INF 112: Programação II

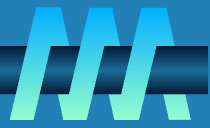
Aula 05

➔ **Introdução à análise de complexidade – parte 2**

Conceitos de complexidade assintótica (notação *O*)

Fábio R. Cerqueira, UFV, DPI, frcerqueira@gmail.com

Taxa de crescimento



Como já visto, a análise de eficiência de um algoritmo se concentra na taxa de crescimento do número de vezes que a operação básica é realizada. Para comparar e ordenar taxas de crescimento, usamos três notações:

- $O(g(n))$: classe de funções que têm crescimento tão ou menos rápido que $g(n)$
- $\Theta(g(n))$: classe de funções que têm crescimento à mesma taxa de $g(n)$
- $\Omega(g(n))$: classe de funções que têm crescimento tão ou mais rápido que $g(n)$

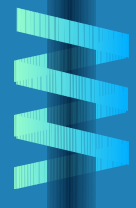
→ Concentrar-nos-emos na notação O.

Taxa de crescimento assintótica

Informalmente, $O(g(n))$ é o conjunto de todas as funções com taxa de crescimento igual ou menor a $g(n)$ (na verdade, igual ou menor a um múltiplo positivo de $g(n)$).

Portanto, as seguintes proposições são todas verdadeiras:

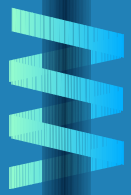
- $n \in O(n^2)$
- $n \in O(n^3)$
- $100n + 5 \in O(n^2)$
- $100n + 5 \in O(n^4)$
- $\frac{1}{2}n(n - 1) \in O(n^2)$



Taxa de crescimento assintótica

Por outro lado:

- $n^3 \notin O(n^2)$
- $0.00001n^3 \notin O(n^2)$
- $n^4 + n + 1 \notin O(n)$



Notação *O*

Definição: Considera-se que uma função $t(n)$ está em $O(g(n))$, denotando-se por $t(n) \in O(g(n))$, se $t(n)$ tiver como limite superior algum múltiplo positivo de $g(n)$ para todo n grande. Em notação matemática, se existe alguma constante c e algum inteiro não-negativo n_0 tal que :

$$t(n) \leq cg(n) \text{ para todo } n \geq n_0 .$$

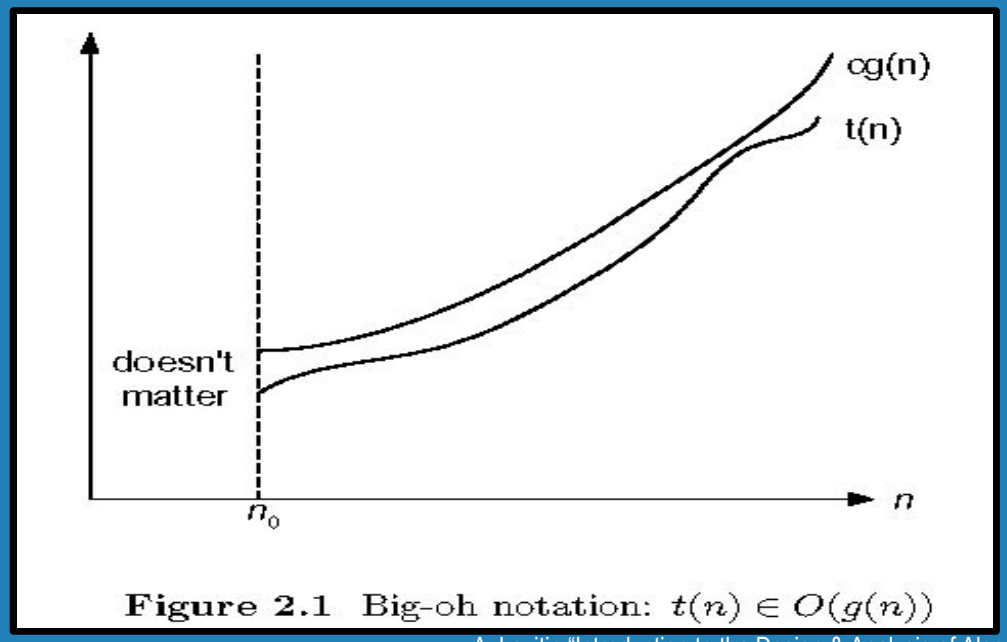


Figure 2.1 Big-oh notation: $t(n) \in O(g(n))$

Notação *O*

- Exemplo: vamos provar formalmente que $100n + 5 \in O(n^2)$:

  $100n + 5 \leq 100n + n$ (para todo $n \geq 5$) $= 101n \leq 101n^2$.

