Lista 6: Notação Grande O

Igor Lacerda

19 de janeiro de 2022

Nota

Nessa lista, eu (tentei) usar LaTeX em todo o documento, mas foi uma experiência nada divertida que se provou extremamente trabalhosa. Nas listas seguintes vou voltar a fazer a parte discursiva a punho mesmo.

Questões Discursivas (QD)

1. A definição de $f(n) \in O(g(n))$ é:

$$f(n) \in O(g(n)) \iff \exists c, n_0 \mid \forall n > n_0 : f(n) \leq c \cdot g(n)$$

2. Se $f_1(n) = O(g_1(n)) \land f_2(n) = O(g_2(n))$, temos que:

$$f_1(n) + f_2(n) = O(\max(g_1(n), g_2(n)))$$

3. A ordem da soma de dois polinômios é a ordem do polinômio de maior grau.

4. Se $f_1(n) = O(g_1(n)) \land f_2(n) = O(g_2(n))$, temos que:

$$f_1(n) \cdot f_2(n) = O(g_1(n) \cdot g_2(n))$$

5. Uma função $f(n)$ pertence a $\Omega(g(n))$ se:

$$f(n) = \Omega(g(n)) \iff g(n) = O(f(n))$$

6. Uma função $f(n)$ pertence a $\Theta(g(n))$ se:

$$f(n) = \Theta(g(n)) \iff g(n) = O(f(n)) \land f(n) = O(g(n))$$

7. É comum ver a notação $O(n)$ sendo usada no lugar da notação $\Theta(n)$, quando se está falando da complexidade de um algoritmo. Essa distinção é importante. Eu poderia dizer que a vasta maioria dos algoritmos conhecidos é $O(n!!)$, mas isso não é verdade para usando a notação teta. Nesse sentido, a notação com $\Theta(n)$ é muito mais informativa. Além disso, é importante usar $\in$ ao invés de $=$.

Exercícios

1. $\models 3x^4 + 1 \in O(x^4/2) \wedge x^4/2 \in O(3x^4 + 1)$

   Pela QD 5, $O(3x^4 + 1) = O(3x^4) = O(x^4) = O(x^4/2)$. Também usando uns princípios básicos, temos $O(x^4/2) = O(x^4) = O(3x^4) = O(3x^4 + 1)$.

2. $\models x \log x \in O(x^2) \wedge x^2 \notin O(x \log x)$

   Suponha $x \geq 1$. Então temos: $\log x \leq x \Rightarrow x \log x \leq x^2$, então para $n_0 = 1 \wedge k = 1$, temos que $x \log(x) \in O(x^2)$. Suponha por contradição que $x^2 \in O(x \log(x))$. Então, para $x \neq 0$, $\exists n_0, c \mid \forall x > n_0, x^2 \leq c \cdot x \log(x) \Rightarrow x \leq c \cdot \log(x) \Rightarrow x/\log(x) \leq c$, o que é impossível pois $\nexists c \mid c \geq x/\log(x) \forall x$.

3. (a) $(n^2 + 8)(n + 1)$

      $O(n^3)$, pois é um polinômio de grau 3.

   (b) $(n \log n + n^2)(n^3 + 2)$

      $O(n^5)$, pois o primeiro fator é $O(n^2)$, pelo exercício 2, o que nos leva a um produto de polinômios, que segue a regra do caso anterior.

   (c) $(n! + 2^n)(n^3 + \log(n^2 + 1))$

      O primeiro fator é $O(n!)$, pela regra da soma. O Segundo fator é $O(n^3)$. Como $n^2 + 1 \approx n^2$ para n grande, podemos supor n^2, o que nos leva a $\log n^2$, ou $2 \log n$, cuja complexidade polinomial é $O(n)$. Então o produto é $O(n^3 \cdot n!)$.

4. (a) $n \cdot \log(n^2 + 1) + n^2 \cdot \log n$

      Assumindo uma definição muito específica para "função simples", aplicamos o raciocínio do item anterior: $O(n^3)$.

   (b) $(n \cdot \log n + 1)^2 + (\log n + 1)(n^2 + 1)$

      $(\log n + 1)(n^2 + 1)$ é $O(\log n \cdot n^2)$. O outro lado é $O(n^2 \cdot \log^2 n)$. Assim, o primeiro lado domina assintoticamente o segundo. O fato é que $\log^2 n$ é $O(n)$, então temos que a função é $O(n^3)$.

   (c) $n^{2^n} + n^{n^2}$

      Essa função é $O(n^{2^n})$, pela regra da soma.

5. *Sejam* $f, g, h : \mathbb{N} \to \mathbb{R}_{\geq 0}$. *Prove a transitividade da notação O(): Se* $f(n) \in O(g(n)) \wedge g(n) \in O(h(n))$ *então* $f(n) \in O(h(n))$

   Temos $\exists a_0, k \mid \forall n > a_0, f(n) \leq k \cdot g(n)$, e $\exists b_0, l \mid \forall n > b_0, g(n) \leq l \cdot h(n)$. Então para $c_0 = \max(a_0, b_0)$, temos que $f(n) \leq k \cdot g(n) \leq k \cdot l \cdot h(n)$, logo $f(n)$ pertence a $O(h(n))$.

6. Isso é permitido pois ao se trabalhar com a notação big-O, multiplicar por constantes não altera a ordem da função. Então, em vez de trabalhar com todos os inteiros, podemos supor que eles são pares.

7. Ora, suponha que 2^n e 3^n são de mesma ordem. Em particular, que $\exists n_0, k \mid \forall n > n_0, 3^n \leq k \cdot 2^n$. Então $(\frac{3}{2})^n \leq k$, o que é absurdo pois estariamos tomando um k arbitrariamente grande.

8. $$O(n \cdot \log n) \subsetneq O(n^{1+\epsilon}) \subsetneq O(n^2/\log n) \subsetneq O((n^2 - n - 100)^4) = O(n^8) \subsetneq O((1+\epsilon)^n)$$

Por ordem, temos:

$$\lim_{n \to \infty} \frac{n \log n}{n^{1+\epsilon}} = \lim_{n \to \infty} \frac{\log n}{n^{\epsilon}} = \lim_{n \to \infty} \frac{1}{n \cdot \epsilon \cdot n^{\epsilon - 1}} = 0$$

Usando L'Hôpital na útlima igualdade. Similarmente:

$$\lim_{n \to \infty} \frac{n^{1+\epsilon}}{\frac{n^2}{\log n}} = \lim_{n \to \infty} \frac{\log n \cdot n^{\epsilon}}{n} = \lim_{n \to \infty} \frac{\log n}{n^{1-\epsilon}} = \lim_{n \to \infty} \frac{1}{(1-\epsilon) \cdot n^{-\epsilon} \cdot n} = 0$$

Felizmente os outros casos não tão trabalhosos. De fato, é óbvio que não tem como $O(n^2/\log n)$ nem chegar perto de $O(n^8)$, pelos graus dos polinômios envolvidos. Mesmo apelando a $O(1/(\log n)) \in O(n)$, chegaríamos num $O(n^3)$, que ainda está distante de $O(n^8)$. Por outro lado, temos dois polinômios de grau 8, então ambos são de ordem n^8. Para completar, no final temos um exponencial com base > 1, que sabemos que cresce mais que qualquer polinômio.

9. **NOOP**

10. **NOOP**