

Projeto e Análise de Algoritmos

Crescimento assintótico de funções

Lehilton Pedrosa

Primeiro Semestre de 2023

Notação assintótica e crescimento de funções

Notação assintótica

- ▶ Vamos expressar complexidade como uma **função em n** .
- ▶ A função mede o número de instruções de um algoritmo.
- ▶ Então, as funções consideradas são sempre positivas.
- ▶ Dependendo do caso, n representa diferentes valores:
 - ▶ **Problemas de precisão arbitrária:** número de bits.
 - ▶ **Problemas em grafos:** número de vértices e/ou arestas
 - ▶ **Problemas com vetores:** tamanho do vetor
 - ▶ **Problemas de busca em textos:** tamanho das strings

Comparação de funções

- ▶ Vamos comparar funções assintoticamente
- ▶ Queremos desprezar as constantes multiplicativas
- ▶ Vamos falar em termos de **ordem** de crescimento

	$n = 100$	$n = 1000$	$n = 10^4$	$n = 10^6$	$n = 10^9$
$\log n$	2	3	4	6	9
n	100	1000	10^4	10^6	10^9
$n \log n$	200	3000	$4 \cdot 10^4$	$6 \cdot 10^6$	$9 \cdot 10^9$
n^2	10^4	10^6	10^8	10^{12}	10^{18}
$100n^2 + 15n$	$1,0015 \cdot 10^6$	$1,00015 \cdot 10^8$	$\approx 10^{10}$	$\approx 10^{14}$	$\approx 10^{20}$
2^n	$\approx 1,26 \cdot 10^{30}$	$\approx 1,07 \cdot 10^{301}$	?	?	?

Notação assintótica e crescimento de funções

- ▶ Definições

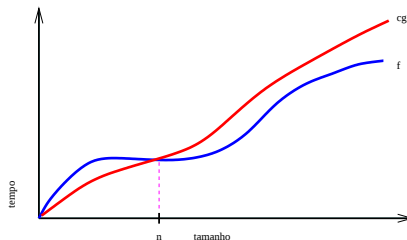
Notação O

Definição

A classe $O(g(n))$ é o conjunto de funções $f(n)$ tais que:

- ▶ existem constantes positivas c e n_0
- ▶ $0 \leq f(n) \leq cg(n)$ para todo $n \geq n_0$

Dizemos que $f(n)$ cresce **no máximo** tão rápido quanto $g(n)$.



Notação O: exemplo

Exemplo

$$\frac{1}{2}n^2 - 3n \in O(n^2)$$

► Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.

► Escolha valores

$$c = \frac{1}{2} \quad \text{e} \quad n_0 = 1.$$

► Então, supondo $n \geq n_0$,

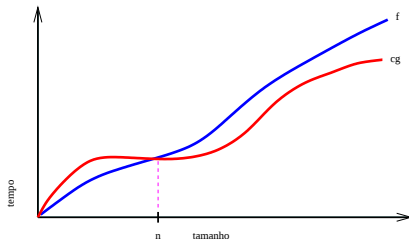
$$\begin{aligned} f(n) &= \frac{1}{2}n^2 - 3n \\ &= \frac{1}{2} \cdot (n^2 - 6n) \\ &\leq \frac{1}{2} \cdot n^2 && \text{(pois } n \geq 0) \\ &= c \cdot g(n). \end{aligned}$$

Definição

A classe $\Omega(g(n))$ é o conjunto de funções $f(n)$ tais que:

- ▶ existem constantes positivas c e n_0
- ▶ $0 \leq cg(n) \leq f(n)$ para todo $n \geq n_0$

Dizemos que $f(n)$ cresce **no mínimo** tão rápido quanto $g(n)$.



