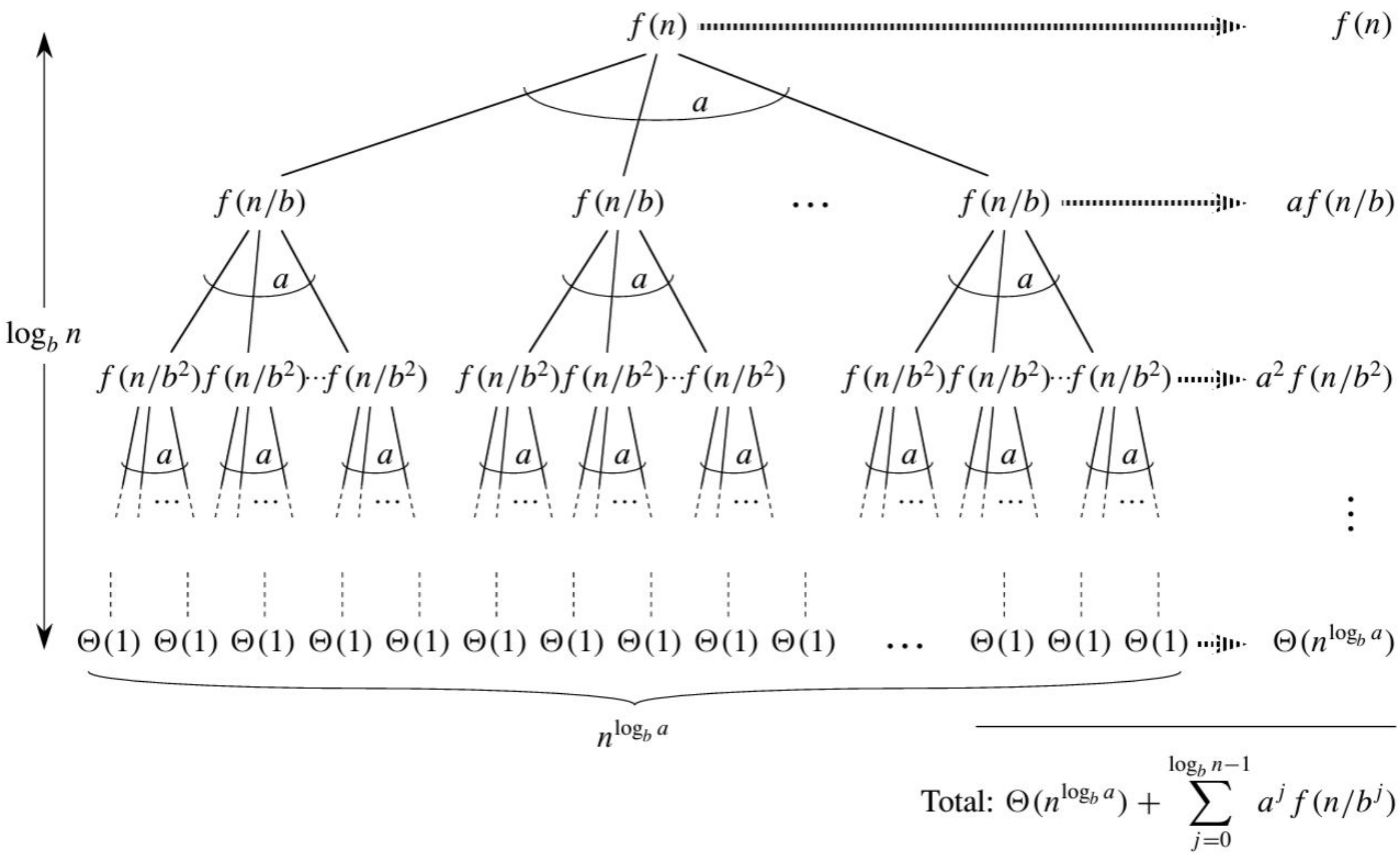


DCC

Teorema Mestre - Intuição



Teorema Mestre - Intuição



Teorema Mestre - Intuição

- A ideia é analisar a relação entre a divisão em sub problemas com o custo de resolvê-los
- O valor $log_b(a)$ codifica de certa forma essa razão.

$$log_b(a) = \frac{ln(a)}{ln(b)}$$

$ln(a) \rightarrow$ Quantidade de subproblemas

$ln(b) \rightarrow$ “Tamanho” do subproblema

Teorema Mestre - Intuição

- Além disso a quantidade de elementos no último nível da nossa recursão também está relacionado com esse valor
- Analisando a soma total do custo basta agora entender qual a relação da função *f(n)* com $n^{log_b a}$.

