

Projeto e Análise de Algoritmos (PAA)

Análise Assintótica

⊗ Apresentar método baseado em limite!!!

Análise assintótica é um ferramental matemático que

- nos dá vocabulário para descrever a eficiência de algoritmos
- e permite formalizar vários princípios da análise.

Princípios da análise de algoritmos:

- 1 • análise de pior caso,
- 2 • pouca atenção para constantes multiplicativas e termos de menor ordem,
- 3 • foco no comportamento dos algoritmos para entradas grandes.

Ela é simples o bastante para ignorar aspectos sobre os quais

- não temos controle ou interesse quando analisamos um algoritmo, como
 - detalhes de implementação, linguagem, compilação e arquitetura.

Mas é precisa o bastante para diferenciar algoritmos,

- e permitir escolhas embasadas ao resolver um problema,
 - ~~◦ particularmente quando as entradas são grandes.~~

Notação O

$$T(n) \in O(f(n)) \Leftrightarrow T(n) \text{ é } O(f(n)) \Leftrightarrow T(n) = O(f(n))$$

Omicron ou big O

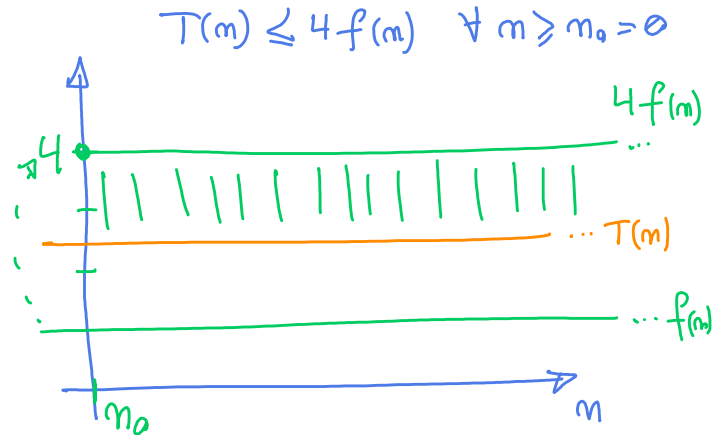
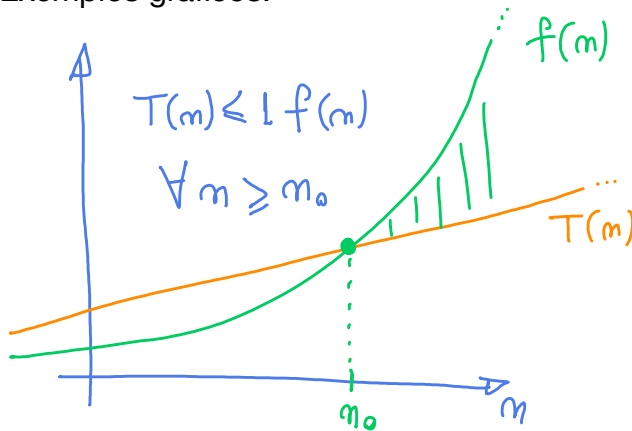
Considere duas funções T e f , definidas nos inteiros positivos, i.e., $n \in \mathbb{Z}^+$

" $\leq$ assintótico"

- $T(n) \in O(f(n))$ significa que, para um n suficientemente grande,
 - uma constante vezes $f(n)$ limita superiormente $T(n)$.

- A intuição é que, assintoticamente, T cresce no máximo tão rápido quanto f .

Exemplos gráficos:



Definição formal:

- $T(n) = O(f(n))$ se existem constantes $C, n_0 > 0$ tais que
 - $T(n) \leq C \cdot f(n)$ para todo $n > n_0$
- Note que C e n_0 não podem depender de n .

$$T(n) \leq C f(n), n \geq n_0$$



$$\frac{T(n)}{f(n)} \leq C, n \geq n_0 \Leftrightarrow \lim_{n \rightarrow +\infty} \frac{T(n)}{f(n)} < +\infty$$

Deixar escrito no quadro

Definição formal:

- $T(n) = O(f(n))$ se existem constantes n_0 e $c > 0$
 - tais que $T(n) \leq c \cdot f(n)$ para todo $n \geq n_0$.
- Note que n_0 e c não podem depender de n .

