Projeto e Análise de Algoritmos - Notas colaborativas

Flávio L. C. de Moura

20 de abril de 2018

Capítulo 1

Fundamentos Matemáticos

Capítulo 2

Notação Assintótica

Definição 1 (O conjunto $O(g)$)**.** *Seja g uma função dos inteiros não-negativos nos reais positivos. Então $O(g)$ é o conjunto das funções (também dos inteiros não-negativos nos reais positivos) tal que existem uma constante real $c > 0$ e uma constante inteira $n_0 > 0$ satisfazendo a desigualdade $f(n) \leq c \cdot g(n), \forall n \geq n_0$. Alternativamente, $O(g(n)) = \{f(n)$: existem constantes positivas c e n_0 tais que $0 < f(n) \leq c \cdot g(n), \forall n \geq n_0$. $\}$*

Definição 2 (O conjunto $\Omega(g)$)**.** *Seja g uma função dos inteiros não-negativos nos reais positivos. Então $\Omega(g)$ é o conjunto das funções (também dos inteiros não-negativos nos reais positivos) tal que existem uma constante real $c > 0$ e uma constante inteira $n_0 > 0$ satisfazendo a desigualdade $c \cdot g(n) \leq f(n), \forall n \geq n_0$. Alternativamente, $\Omega(g(n)) = \{f(n)$: existem constantes positivas c e n_0 tais que $0 < c \cdot g(n) \leq f(n), \forall n \geq n_0$. $\}$*

Definição 3 (O conjunto $\Theta(g)$)**.** *Seja g uma função dos inteiros não-negativos nos reais positivos. Então $\Theta(g)$ é o conjunto das funções (também dos inteiros não-negativos nos reais positivos) tal que existem constantes reais positivas c_1 e $[]_2$ e uma constante inteira $n_0 > 0$ satisfazendo a desigualdade $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n), \forall n \geq n_0$. Alternativamente, $\Theta(g(n)) = \{f(n)$: existem constantes positivas c_1, c_2 e n_0 tais que $0 < c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n), \forall n \geq n_0$. $\}$*

Lema 1. *Uma função $f \in O(g)$ se $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c < \infty$, incluindo o caso em que $c = 0$.*

Lema 2. *Uma função $f \in \Omega(g)$ se $\lim_{n\to\infty} \frac{f(n)}{g(n)} > 0$, incluindo o caso em que o limite é igual a ∞.*

Lema 3. *Uma função $f \in \Theta(g)$ se $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$, para alguma constante $0 < c < \infty$.*

Definição 4 (Complexidade do Pior Caso)**.** *Sejam D_n o conjunto das entradas de tamanho n para o algoritmo em questão, e $I \in D_n$. Seja $t(I)$ o número de operações básicas executadas pelo algoritmo na entrada I. Definimos a função W por*

$$W(n) = \max\{t(I) \mid I \in D_n\}$$

Definição 5 (Complexidade do Caso Médio)**.** *Sejam D_n o conjunto das entradas de tamanho n para o algoritmo em questão, e $I \in D_n$. Sejam $t(I)$ o número de operações básicas executadas pelo algoritmo na entrada I, e $Pr(I)$ a probabilidade da entrada I ocorrer. Definimos a função A por*

$$A(n) = \sum_{I \in D_n} Pr(I) \cdot t(I)$$

Lema 4. *Se $f \in O(g)$ e $g \in O(h)$ então $f \in O(h)$, ou seja O é transitiva. Também são transitivos Ω, Θ, o e ω.*

Lema 5.

1. *$f \in O(g)$ se, e somente se $g \in \Omega(f)$.*
2. *Se $f \in \Theta(g)$ então $g \in \Theta(f)$.*
3. *Θ define uma relação de equivalência sobre as funções. Cada conjunto $\Theta(f)$ é uma classe de equivalência que chamamos de classe de complexidade.*
4. *$O(f + g) = O(\max\{f, g\})$. Equações análogas valem para Ω e Θ. Estas equações são úteis na análise de algoritmos complexos onde f e g podem descrever o trabalho feito em diferentes partes do algoritmo.*

Capítulo 3

Algoritmos de Ordenação

3.1 Ordenação em Tempo Linear

Capítulo 4

Algoritmos em Grafos

Definição 6. *Um **grafo** (não dirigido) G é um par (V, E) onde V é um conjunto finito não-vazio, e E é um conjunto de pares não-ordenados de elementos de V. Um **digrafo** (ou um grafo dirigido) G é um par (V, E) onde V é um conjunto finito não-vazio, e E é uma relação binária sobre V.*