

Notas sobre Crescimento de Função (1)

Anjolina Grisi de Oliveira

1 Notação O grande

Motivação

- Estimar custo de algoritmo em relação ao tempo e espaço.

Definição 1 *Sejam f e g funções do conjunto dos inteiros (ou conjunto dos números reais) no conjunto dos números reais. Dizemos que $f(x)$ é $O(g(x))$ se existem constantes c e n_0 de forma que:*

$$|f(x)| \leq C |g(x)|$$

quando $x > n_0$. Dizemos que $f(x)$ é “O grande de $g(x)$ ”.

- Usamos a notação $f(x) = O(g(x))$ quando temos que $f(x)$ é “O grande de $g(x)$ ”. Isso significa que $f(x)$ é limitada superiormente por $g(x)$. Aqui há uma abuso de notação, pois na realidade $f(x) \in O(g(x))$, pois $O(g(x))$ é o conjunto das funções que g limita superiormente.
- A notação “big-Oh” (“O grande”) é utilizada para expressar formalmente um limite superior para a função. Quando dizemos que $f(x)$ é $O(x^2)$, queremos dizer que f nunca cresce mais que $c \cdot x^2$ a partir de um determinado valor de x e para uma constante c .
- Em termos de algoritmos, quando dizemos que dentre as soluções algorítmicas para um determinado problema, a melhor é $O(x^2)$ em função da entrada, isso significa que se alguém achar um outro algoritmo cuja função para estimar o tempo é $O(\log n)$ significa que a **cota superior** (“upper bound”) para aquele problema diminuiu.

Exemplo 1 *Prove que $f(x) = x^2 + 2x + 1$ é $O(x^2)$.*

$$x^2 + 2x + 1 \leq x^2 + 2x^2 + x^2 \rightarrow x^2 + 2x + 1 \leq 4x^2.$$

Logo, $c = 4$ e $x > 1$ ($n_0 = 1$).

Outra abordagem:

$2x \leq x^2$ para $x > 2$. Logo $x^2 + 2x + 1 \leq x^2 + x^2 + x^2 = 3x^2$. Logo $c = 3$ e $x > 2$.

- Observe que qualquer função com valores maiores que x^2 é cota superior para $f(x)$. Dessa forma, se $f(x)$ é $O(x^2)$, $f(x)$ é também $O(x^3)$, $O(x^4)$, etc.

- Também é verdade que x^2 é $O(x^2 + 2x + 1)$, pois $x^2 \leq x^2 + 2x + 1$, para $x > 1$. Nesse caso, as duas funções são da mesma ordem.
- O monômio de maior grau domina o crescimento de funções polinomiais. Dessa forma podemos ignorar todos os monômios de menor grau. Isso é estabelecido no teorema dado a seguir.

Teorema 1 *Seja $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$, onde $a_0, a_1, \dots, a_{n-1}, a_n$ são números reais. Então $f(n)$ é $O(x^n)$.*

Mais exemplos

Exemplo 2 *Como estimar a soma dos primeiros n inteiros positivos.*

$1 + 2 + 3 + \dots + n \leq n + n + \dots + n = n \cdot n = n^2$. Logo, a soma dos primeiros n inteiros positivos é $O(n^2)$.

Exemplo 3 *Estimando $n!$.*

$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n = n \cdot n \cdot n \cdot \dots \cdot n = n^n$. Logo $n!$ é $O(n^n)$.

Podemos concluir também que $\log n! \leq \log n^n \rightarrow \log n! \leq n \cdot \log n$. Logo, $\log n!$ é $O(n \cdot \log n)$.

Exemplo 4 *Para $n > 1$ temos que $n < 2^n$. Logo, $\log n < \log 2^n$ (usando a base 2 $\rightarrow \log n < n$. Logo $\log n$ é $O(n)$.*

1.1 O crescimento de combinação de funções

Teorema 2 *Suponha que $f_1(x)$ é $O(g_1(x))$ e $f_2(x)$ é $O(g_2(x))$. Então $(f_1 + f_2)(x)$ é $O(\max(|g_1(x)|, |g_2(x)|))$.*

- Se $f_1(x)$ e $f_2(x)$ são ambas $O(g(x))$. Então $(f_1 + f_2)(x)$ é O de que função?

Teorema 3 *Suponha que $f_1(x)$ é $O(g_1(x))$ e $f_2(x)$ é $O(g_2(x))$. Então $(f_1 f_2)(x)$ é $O(g_1(x) \cdot g_2(x))$.*

2 Funções com a mesma ordem de grandeza

Quando duas funções possuem a mesma ordem de grandez usamos a notação “big-theta” (Θ) para expressar essa relação.

Definição 2 *Sejam f e g funções do conjunto dos inteiros ou do conjunto dos números reais no conjunto dos números reais. Dizemos que $f(x)$ é $\Theta(g(x))$ se existem constantes n_0, c_1, c_2 , de forma que se $x > n_0$ então $c_1 g(x) \leq f(x) \leq c_2 g(x)$.*

- É como se as funções $c_1 g(x)$ e $c_2 g(x)$ formassem um envelope em torno dos valores de $f(x)$. Uma mudança nos valores das constantes muda a largura do envelope.

3 Limite inferior

- Usamos a notação “big-omega” (Ω) para denotar um limite inferior ou cota inferior (“*lower bound*”).
- Em termos de algoritmos, significa que às vezes para um determinado problema é possível mostrar que qualquer algoritmo usado requer um mínimo de operações.

Definição 3 *Sejam f e g funções do conjunto dos inteiros (ou conjunto dos números reais) no conjunto dos números reais. Dizemos que $f(x)$ é $\Omega(g(x))$ se existem constantes c e n_0 de forma que:*

$$|f(x)| \leq C |g(x)|$$

quando $x > n_0$. Dizemos que $f(x)$ é “big-Omega de $g(x)$ ”.