$$T(n) = O(f(n)) \text{ se}$$

$$\exists c \in \mathbb{Z}^+ \wedge n_0 \in \mathbb{Z}^+ :$$

$$T(n) \leq c f(n) \quad \forall n \geq n_0$$

$$T(n) = 10n \lg n + 10n \text{ é } O(n \lg n)?$$

Exemplo 1: $T(n) = 10n \lg n + 10n$ é $O(n \lg n)$? *Sim*

Prova: Queremos encontrar constantes c e n_0 tais que,

- para todo $n \geq n_0$ temos $T(n) = 10n \lg n + 10n \leq c \cdot n \lg n$

Para tanto, vamos manipular $T(n)$ usando limitantes superiores

$$T(n) = 10n \lg n + 10n \leq 10n \lg n + 10n \lg n = 20n \lg n$$

$\lg n \geq 1 \quad \text{p/ } n \geq 2 = n_0$
 $c = 20$

Como, para $c=20$ e $n_0=2$ temos $T(n) \leq c n \lg n$, então $T(n) \in O(n \lg n)$

Quiz: $T(n) = n \in O(2^n)$? *Sim*, $n \leq c \cdot 2^n$ p/ $n \geq n_0$? *Por indução*

Bem: $n_0 = 0 \Rightarrow n_0 = 0 \leq 1 = c \cdot 2^0$ p/ $c \geq 1$ | H.I.: p/ $n-1 \geq n_0$ temos $n-1 \leq c \cdot 2^{n-1}$

Passo: $n = (n-1) + 1 \stackrel{H.I.}{\leq} c \cdot 2^{n-1} + 1 \stackrel{n-1 \geq 0}{\leq} c \cdot 2^{n-1} + c \cdot 2^{n-1} = c \cdot 2^n$ *Esgr usando limite*

n	2^n
0	1
1	2
2	4
3	8

Exemplo 2: $T(n) = a_k n^k + a_{k-1} n^{k-1} + \dots + a_1 n + a_0$ é $O(n^k)$?

$$C_0 T(n) = a_k n^k + a_{k-1} n^{k-1} + \dots + a_1 n + a_0 = \sum_{i=0}^k a_i n^i \text{ é } O(n^k)? \text{ Sim}$$

Prova: Queremos encontrar constantes C e n_0 tais que,

- para todo $n \geq n_0$ temos $T(n) = \sum_{i=0}^k a_i n^i \leq n^k$
- Para tanto, vamos manipular $T(n)$ usando limitantes superiores

$$T(n) = \sum_{i=0}^k a_i n^i \leq \sum_{i=0}^k |a_i| n^i \leq \sum_{i=0}^k |a_i| n^k = n^k \sum_{i=0}^k |a_i| = \underbrace{(|a_k| + \dots + |a_1|)}_C \cdot n^k$$

$\leftarrow \text{para } n \geq 1$

- Como, para $C = |a_k| + \dots + |a_1|$ e $n_0 = 1$
 - temos $T(n) \leq C n^k$, concluímos que $T(n) \in O(n^k)$

$$T(n) = n^k \notin O(n^{k-1})$$

Exemplo 3: $T(n) = n^k$ não é $O(n^{k-1})$. Queremos mostrar que $\nexists C$ e $n_0 : T(n) \leq C n^{k-1}$

Prova: Por contradição, vamos supor que $T(n) = n^k$ é $O(n^{k-1})$, ou seja,

- supomos que existem C e n_0 tais que, para $n \geq n_0$
 - temos $T(n) = n^k \leq C n^{k-1}$
- Como $n \geq 0$ temos $n^k \leq C n^{k-1} \Rightarrow n \leq C$

- Como o resultado deve valer para todo $n \geq n_0$
 - $C \geq n$ não é constante, o que é um absurdo. ✓

ou seja, mostramos que $\nexists C$ constante que satisfaça a condição

Exemplo 4: $T(n) = 2^{n+10}$ é $O(2^n)$?

$$T(n) = 2^{n+10} \in O(2^n)? \text{ Sim}$$

Prova: Queremos encontrar constantes c e n_0 tais que,

- para todo $n \geq n_0$ temos $T(n) = 2^{n+10} \leq c \cdot 2^n$

$$T(n) = 2^{n+10} = \underbrace{2^{10}}_c \cdot 2^n$$

- Como para $c = 2^{10}$ e $n_0 = 0$ temos $T(n) \leq c \cdot 2^n$

- concluimos que $T(n) \in O(2^n)$

$$T(n) = 2^{10n} \notin O(2^n)$$

Exemplo 5: $T(n) = 2^{10n}$ não é $O(2^n)$.