Notação Ω : exemplo

Exemplo

$$\frac{1}{2}n^2 - 3n \in \Omega(n^2)$$

- ▶ Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- ▶ Escolha valores

$$c = \frac{1}{4} \quad \text{e} \quad n_0 = 12.$$

- ▶ Então, supondo $n \geq n_0$,

$$\begin{aligned} f(n) &= \frac{1}{2}n^2 - 3n \\ &= \frac{n}{2} \cdot (n - 6) \\ &\geq \frac{n}{2} \cdot \frac{n}{2} && (\text{pois } n \geq 12) \\ &= c \cdot g(n). \end{aligned}$$

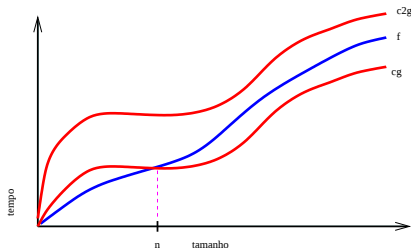
Notação Θ

Definição

A classe $\Theta(g(n))$ é o conjunto de funções $f(n)$ tais que:

- ▶ existem constantes positivas c_1, c_2 e n_0
- ▶ $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ para todo $n \geq n_0$

Dizemos que $f(n)$ cresce **tão rápido** quanto $g(n)$.



Notação Θ : exemplo

Exemplo

$$\frac{1}{2}n^2 - 3n \in \Theta(n^2)$$

- ▶ Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- ▶ Escolha valores

$$c_1 = \frac{1}{4}, \quad c_2 = \frac{1}{2} \quad \text{e} \quad n_0 = 12.$$

- ▶ Então, supondo $n \geq n_0$, já verificamos que

$$c_1 g(n) \leq f(n) \leq c_2 g(n).$$

Definição

A classe $o(g(n))$ é o conjunto de funções $f(n)$ tais que:

- ▶ para **toda** constante $c > 0$, existe um número n_0
- ▶ tal que $0 \leq f(n) < cg(n)$ para todo $n \geq n_0$

Dizemos que $f(n)$ cresce **mais lentamente** que $g(n)$.

Notação o: exemplo

Exemplo

$$1000n^2 \in o(n^3)$$

- ▶ Temos $f(n) = 1000n^2$ e $g(n) = n^3$.
- ▶ Seja $c > 0$ uma constante **arbitrária**
- ▶ Defina $n_0 = \left\lceil \frac{1000}{c} \right\rceil + 1$.
- ▶ Então, supondo $n \geq n_0$,

$$\begin{aligned} f(n) &= 1000n^2 \\ &= \frac{1000}{n} \cdot n^3 \\ &< c \cdot n^3 && \text{(pois } n \geq n_0) \\ &= c \cdot g(n). \end{aligned}$$

Definição

A classe $\omega(g(n))$ é o conjunto de funções $f(n)$ tais que:

- ▶ para **toda** constante $c > 0$, existe um número n_0
- ▶ tal que $0 \leq cg(n) < f(n)$ para todo $n \geq n_0$

Dizemos que $f(n)$ cresce **mais rapidamente** que $g(n)$.

Notação ω : exemplo

Exemplo

$$\frac{1}{1000}n^2 \in \omega(n)$$

- ▶ Temos $f(n) = \frac{1}{1000}n^2$ e $g(n) = n$.
- ▶ Seja $c > 0$ uma constante **arbitrária**.
- ▶ Defina $n_0 = \lceil 1000c \rceil + 1$.
- ▶ Então, supondo $n \geq n_0$,

$$\begin{aligned} f(n) &= \frac{1}{1000}n^2 \\ &= \frac{n}{1000} \cdot n \\ &> c \cdot n && \text{(pois } n \geq n_0) \\ &= c \cdot g(n). \end{aligned}$$

Notação assintótica e crescimento de funções

- Propriedades das notações assintóticas

Equivalências

Condições equivalentes

Sejam $f(n)$ e $g(n)$ funções não negativas, então:

- ▶ $f(n) \in o(g(n))$ se, e somente se, $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$.
- ▶ $f(n) \in O(g(n))$ se, e somente se, $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} < \infty$.
- ▶ $f(n) \in \Theta(g(n))$ se, e somente se, $0 < \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} < \infty$.
- ▶ $f(n) \in \Omega(g(n))$ se, e somente se, $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} > 0$.
- ▶ $f(n) \in \omega(g(n))$ se, e somente se, $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \infty$.

