

Notas de Aula 3 - Algoritmos e Estrutura de Dados 2 (AE43CP) – Complexidade de Algoritmos (parte 3)

Prof. Jefferson T. Oliva

### **Resumo da aula anterior**

Na aula anterior foi abordada a análise de complexidade algoritmos aplicada em algoritmos iterativos.

Essa análise pode ser feita em algoritmos em recursivos? Sim, mas de forma um pouco diferente.

O exemplo apresentado no slide 3 é obviamente  $O(n)$ , pois o código não passa de um disfarce de algoritmo iterativo. Desse modo, para analisar o algoritmo, basta fazer isso como tal (algoritmo iterativo): slide 4. Assim, o exemplo apresentado no slide 3 é um exemplo de má aplicação da recursividade.

Em muitos casos (até para recursividade mal empregada), é difícil transformar a sub-rotina em iteração, sendo necessário o uso de recorrência para analisar tais algoritmos.

### **Recorrência**

Recorrência é uma função matemática que descreve uma função em termos de seu valor em entradas anteriores da mesma função. Essas funções são comumente aplicadas para a análise de algoritmos de divisão-e-conquista.

Exemplo: sequência de Fibonacci (slide 8).

Para a análise de recorrência, duas situações devem ser definidas:

- Caso base: com tempo de execução constante, em algoritmo recursivo, o caso base ocorre quando um critério de parada é atingido
- Caso indutivo: com o tempo de execução em função do tamanho do problema, esse caso ocorre quando o teste para caso base falha.

**Ver slides 9 e 13.**

### **Resolução de Recorrências**

Diversas técnicas podem ser aplicadas para a resolução de recorrências, por exemplo:

- Método da substituição: é um método que requer “experiência” do analista
- Método da iteração: forma do método da substituição
- Método da árvore de recursão
- Método mestre

**Ver slides de 16 a 32.**

Obs.: quando se chega a uma solução que consiste em um limite assintótico logarítmico não interessa a base do logaritmo, pois se uma função  $f(n)$  pertence a  $O(\log_3 n)$  (base 3), podemos dizer que a complexidade é  $O(\lg n)$ . Logo, a recorrência apresentada no slide 27 pode ser denotada como  $O(n \lg n)$

## Referências

Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; Clifford, S. Algoritmos: teoria e prática. Elsevier, 2012.

Horowitz, E., Sahni, S. Rajasekaran, S. Computer Algorithms. Computer Science Press, 1998.

Rosa, J. L. G. Análise de Algoritmos - parte 1. SCC-201 - Introdução à Ciência da Computação II. Slides. Ciência de Computação. ICMC/USP, 2016.

Szwarcfiter, J.; Markenzon, L. Estruturas de Dados e Seus Algoritmos. LTC, 2010.

Ziviani, M. Projetos de Algoritmos: com implementações em Pascal e C. Thomson, 2004.