Queremos mostrar que $\nexists c \text{ e } n_0 : T(n) \leq c \cdot 2^n$

Prova: Por contradição, vamos supor que $T(n) = 2^{10n}$ é $O(2^n)$, ou seja,

$$\forall n \geq n_0$$

- supomos que existem c e n_0 tais que, para $n \geq n_0$

- temos $T(n) = 2^{10n} \leq c \cdot 2^n$

- Como $2^{a \cdot b} = (2^a)^b$ temos $T(n) = 2^{10n} = (2^n)^{10} = 2^n \cdot 2^{9n} \leq c \cdot 2^n \Rightarrow$

$$\Rightarrow 2^{9n} \leq c$$

- Como o resultado deve valer para todo $n \geq n_0$

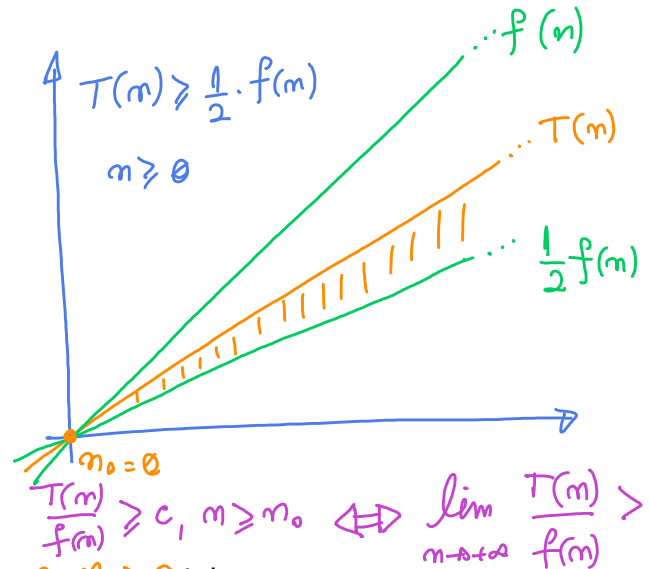
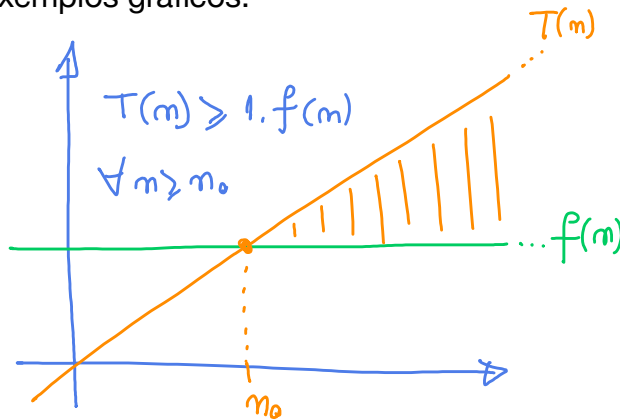
- $c \geq 2^{9n}$ não é constante, o que é uma contradição. ✓

Notação Omega Ω "≥ assintótico"

Considere duas funções T e f , definidas nos inteiros positivos, i.e., $\forall n \in \mathbb{Z}^+$

- $T(n) \in \Omega(f(n))$ significa que, para um n suficientemente grande,
 - uma constante vezes $f(n)$ limita inferiormente $T(n)$.
- A intuição é que, assintoticamente, T cresce pelo menos tão rápido quanto f .

Exemplos gráficos:



Definição formal:

- $T(n) \in \Omega(f(n))$ se existem constantes $c, n_0 > 0$ tais que
 - $T(n) \geq c f(n)$ para todo $n \geq n_0$
- Note que c e n_0 não podem depender de n .

Quiz: se $g(n) \in O(f(n))$ então $f(n) \in \Omega(g(n))$? Sim, pois $g(n) \leq c f(n) \forall n \geq n_0 \Rightarrow f(n) \geq \frac{1}{c} g(n) \forall n \geq n_0$ times $f(n) \geq \frac{1}{c} g(n)$

Exemplo 1: $T(n) = 2^{n+10}$ é $\Omega(2^n)$?

$T(n) = 2^{n+10} \in \Omega(2^n)$? Sim

Prova: Queremos encontrar constantes c e n_0 tais que,

- para todo $n \geq n_0$ temos $T(n) = 2^{n+10} \geq c 2^n$

$$T(n) = 2^{n+10} = \underbrace{2^{10}}_c \cdot 2^n \geq 2^n$$

- Portanto, para $c = 2^{10}$ e $n_0 = 0$ temos $T(n) \geq c \cdot 2^n$
 $c = 1$

$T(n) = 2^{10n} \in \Omega(2^n)$? Sim

Exemplo 2: $T(n) = 2^{10n}$ é $\Omega(2^n)$?