Transitividade

- ▶ Se $f(n) \in O(g(n))$ e $g(n) \in O(h(n))$, então $f(n) \in O(h(n))$.
- ▶ Se $f(n) \in \Omega(g(n))$ e $g(n) \in \Omega(h(n))$, então $f(n) \in \Omega(h(n))$.
- ▶ Se $f(n) \in \Theta(g(n))$ e $g(n) \in \Theta(h(n))$, então $f(n) \in \Theta(h(n))$.
- ▶ Se $f(n) \in o(g(n))$ e $g(n) \in o(h(n))$, então $f(n) \in o(h(n))$.
- ▶ Se $f(n) \in \omega(g(n))$ e $g(n) \in \omega(h(n))$, então $f(n) \in \omega(h(n))$.

Propriedades das classes

Reflexividade

- ▶ $f(n) \in O(f(n))$.
- ▶ $f(n) \in \Omega(f(n))$.
- ▶ $f(n) \in \Theta(f(n))$.

Simetria

- ▶ $f(n) \in \Theta(g(n))$ se, e somente se, $g(n) \in \Theta(f(n))$.

Simetria Transposta

- ▶ $f(n) \in O(g(n))$ se, e somente se, $g(n) \in \Omega(f(n))$.
- ▶ $f(n) \in o(g(n))$ se, e somente se, $g(n) \in \omega(f(n))$.

Regra de l'Hôpital

Regra de l'Hôpital

Sejam f e g funções diferenciáveis tais que

$$\lim_{n \rightarrow \infty} f(n) = \lim_{n \rightarrow \infty} g(n) = \infty.$$

Se o limite de $\frac{f'(n)}{g'(n)}$ existir, então

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \lim_{n \rightarrow \infty} \frac{f'(n)}{g'(n)}.$$

Exercício: descreva as outras formas da Regra de l'Hôpital.

Exemplo

Relacione $f(n) = \ln n$ e $g(n) = \sqrt{n}$ usando classes de funções adequadas.

- ▶ Temos $f'(n) = \frac{1}{n}$ e $g'(n) = \frac{1}{2\sqrt{n}}$
- ▶ Assim,

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \lim_{n \rightarrow \infty} \frac{f'(n)}{g'(n)} = \lim_{n \rightarrow \infty} \frac{2\sqrt{n}}{n} = \lim_{n \rightarrow \infty} \frac{2}{\sqrt{n}} = 0.$$

- ▶ Portanto, $f(n) = o(g(n))$.

Ordenando várias funções

Exemplo

Ordene de acordo com a ordem de complexidade e relacione funções adjacentes nessa ordem.

- ▶ 2^π
- ▶ $\log n$
- ▶ n
- ▶ $n \log n$
- ▶ n^2
- ▶ $100n^2 + 15n$
- ▶ 2^n

Recorrências

Complexidade de MERGE-SORT

MERGE-SORT(A, p, r)

```
1  se  $p < r$   
2     $q \leftarrow \lfloor (p + r)/2 \rfloor$   
3    MERGE-SORT( $A, p, q$ )  
4    MERGE-SORT( $A, q + 1, r$ )  
5    INTERCALA( $A, p, q, r$ )
```

- ▶ Vamos relembrar a complexidade do MERGE-SORT
- ▶ Tamanho da entrada: $n = r - p + 1$
- ▶ Seja $T(n)$ o número de instruções executadas no pior caso

Complexidade de MERGE-SORT

MERGE-SORT(A, p, r)

```
1  se  $p < r$ 
2     $q \leftarrow \lfloor (p + r)/2 \rfloor$ 
3    MERGE-SORT( $A, p, q$ )
4    MERGE-SORT( $A, q + 1, r$ )
5    INTERCALA( $A, p, q, r$ )
```

Linha	Tempo
1	c_1
2	c_2
3	$T(\lceil n/2 \rceil)$
4	$T(\lfloor n/2 \rfloor)$
5	$c_5 n + d_5$

$$\begin{aligned} T(n) &= T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + c_5 n + d_5 + c_1 + c_2 \\ &= T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + an + b \end{aligned}$$

Resolução de recorrências

Obtemos a recorrência:

$$T(n) = \begin{cases} d & \text{se } n = 1 \\ T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + an + b & \text{se } n \geq 2, \end{cases}$$

- ▶ Queremos uma **fórmula fechada** para $T(n)$.
- ▶ Não é necessária a solução exata.
- ▶ Basta encontrar uma função $f(n)$ tal que $T(n) \in \Theta(f(n))$.