$$\Theta(n^{log_b a}) + \sum_{j=0}^{lob_b n-1} a^j f\left(\frac{n}{b^j}\right)$$

Teorema Mestre - Intuição

Relembrando:

$$\Theta(n^{log_b a}) + \sum_{j=0}^{lob_b n-1} a^j f\left(\frac{n}{b^j}\right)$$

Teorema Mestre - Intuição

Relembrando:

- O primeiro termo da soma representa o custo da recursão em si.

$$\underline{\Theta(n^{log_b a})} + \sum_{j=0}^{lob_b n-1} a^j f\left(\frac{n}{b^j}\right)$$

Teorema Mestre - Intuição

Relembrando:

- O primeiro termo da soma representa o custo da recursão em si.
- Enquanto o segundo termo contabiliza o custo

$$\Theta(n^{log_b a}) + \underbrace{\sum_{j=0}^{lob_b n-1} a^j f\left(\frac{n}{b^j}\right)}$$

Teorema Mestre - Intuição

Relembrando:

- O primeiro termo da soma representa o custo da recursão em si.
- Enquanto o segundo termo contabiliza o custo

Vamos agora pensar nos vários casos.

$$\Theta(n^{\log_b a}) + \underbrace{\sum_{j=0}^{lob_b n - 1} a^j f\left(\frac{n}{b^j}\right)}$$

Teorema Mestre - Intuição caso 1

- Se a função *f(n)* for limitada superiormente por uma função *g(n)* "menor" que $n^{log_b a}$

$$\Theta(n^{log_b a}) + \sum_{j=0}^{lob_b n - 1} a^j f\left(\frac{n}{b^j}\right)$$

Teorema Mestre - Intuição caso 1

- Se a função *f(n)* for limitada superiormente por uma função *g(n)* "menor" que $n^{log_b a}$
- Como *a* é uma constante, isso significa que o segundo termo é estritamente dominado pelo primeiro

$$\Theta(n^{log_b a}) + \sum_{j=0}^{lob_b n - 1} \underline{a^j} f\left(\frac{n}{b^j}\right)$$

Teorema Mestre - Intuição caso 1

- Dessa forma o segundo termo se torna irrelevante no cálculo da complexidade assintótica.

$$\Theta(n^{log_b a}) + \underline{\sum_{j=0}^{lob_b n-1} a^j f\left(\frac{n}{b^j}\right)}$$

DCC

Teorema Mestre - Intuição caso 1

- Este é o caso em que a divisão dos subproblemas é mais custosa que o passo da conquista.

$$\Theta(n^{log_b a}) + \sum_{j=0}^{lob_b n-1} a^j f\left(\frac{n}{b^j}\right)$$

Teorema Mestre - Intuição caso 2

- Se *f(n)* for limitada por $n^{log_b a}$ tanto inferiormente quanto superiormente.

$$\Theta(n^{log_b a}) + \sum_{j=0}^{lob_b n-1} a^j f\left(\frac{n}{b^j}\right)$$

Teorema Mestre - Intuição caso 2

- Se *f(n)* for limitada por $n^{log_b a}$ tanto inferiormente quanto superiormente.
- Esse caso se assemelha muito com o exemplo *n log(n)* da aula passada.

$$\Theta(n^{log_b a}) + \sum_{j=0}^{lob_b n-1} a^j f\left(\frac{n}{b^j}\right)$$

Teorema Mestre - Intuição caso 2

- Se *f(n)* for limitada por $n^{log_b a}$ tanto inferiormente quanto superiormente.
- Esse caso se assemelha muito com o exemplo *n log(n)* da aula passada.
- Mas agora a soma de cada camada é $\Theta(n^{log_b a})$

$$\Theta(n^{log_b a}) + \sum_{j=0}^{lob_b n-1} a^j f\left(\frac{n}{b^j}\right)$$

Teorema Mestre - Intuição caso 2

- Nesse caso tanto a divisão quanto a conquista tem custos semelhantes.

$$\Theta(n^{log_b a}) + \sum_{j=0}^{lob_b n-1} a^j f\left(\frac{n}{b^j}\right)$$

DCC

Teorema Mestre - Intuição caso 3

- Se a função *f(n)* for limitada inferiormente por uma função *g(n)* “maior” que $n^{log_b a}$

$$\Theta(n^{log_b a}) + \sum_{j=0}^{lob_b n-1} a^j f\left(\frac{n}{b^j}\right)$$

Teorema Mestre - Intuição caso 3

- Se a função *f(n)* for limitada inferiormente por uma função *g(n)* “maior” que $n^{log_b a}$
- Nesse caso o custo de criar os subproblemas é negligenciável.