Prova: Queremos encontrar constantes c e n_0 tais que,

- para todo $n \geq n_0$ temos $T(n) = 2^{10n} \geq c \cdot 2^n$

$$T(n) = 2^{10n} = 2^n \cdot 2^{9n} \geq 2^n$$

$p/n > 0$

- Portanto, para $c = 1$ e $n_0 = 0$ temos $T(n) \geq c \cdot 2^n$

$n^k \in O(2^n)$? * Fazer usando limite

$$n_0^k \leq c \cdot 2^{n_0} \Rightarrow c \geq \frac{n_0^k}{2^{n_0}}$$

Prova por ind. mat.

Base: $n = n_0$

$$n^k = n_0^k = n_0^k \cdot \frac{2^{n_0}}{2^{n_0}} = c \cdot 2^{n_0} = c \cdot 2^n$$

H.I.: Para $n-1 \geq n_0$ temos $(n-1)^k \leq c 2^{n-1}$

Passo: $n^k = ((n-1)+1)^k = (n-1)^k + \binom{k}{1}(n-1)^{k-1} + \dots + 1 \leq c \cdot 2^{n-1} + \dots$
H.I.

$$\leq c \cdot 2^n \text{ se } (n-1)^k \geq \dots$$

$$2(n-1)^k \geq n^k \Leftrightarrow \log 2(n-1) \geq \log n$$

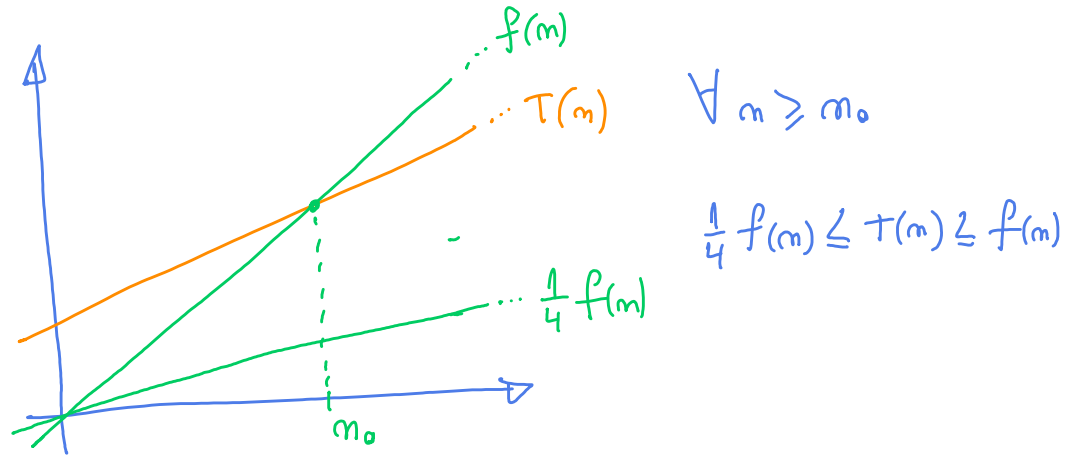
$$2n-2 \geq n \Rightarrow n \geq 2$$

Notação Theta

Considere duas funções T e f , definidas nos inteiros positivos, i.e., $\forall n \in \mathbb{Z}^+$.

- $T(n) \in \Theta(f(n))$ significa que, para um n suficientemente grande,
 - $f(n)$ limita $T(n)$ tanto inferiormente quanto superiormente.
- A intuição é que, assintoticamente, T e f crescem igualmente rápido.

Exemplo gráfico:



Definição formal:

- $T(n) \in \Theta(f(n))$ se existem constantes $c_1, c_2, n_0 > 0$ tais que $c_1 f(n) \leq T(n) \leq c_2 f(n)$ para todo $n \geq n_0$.
- Note que n_0, c_1, c_2 não podem depender de n .

Exemplo 1: $T(n) = \max\{f, g\}$ é $\Theta(f(n) + g(n))$.

$$T(n) = \max\{f(n), g(n)\} \in \Theta(f(n) + g(n))$$

Prova: Queremos encontrar constantes C_1, C_2 e n_0 tais que,

- para todo $n \geq n_0$ temos $C_1(f(n) + g(n)) \leq \max\{f(n), g(n)\} \leq C_2(f(n) + g(n))$
- Buscando primeiro o limitante superior temos

$$\max\{f(n), g(n)\} \leq 1(f(n) + g(n)) \quad f(n) \geq 0 \text{ e } g(n) \geq 0 \quad \forall n \in \mathbb{Z}^+$$

Portanto, para $C_2 = 1$ e $n_0 = 0$ temos $T(n) \leq C_2(f(n) + g(n))$

- Buscando agora o limitante inferior observamos que

$$f(n) + g(n) = \max\{f(n), g(n)\} + \min\{f(n), g(n)\} \leq 2 \max\{f(n), g(n)\}$$