Resolução de recorrências

Existem alguns métodos comuns para resolver recorrências:

- ▶ substituição
- ▶ iteração
- ▶ árvore de recorrência

Veremos também o chamado **Teorema Master**

- ▶ aplicável a uma família comum de recorrências
- ▶ fornece uma fórmula fechada diretamente

Recorrências

- ▶ Método da substituição

Método da substituição

Ideia:

1. **adivinhar** uma solução
2. demonstrar que é válida usando indução

Nem sempre é fácil chutar a solução

- ▶ é necessário ter experiência
- ▶ mas vamos obter sugestões com os métodos estudados

Exemplo

Exemplo

Encontre uma fórmula fechada para

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

Chutamos que $T(n) \in O(n \log_2 n)$.

- ▶ Observe que há uma constante escondida na notação O .
- ▶ Para usar indução, precisamos conhecer essa constante.
- ▶ Então chutamos que $T(n) \leq 3n \log_2 n$.

Exemplo: passo da indução

Suponha que a desigualdade vale para $k < n$.

$$\begin{aligned}T(n) &= T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + n \\&\leq 3 \left\lceil \frac{n}{2} \right\rceil \log_2 \left\lceil \frac{n}{2} \right\rceil + 3 \left\lfloor \frac{n}{2} \right\rfloor \log_2 \left\lfloor \frac{n}{2} \right\rfloor + n && \text{(pela h.i.)} \\&\leq 3 \left\lceil \frac{n}{2} \right\rceil \log_2 n + 3 \left\lfloor \frac{n}{2} \right\rfloor (\log_2 n - 1) + n \\&= 3 \left(\left\lceil \frac{n}{2} \right\rceil + \left\lfloor \frac{n}{2} \right\rfloor \right) \log_2 n - 3 \left\lfloor \frac{n}{2} \right\rfloor + n \\&= 3n \log_2 n - 3 \left\lfloor \frac{n}{2} \right\rfloor + n \\&\leq 3n \log_2 n.\end{aligned}$$

Mostramos o passo da indução!

Exemplo: base da indução

Ainda falta a base

- ▶ Temos $T(1) = 1$, mas $3 \cdot 1 \cdot \log_2 1 = 0$
- ▶ A desigualdade não vale para $n = 1$
- ▶ Não temos uma base da indução

Ok, queremos mostrar apenas $T(n) \in O(n \log_2 n)$.

- ▶ Basta mostrar que a desigualdade vale para $n \geq n_0$
- ▶ Vamos escolher $n_0 = 2$

Base da indução:

- ▶ Para $n = 2$ e $n = 3$, temos

$$T(2) = T(1) + T(1) + 2 = 4 \leq 3 \cdot 2 \cdot \log_2 2 = 6$$

$$T(3) = T(2) + T(1) + 3 = 8 \leq 3 \cdot 3 \cdot \log_2 3 \approx 14,26$$

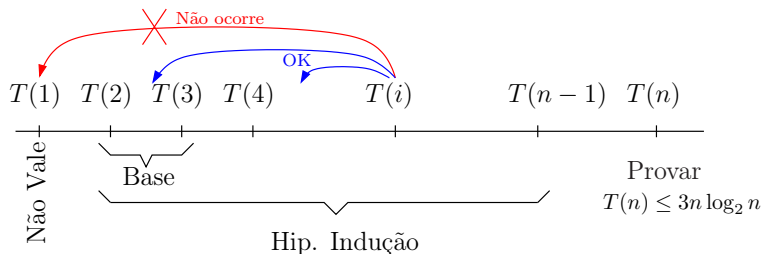
- ▶ Por que precisamos de **dois casos básicos**?