  → Portanto, podemos tomar 101 e 5 como valores para as constantes c e n_0, respectivamente, de modo que:

  $100n + 5 \leq 101n^2, \quad n \geq 5.$

- Note que existem infinitos pares c e n_0. Mas basta mostrar um par possível e já está provado. Poderíamos encontrar outro par assim:

  $100n + 5 \leq 100n^2 + 5 \leq 100n^2 + 5n^2$ (para todo $n \geq 1$) $= 105n^2$

  → Portanto, podemos tomar 105 e 1 como valores para as constantes c e n_0, respectivamente, de modo que:

  $100n + 5 \leq 105n^2, \quad n \geq 1.$

Algumas propriedades da taxa de crescimento assintótica

- $t(n) \in O(t(n))$
- Se $t(n) \in O(g(n))$ e $g(n) \in O(h(n))$, então $t(n) \in O(h(n))$
- Se $t_1(n) \in O(g_1(n))$ e $t_2(n) \in O(g_2(n))$, então:

$$t_1(n) + t_2(n) \in O(\max\{g_1(n), g_2(n)\})$$

Algumas propriedades da taxa de crescimento assintótica

- **No que a terceira propriedade mostrada influi na análise de um algoritmo que contém várias partes a serem executadas (ex.: várias funções)?**
  - **Implica que a eficiência geral do algoritmo será determinada pela parte que apresentar a maior taxa de crescimento, isto é, a parte menos eficiente:**

$$\begin{array}{l} t_1(n) \in O(g_1(n)) \\ t_2(n) \in O(g_2(n)) \\ \ldots \\ t_i(n) \in O(g_i(n)) \end{array} \Rightarrow t_1(n) + t_2(n) + \ldots + t_i(n) \in O(\max\{g_1(n), g_2(n), \ldots, g_i(n)\})$$

Taxa de crescimento de algumas funções importantes

- Toda função logarítmica $\log_a n$ pertence à mesma classe $O(\log n)$ não importando a base do logaritmo (a > 1).
- Todos os polinômios de grau k pertencem à mesma classe:

  $a_k n^k + a_{k-1} n^{k-1} + \ldots + a_0 \in O(n^k)$.
- Funções exponenciais a^n têm taxas de crescimento diferentes para diferentes valores de a.
- As taxas de crescimento a seguir podem ser ordenadas assim:

  $\log n < n^\alpha \ (\alpha>0) < a^n < n! < n^n$.

Classes bem conhecidas de eficiência assintótica

1	constante
$\log n$	logarítmico
n	linear
$n \log n$	n-log-n
n^2	quadrático
n^3	cúbico
2^n	exponential
$n!$	fatorial

- Note que se $t(n) = c$, onde c é alguma constante não-negativa, então, por definição, $t(n) \in O(1)$.

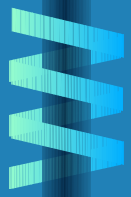
Exercícios

1) Use a notação *O* para indicar, formalmente, a classe de eficiência a que pertence o algoritmo de busca (*SequentialSearch*) dado na aula passada:

   a) No pior caso; b) No melhor caso c) No caso médio

2) Responda se as proposições abaixo são falsas ou verdadeiras. Quando forem verdadeiras, prove sua resposta formalmente.

   a) $n(n+1)/2 \in O(n^3)$ b) $n(n+1)/2 \in O(n^2)$ c) $n(n+1)/2 \in O(n)$



Exercícios

3) Dado um problema, suponha que dois algoritmos que o resolvam tenham o número de operações básicas realizadas, no pior caso, dado por $5n^3 + 2n^2 + 3$ e $10n^3 + 4n^2 + 9$, respectivamente. Pergunta-se:

→ Pode-se afirmar que ambos, para o pior caso, têm a mesma complexidade assintótica?

→ O fato das constantes de seus termos serem diferentes quer dizer o quê?