- Assim

$$T(n) = \max\{f(n), g(n)\} \geq \frac{1}{2}(f(n) + g(n))$$

$\Rightarrow T(n) = \frac{1}{2} \cdot n^2 + 3n$ Portanto, para $C_1 = \frac{1}{2}$ e $n_0 = 0$ temos $T(n) \geq C_1(f(n) + g(n))$

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

- $T(n)$ é $O(n)$? Não, pois $\frac{1}{2}n^2 + 3n \leq Cn \Rightarrow C \geq n/2 + 3 \quad \forall n \geq n_0$ (contradição)
- $T(n)$ é $\Omega(n)$? Sim, $\frac{1}{2}n^2 + 3n \geq Cn$ vale p/ $C = 1$ e $n_0 = 0$
- $T(n)$ é $O(n^3)$? Sim, $\frac{1}{2}n^2 + 3n \leq 4n^2 \leq Cn^3$ vale p/ $C = 2$ e $n_0 = 2$
- $T(n)$ é $\Theta(n^2)$? Sim, pois $C_1 \cdot n^2 \leq \frac{1}{2}n^2 + 3n \leq C_2 n^2$ p/ $C_1 = 1/2, C_2 = 4$ e $n_0 = 0$

ou $C_1 = 1/2, C_2 = 2$ e $n_0 = 2$

(Bônus) Exemplo 3: $T(n) = 2^n + 15n^2$ é $\Theta(2^n)$?

Prova: Queremos encontrar constantes c_1 , c_2 e n_0 tais que,

- para todo $n \geq n_0$, temos $c_1 \cdot 2^n \leq T(n) \leq c_2 \cdot 2^n$

- Buscando primeiro o limitante inferior temos

$$T(n) = 2^n + 15n^2 \geq 2^n$$

- Portanto, para $c_1 = 1$ e $n_0 = 0$ temos $c_1 \cdot 2^n \leq T(n)$.

- Buscando agora o limitante superior temos

$$T(n) = 2^n + 15n^2$$

$$\leq 2^n + 2^n \quad (\text{para } n \geq 16, \text{ pois } 2^n \geq 15n^2 \rightarrow$$

$$\rightarrow n \geq \lg(15n^2) \rightarrow n \geq \lg 15 + 2 \lg n \rightarrow$$

$$\rightarrow n - 2 \lg n \geq \lg 15)$$

$$= 2 \cdot 2^n$$

- Portanto, para $c_2 = 2$ e $n_0 = 16$ temos $T(n) \leq c_2 \cdot 2^n$.

Note que, quando demonstramos que uma função é Theta de outra,

- podemos chegar a n_0 s diferentes para os limitantes inferior e superior.

- No final usamos o maior dentre eles.

(Bônus) Notação little omicron

Considere duas funções T e f , definidas nos inteiros positivos.

- $T(n) = \text{littleomicron}(f(n))$ significa que, para qualquer constante $c > 0$,
 - existe $n_0 > 0$ tal que $T(n) \leq c * f(n)$ para todo $n \geq n_0$.
- Intuitivamente, quer dizer que T cresce assintoticamente mais devagar que f .

Exemplo: para qualquer $k \geq 1$ temos $T(n) = n^{(k-1)} = \text{littleomicron}(n^k)$?

Prova: Tome uma constante $c > 0$ arbitrária.

- Queremos encontrar um limitante inferior n_0 para n tal que
$$T(n) = n^{(k-1)} \leq c * n^k \rightarrow 1 \leq c * n \rightarrow n \geq 1/c$$
- Escolhendo $n_0 = 1/c$ temos $T(n) \leq c * n^k$.
- Como c foi escolhido arbitrariamente a prova está concluída.

(Bônus) Notação little omega

Considere duas funções T e f , definidas nos inteiros positivos.

- $T(n) = \text{littleomega}(f(n))$ significa que, para qualquer constante $c > 0$
 - existe $n_0 > 0$ tal que $T(n) \geq c * f(n)$ para todo $n \geq n_0$.
- Intuitivamente, quer dizer que T cresce assintoticamente mais rápido que f .

Exemplo: para qualquer $k \geq 1$ temos $T(n) = n^k = \text{littleomega}(n^{(k-1)})$?

Prova: Tome uma constante $c > 0$ arbitrária.

- Queremos encontrar um limitante inferior n_0 para n tal que
$$T(n) = n^k \geq c * n^{(k-1)} \rightarrow n \geq c$$
- Escolhendo $n_0 = c$ temos $T(n) \geq c * n^{(k-1)}$.
- Como c foi escolhido arbitrariamente a prova está concluída.