Exemplo: verificando a base

Não podemos ter $k = 1$ quando usamos a hipótese da indução:

$$T(1) = 1$$

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



Modificando um pouco o exemplo

Mas se tivéssemos $T(1) = 8$?

- ▶ A desigualdade não vale para $n = 2$
- ▶ Temos $T(2) = 8 + 8 + 2 = 18$, mas $3 \cdot 2 \cdot \log_2 2 = 6$
- ▶ Podemos escolher uma **constante multiplicativa** maior.
- ▶ Tentando $T(n) \leq 10n \log_2 n$, obtemos

$$T(2) = 18 \leq 10 \cdot 2 \cdot \log_2 2 = 20$$

$$T(3) = 29 \leq 10 \cdot 3 \cdot \log_2 3 \approx 47,55$$

Conclusão:

- ▶ Se o passo da indução vale, então podemos escolher valores adequados para as constantes c e n_0 .

Encontrando as constantes

Suponha que já adivinhamos que $T(n) \in O(n \log_2 n)$

- ▶ Chutamos que $T(n) \leq cn \log_2 n$ para $c = 3$.
- ▶ Depois escolhemos $n_0 = 2$.
- ▶ Como podemos encontrar essas constantes?

Uma maneira possível:

1. Suponha $T(n) \leq cn \log_2 n$ para algum c genérico.
2. Substitua e desenvolva a expressão para $T(n)$.
3. Determine c e n_0 de forma a provar o passo da indução.

Primeira tentativa

$$\begin{aligned}T(n) &= T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + n \\&\leq c \lceil \frac{n}{2} \rceil \log_2 \lceil \frac{n}{2} \rceil + c \lfloor \frac{n}{2} \rfloor \log_2 \lfloor \frac{n}{2} \rfloor + n \\&\leq c \lceil \frac{n}{2} \rceil \log_2 n + c \lfloor \frac{n}{2} \rfloor \log_2 n + n \\&= c \left(\lceil \frac{n}{2} \rceil + \lfloor \frac{n}{2} \rfloor \right) \log_2 n + n \\&= cn \log_2 n + n\end{aligned}$$

Não deu certo...

Segunda tentativa

$$\begin{aligned}T(n) &= T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + n \\&\leq c \left\lceil \frac{n}{2} \right\rceil \log_2 \left\lceil \frac{n}{2} \right\rceil + c \left\lfloor \frac{n}{2} \right\rfloor \log_2 \left\lfloor \frac{n}{2} \right\rfloor + n \\&\leq c \left\lfloor \frac{n}{2} \right\rfloor \log_2 n + c \left\lfloor \frac{n}{2} \right\rfloor (\log_2 n - 1) + n \\&= c \left(\left\lceil \frac{n}{2} \right\rceil + \left\lfloor \frac{n}{2} \right\rfloor \right) \log_2 n - c \left\lfloor \frac{n}{2} \right\rfloor + n \\&= cn \log_2 n - c \left\lfloor \frac{n}{2} \right\rfloor + n \\&\leq cn \log_2 n. \quad (\text{escolhendo } n \geq n_0 \text{ adequado})\end{aligned}$$

- ▶ Para a última desigualdade, queremos $-c \lceil n/2 \rceil + n \leq 0$
- ▶ Basta que $c = 3$ e $n_0 \geq 2$.

Completando o exemplo

- ▶ Já mostramos que $T(n) \in O(n \log_2 n)$.
- ▶ Mas queremos mostrar que $T(n) \in \Theta(n \log_2 n)$.
 - ▶ Resta mostrar $T(n) \in \Omega(n \log_2 n)$.
 - ▶ A prova é similar.
 - ▶ Faça como exercício.

Adivinhando uma solução

Não há receita genérica para adivinhar soluções.

- ▶ A experiência é um fator importante.
- ▶ Normalmente, recorrências similares têm a mesma ordem de crescimento.