$$\underline{\Theta(n^{log_b a})} + \sum_{j=0}^{lob_b n - 1} a^j f\left(\frac{n}{b^j}\right)$$

Teorema Mestre - Intuição caso 3

- Se a função *f(n)* for limitada inferiormente por uma função *g(n)* "maior" que $n^{log_b a}$
- Nesse caso o custo de criar os subproblemas é negligenciável.
- O custo depende assintoticamente da função *f(n)*.

$$\Theta(n^{log_b a}) + \underline{\sum_{j=0}^{lob_b n - 1} a^j f\left(\frac{n}{b^j}\right)}$$

Teorema Mestre

- É importante lembrar que isso é apenas uma intuição sobre o que significa o Teorema Mestre, para facilitar seu entendimento.
- De forma alguma pode ser considerada como uma demonstração do teorema, vários detalhes foram omitidos
- Mas essa análise ilustra a ideia geral da demonstração formal.

Teorema Mestre - Enunciado

- Sejam *a≥1* e *b>1* constantes, *f(n)* uma função assintoticamente positiva e *T(n)* uma relação de recorrência definida da forma:

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

- Então*T(n)* é asintoticamente:

Caso 1: Se $f(n) = O(n^{log_b(a-\epsilon)}),\ \epsilon > 0 \implies T(n) = \Theta(n^{log_b(a)})$.

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

Caso 3: Se $\begin{cases} f(n) = \Omega(n^{log_b(a+\epsilon)}), \epsilon > 0 \\ af(n/b) \leq cf(n), \forall n \geq m, c < 1 \end{cases} \implies T(n) = \Theta(f(n))$.

Teorema Mestre

- Nos casos 1 e 3, não basta a função *f(n)* ser maior (ou menor) que $n^{log_b a}$, essa diferença deve ser por um fator $n^\epsilon, \epsilon > 0$.
- Dizemos que *f(n)* é **polinomialmente** maior (ou menor) que $n^{log_b a}$.
- Portanto nem sempre conseguimos aplicar o Teorema Mestre.

Exemplos:

$$T(n) = 9T(n/3) + n$$

Exemplos:

$$T(n) = 9T(n/3) + n$$

Temos:

$a = 9$

Exemplos:

$$T(n) = 9T(n/3) + n$$

Temos:

$a = 9$

$b = 3$

Exemplos:

$$T(n) = 9T(n/3) + n$$

Temos:

$a = 9$

$b = 3$

$f(n) = n$

Exemplos:

$$T(n) = 9T(n/3) + n$$

Temos:

$a = 9$

$b = 3$

$f(n) = n$

$n^{log_b(a)} = n^{log_3(9)} = n^2 = \Theta(n^2)$

Exemplos:

$$T(n) = 9T(n/3) + n$$

Temos:

$a = 9$

$b = 3$

$f(n) = n$

$n^{log_b(a)} = n^{log_3(9)} = n^2 = \Theta(n^2)$

Vamos aplicar o caso 1! Tome $\epsilon = 1$:

Exemplos:

$$T(n) = 9T(n/3) + n$$

Temos:

$a = 9$

$b = 3$

$f(n) = n$

$n^{log_b(a)} = n^{log_3(9)} = n^2 = \Theta(n^2)$

Vamos aplicar o caso 1! Tome $\epsilon = 1$:

$$n^{log_b(a-\epsilon)} = n^{log_3(9-1)} = n^{log_3(8)}$$

Exemplos:

$$T(n) = 9T(n/3) + n$$

Temos:

$a = 9$

$b = 3$

$f(n) = n$

$n^{log_b(a)} = n^{log_3(9)} = n^2 = \Theta(n^2)$

Vamos aplicar o caso 1! Tome $\epsilon = 1$:

$n^{log_b(a-\epsilon)} = n^{log_3(9-1)} = n^{log_3(8)}$

$f(n) = n < n^{log_3(8)} < n^2$

Exemplos:

$$T(n) = 9T(n/3) + n$$

Temos:

$a = 9$

$b = 3$

$f(n) = n$

$n^{log_b(a)} = n^{log_3(9)} = n^2 = \Theta(n^2)$

Vamos aplicar o caso 1! Tome $\epsilon = 1$:

$n^{log_b(a-\epsilon)} = n^{log_3(9-1)} = n^{log_3(8)}$

$f(n) = n < n^{log_3(8)} < n^2$

$f(n) = O(n^{log_b(a-\epsilon)})$

Exemplos:

$$T(n) = 9T(n/3) + n$$

Temos:

