Análise e Projeto de Algoritmos

Aula 2

Thiago Cavalcante – thiago.kun@gmail.com

01 de novembro de 2019

Universidade Federal de Alagoas – UFAL
Campus Arapiraca
Unidade de Ensino de Penedo

Três propriedades de um bom algoritmo:

- Correto
- Eficiente
- Fácil de implementar

Três propriedades de um bom algoritmo:

- Correto
- Eficiente
- Fácil de implementar

Precisamos de ferramentas que sejam independentes de **linguagem** ou **máquina**

- Modelo **RAM** (*Random Access Machine*)
- Análise assintótica de complexidade no pior caso (notação **Big-Oh**)

Modelo RAM

- Operações simples levam **um ciclo de tempo** para serem realizadas (`+ - * / if call`)
- Laços e subrotinas **não são operações simples**, e sim várias operações simples compostas
- Cada acesso à memória leva **um ciclo de tempo** para ser completado e a memória é considerada **infinita**

Simples, porém funciona **muito bem** na prática

Complexidade de **melhor** caso, **pior** caso e caso **médio**

Para entender um algoritmo, precisamos analisar o que acontece em **todas as instâncias**

- **Melhor**: número de ciclos mínimo
- **Médio**: número médio de ciclos de todas as instâncias
- **Pior**: número de ciclos máximo

- **Melhor**: número de ciclos mínimo
- **Médio**: número médio de ciclos de todas as instâncias
- **Pior**: número de ciclos máximo – **Mais útil!**

Cada caso define uma **função de tempo** $\times$ **tamanho do problema**

Usamos a Notação **Big-Oh** para simplificar essas funções

Estamos preocupados apenas com os limites **superior** e **inferior**

Constantes **multiplicativas** são **ignoradas**

$f(n) = 2n$ é igual a $g(n) = n$

$$f(n) = O(g(n))$$

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

$$f(n) = \Theta(g(n))$$

$$3n^2 - 100n + 6 = O(n^2)$$

$$3n^2 - 100n + 6 = O(n^3)$$

$$3n^2 - 100n + 6 \neq O(n)$$

$$3n^2 - 100n + 6 = \Omega(n^2)$$

$$3n^2 - 100n + 6 \neq \Omega(n^3)$$

$$3n^2 - 100n + 6 = \Omega(n)$$

$$3n^2 - 100n + 6 = \Theta(n^2)$$

$$3n^2 - 100n + 6 \neq \Theta(n^3)$$

$$3n^2 - 100n + 6 \neq \Theta(n)$$

$$2^{n+1} = \Theta(2^n)?$$

$$(x + y)^2 = O(x^2 + y^2)?$$

Relações de dominância

As funções são separadas em classes

$f(n) = 0.34n$ e $g(n) = 234234n$ são ambas da classe $\Theta(n)$

Quando $f(n) = O(g(n))$ ou $g(n) = O(f(n))$ (mas não ambos), as classes são diferentes

Quando $f(n) = O(g(n))$, dizemos que g **domina** f (ou $g >> f$)

Funções **constantes**, $f(n) = 1$

Funções **logarítmicas**, $f(n) = \log n$

Funções **lineares**, $f(n) = n$

Funções **superlineares**, $f(n) = n \log n$

Funções **quadráticas**, $f(n) = n^2$

Funções **cúbicas**, $f(n) = n^3$

Funções **exponenciais**, $f(n) = c^n$

Funções **fatoriais**, $f(n) = n!$

$$n! >> c^n >> n^3 >> n^2 >> n \log n >> n >> \log n >> 1$$

Adição

$$O(f(n)) + O(g(n)) \rightarrow O(max(f(n), g(n)))$$

$$\Omega(f(n)) + \Omega(g(n)) \rightarrow \Omega(max(f(n), g(n)))$$

$$\Theta(f(n)) + \Theta(g(n)) \rightarrow \Theta(max(f(n), g(n)))$$

Ex.: $n^3 + n^2 + n + 1 = O(n^3)$

Multiplicação por constante

$O(c \times f(n)) \to O(f(n))$

$\Omega(c \times f(n)) \to \Omega(f(n))$

$\Theta(c \times f(n)) \to \Theta(f(n))$

Obs.: $c > 0$

Multiplicação

$$O(f(n)) * O(g(n)) \to O(f(n) * g(n))$$

$$\Omega(f(n)) * \Omega(g(n)) \to \Omega(f(n) * g(n))$$

$$\Theta(f(n)) * \Theta(g(n)) \to \Theta(f(n) * g(n))$$

Mostre que se $f(n) = O(g(n))$ e $g(n) = O(h(n))$, então $f(n) = O(h(n))$

Exemplos

- *selection sort*
- *insertion sort*
- busca em strings
- multiplicação de matrizes

Logaritmos

- Busca binária
- Árvores
- Bits
- Exponenciação
- Exponenciação rápida

Propriedades dos Logaritmos

- Bases principais: 2, e, 10
- $\log_a(xy) = \log_a(x) + \log_a(y)$
- $\log_a b = \frac{\log_c b}{\log_c a}$
- A base do logaritmo não tem muita influência no crescimento da função
- O logaritmo de qualquer função polinomial é $\Theta(\log n)$
- log(fatorial) é uma soma

Quantas buscas devem ser feitas na lista telefônica se dividirmos ela em 1/3 e 2/3?