Exemplo

Resolva a recorrência

$$T(n) = \begin{cases} 1 & \text{se } n = 1 \\ 2T(\lfloor n/2 \rfloor) + n & \text{se } n \geq 2 \end{cases}$$

- ▶ Ela muito parecida com a anterior.
- ▶ Então chutamos $T(n) \in \Theta(n \log_2 n)$.
- ▶ Mostre isso como exercício.

Adivinhando uma solução (cont)

Exemplo

Resolva a recorrência

$$T(n) = \begin{cases} 1 & \text{se } n = 1 \\ 2T(\lfloor n/2 \rfloor + 17) + n & \text{se } n \geq 2 \end{cases}$$

- ▶ Ela parece mais difícil por causa do termo 17.
- ▶ Intuitivamente, parece que $T(\lfloor n/2 \rfloor) \approx T(\lfloor n/2 \rfloor + 17)$
- ▶ Chutamos que $T(n) \in \Theta(n \log_2 n)$.
- ▶ Mostre isso como exercício.

Algumas vezes é necessário fortalecer a hipótese de indução.

Exemplo

Resolva a recorrência

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

Chutamos que $T(n) \leq cn$ para alguma constante c .

$$\begin{aligned} T(n) &= T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + 1 \\ &\leq c\lceil n/2 \rceil + c\lfloor n/2 \rfloor + 1 \\ &= cn + 1. \end{aligned}$$

- ▶ Não deu certo
- ▶ Mas de fato $T(n) \in \Theta(n)$!

Fortalecendo a hipótese

- ▶ Vamos fazer uma hipótese de indução um pouco diferente.
- ▶ Vamos mostrar que $T(n) \leq cn - b$ para $c > 0$ e $b > 0$

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

- ▶ Para a última de desigualdade, basta escolher $b \geq 1$.
- ▶ Isso mostra o passo indutivo.

Recorrências

- ▶ Método da iteração

Método da iteração

Idéia:

1. expandir a recorrência iterativamente
2. reescrevê-la como uma somatória em termos de n

Não é necessário adivinhar a resposta!

- ▶ Mas é um pouco mais trabalhoso
- ▶ É útil conhecer limitantes de somatórias
- ▶ Nem sempre é trivial encontrar um termo geral da soma

Exemplo

Exemplo

Encontre uma fórmula fechada para

$$T(n) = \begin{cases} b & \text{se } n \leq 3 \\ 3T(\lfloor n/4 \rfloor) + n & \text{se } n \geq 4 \end{cases}$$

Iterando a recorrência

$$\begin{aligned} T(n) &= n + 3T(\lfloor n/4 \rfloor) \\ &\leq n + 3(n/4 + 3T(\lfloor n/16 \rfloor)) \\ &\leq n + 3(n/4 + 3(n/16 + 3T(\lfloor n/64 \rfloor))) \\ &\leq n + 3n/4 + 9n/16 + 27T(\lfloor n/64 \rfloor). \end{aligned}$$

Queremos

- ▶ identificar o termo geral da soma: $3^i n/4^i$
- ▶ e o argumento da função: $\lfloor n/4^k \rfloor$

Encontrando a função

Quantas vezes iteramos?

- ▶ suponha que iteramos k vezes
- ▶ paramos quando $\lfloor n/4^k \rfloor \leq 3$, i.e., quando $k + 1 > \log_4 n \geq k$

A soma completa é

$$\begin{aligned} T(n) &\leq \left(n + 3n/4 + 9n/16 + \dots + 3^{k-1}n/4^{k-1} \right) + 3^k b \\ &\leq \left(n + 3n/4 + 9n/16 + 27n/64 + \dots \right) + 3^{\log_4 n} b \\ &= n \sum_{i=0}^{\infty} \left(\frac{3}{4} \right)^i + n^{\log_4 3} b \\ &= 4n + n^{\log_4 3} b. \end{aligned}$$

Como $\log_4 3 < 1$, concluímos que $T(n) \in O(n)$.