$a = 9$

$b = 3$

$f(n) = n$

$n^{log_b(a)} = n^{log_3(9)} = n^2 = \Theta(n^2)$

Vamos aplicar o caso 1! Tome $\epsilon = 1$:

$n^{log_b(a-\epsilon)} = n^{log_3(9-1)} = n^{log_3(8)}$

$f(n) = n < n^{log_3(8)} < n^2$

$f(n) = O(n^{log_b(a-\epsilon)})$

Logo:

$$T(n) = \Theta(n^{log_b(a)}) = \Theta(n^2)$$

Exemplos:

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

DCC

Exemplos:

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

Temos:

$a = 1$

Exemplos:

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

Temos:

$a = 1$

$b = 3/2$

DCC

Exemplos:

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

Temos:

$a = 1$

$b = 3/2$

$f(n) = 1$

Exemplos:

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

Temos:

$a = 1$

$b = 3/2$

$f(n) = 1$

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

Exemplos:

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

Vamos aplicar o caso 2!

Temos:

$a = 1$

$b = 3/2$

$f(n) = 1$

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

Exemplos:

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

Vamos aplicar o caso 2!

$$f(n) = \Theta(n^{log_b(a)}) = \Theta(1)$$

Temos:

$a = 1$

$b = 3/2$

$f(n) = 1$

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

Exemplos:

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

Vamos aplicar o caso 2!

Temos:

$$a = 1$$
$$b = 3/2$$
$$f(n) = 1$$
$$n^{log_b(a)} = n^{log_{3/2}(1)} = n^0 = 1$$

$$f(n) = \Theta(n^{log_b(a)}) = \Theta(1)$$

Logo:

$$T(n) = \Theta(n^{log_b(a)}) = \Theta(n^2)$$

Exemplos:

$$T(n) = 3T(n/4) + nlog(n)$$

Exemplos:

$$T(n) = 3T(n/4) + nlog(n)$$

Temos:

$$a = 3$$

$$b = 4$$

$$f(n) = nlog(n)$$

$$n^{log_b(a)} = n^{log_4(3)} < n$$

Exemplos:

$$T(n) = 3T(n/4) + nlog(n)$$

Temos:

$a = 3$

$b = 4$

$f(n) = nlog(n)$

$n^{log_b(a)} = n^{log_4(3)} < n$

Vamos aplicar o caso 3! Tome $\epsilon = 1$:

$n^{log_b(a+\epsilon)} = n^{log_4(3+1)} = n$

$f(n) = nlog(n) = \Omega(n)$

Exemplos:

$$T(n) = 3T(n/4) + nlog(n)$$

Agora basta mostrar a condição de regularidade.

Exemplos:

$$T(n) = 3T(n/4) + nlog(n)$$

Agora basta mostrar a condição de regularidade.

$$af(n/b) \leq cf(n), \forall n \geq m, c < 1$$

Exemplos:

$$T(n) = 3T(n/4) + nlog(n)$$

Agora basta mostrar a condição de regularidade.

$$af(n/b) \leq cf(n), \forall n \geq m, c < 1$$
$$3f(n/4) \leq cf(n)$$

Exemplos:

$$T(n) = 3T(n/4) + nlog(n)$$

Agora basta mostrar a condição de regularidade.

$$af(n/b) \leq cf(n), \forall n \geq m, c < 1$$
$$3f(n/4) \leq cf(n)$$
$$3(n/4log(n/4)) \leq cnlog(n)$$

Exemplos:

$$T(n) = 3T(n/4) + nlog(n)$$

Agora basta mostrar a condição de regularidade.

$$af(n/b) \leq cf(n), \forall n \geq m, c < 1$$
$$3f(n/4) \leq cf(n)$$
$$3(n/4log(n/4)) \leq cnlog(n)$$

Note que se tomarmos $c = 3/4$ e $m = 1$ a afirmação vale. Logo:

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

Exemplos:

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

Exemplos:

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

Temos:

$$a = 2$$

$$b = 2$$

$$f(n) = nlog(n)$$

$$n^{log_b(a)} = n^{log_2(2)} = n$$

Exemplos:

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

Observe a razão $\frac{f(n)}{n^{log_b(a)}}$

$$\frac{f(n)}{n^{log_b(a)}} = \frac{nlog(n)}{n} = log(n)$$

A função *log(n)* é asintoticamente menor que n^{ε} para toda constante positiva ε.
Logo o Teorema Mestre não pode ser utilizado!