Simplificando

Podemos supor que $n = 4^k$

- ▶ As contas ficam mais simples
- ▶ Não prova o caso geral
- ▶ Mas obtemos um bom chute
- ▶ E verificamos com o método da substituição

Exercício: verifique que $T(n) = \Theta(n)$.

Recorrências

- ▶ Método da árvore de recorrência

Árvore de recorrência

Idéia:

1. crie uma árvore de recorrência:
 - ▶ os nós representam os termos independentes
 - ▶ os filhos representam as subfunções recorrentes
2. somamos os termos de cada nível da árvore
3. depois somamos todos os níveis

Vantagens

- ▶ útil quando há vários termos recorrentes
- ▶ é mais fácil organizar as contas

Exemplo

Exemplo

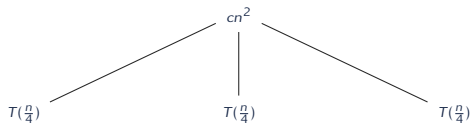
Encontre uma fórmula fechada para

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

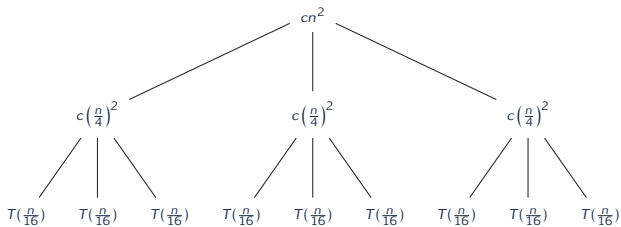
Simplificando

- ▶ Queremos obter uma fórmula fechada apenas
- ▶ De novo, supomos que $n = 4^k$
- ▶ Depois, verificamos com o método da substituição

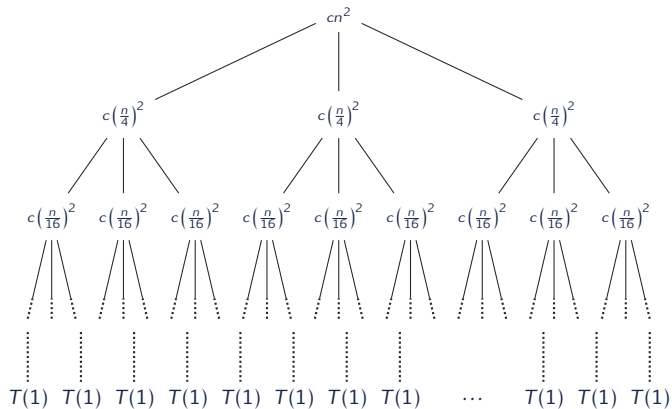
Árvore de recorrência



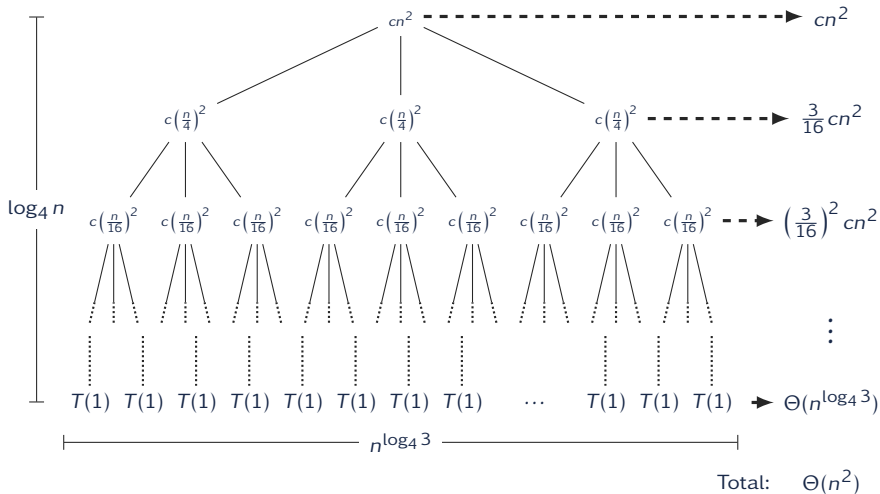
Árvore de recorrência



Árvore de recorrência



Árvore de recorrência



Somando os termos

- ▶ A árvore tem altura $\log_4 n$
- ▶ A soma de cada nível interno é $\left(\frac{3}{16}\right)^i cn^2$
- ▶ O número de **folhas** é $3^{\log_4 n} = n^{\log_4 3}$

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

Concluimos que $T(n) \in O(n^2)$.

Outro exemplo

Exemplo

Encontre uma fórmula fechada para

$$T(n) = \begin{cases} 1 & \text{se } n \leq 2 \\ T(\lceil n/3 \rceil) + T(\lfloor 2n/3 \rfloor) + n & \text{se } n \geq 3 \end{cases}$$

Esse exemplo é um pouco mais complicado

- ▶ Os termos recorrentes são diferentes
- ▶ A árvore não será completa
- ▶ Vamos **mostrar juntos** que $T(n) \in O(n \log_2 n)$

Recorrências

- ▶ Recorrências e notação assintótica

Recorrências com notação assintótica

- ▶ Escrevemos

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

- ▶ Para representar a **família de recorrências**

$$T(n) = \begin{cases} a & \text{se } n \leq 2 \\ 3T(\lfloor n/4 \rfloor) + f(n) & \text{se } n \geq 3 \end{cases}$$

onde a é uma constante e $f(n)$ está em $\Theta(n^2)$

- ▶ Todas elas têm a mesma solução assintótica
- ▶ A mesma coisa vale as outras classes de função

Cuidados com a notação assintótica

Exemplo

$$T(n) = \begin{cases} 1 & \text{se } n = 1 \\ 2T(\lfloor n/2 \rfloor) + n & \text{se } n \geq 2, \end{cases}$$

- ▶ O uso descuidado da notação assintótica leva a **erros**
- ▶ Já sabemos que $T(n) = \Theta(n \log n)$
- ▶ Mas vamos “provar” que $T(n) = O(n)$!

Cuidados com a notação assintótica (cont)

Vamos mostrar que $T(n) \leq cn$ para alguma constante $c > 0$.

$$\begin{aligned}T(n) &= 2T(\lfloor n/2 \rfloor) + n \\&\leq 2c\lfloor n/2 \rfloor + n \\&\leq cn + n \\&= O(n) \quad \leftarrow \text{ERRADO!}\end{aligned}$$

Qual é o erro?

- ▶ A nossa afirmação é que $T(n) \leq cn$
- ▶ Mas mostramos que $T(n) \leq (c + 1)n$
- ▶ Não concluímos o passo indutivo

Recorrências

- ▶ Teorema Master

Teorema Master

Veremos um teorema para resolver recorrências da forma

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

- ▶ os valores a e b são constantes
- ▶ também vale se substituirmos n/b por $\lfloor n/b \rfloor$ ou $\lceil n/b \rceil$
- ▶ o teorema **não** se aplica a todas recorrências dessa forma

Teorema Master

Teorema Master

Sejam $a \geq 1$ e $b > 1$ constantes, seja $f(n)$ uma função e seja $T(n)$ definida para os inteiros não negativos pela relação de recorrência

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

Então $T(n)$ tem os seguintes limitantes assintóticos:

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

Exemplos de Recorrências

Exemplos para os quais Teorema Master se aplica

► Caso 1:

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

$$T(n) = 4T(n/2) + n \log n$$

► Caso 2:

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

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

► Caso 3:

$$T(n) = T(3n/4) + n \log n$$

Exemplos de Recorrências

Exemplos para os quais Teorema Master não se aplica

- ▶ $T(n) = T(n - 1) + n$
- ▶ $T(n) = T(n - a) + T(a) + n, (a \geq 1 \text{ inteiro})$
- ▶ $T(n) = T(\alpha n) + T((1 - \alpha)n) + n, (0 < \alpha < 1)$
- ▶ $T(n) = T(n - 1) + \log n$
- ▶ $T(n) = 2T(\frac{n}{2}) + n \log n$