

# Exercícios - Recorrências: Provando Soluções

## Análise de Algoritmos - DINF - UFPR

Prove os exercícios por indução, encontrando os valores de  $c$  e  $n_0$ .

**Exercício 1.** Prove que

$$T(n) = \begin{cases} 1 & \text{se } n = 1 \\ T(\lceil n/2 \rceil) + 1 & \text{se } n > 1 \end{cases}$$

é  $O(\log n)$ .

**Exercício 2.** Prove que

$$T(n) = \begin{cases} 8 & \text{se } n = 1 \\ 2T(\lfloor n/2 \rfloor) + n & \text{se } n > 1 \end{cases}$$

é  $\Theta(n \log n)$ .

**Exercício 3.** Prove que

$$T(n) = \begin{cases} 1 & \text{se } n = 1 \\ 2T(\lfloor n/2 \rfloor) + 17 + n & \text{se } n > 1 \end{cases}$$

é  $\Theta(n \log n)$ .

**Exercício 4.** Considere o seguinte problema de busca:

**Entrada:** Uma seqüência de  $n$  números  $A = (a_1, a_2, \dots, a_n)$  e um valor  $v$ .

**Saída:** Um índice  $i$  tal que  $v = A[i]$  ou o valor especial NIL se  $v$  não aparece em  $A$ .

Note que, se o vetor  $A$  está ordenado, podemos comparar o elemento  $A[\lfloor n/2 \rfloor]$  com  $v$ , eliminando metade do vetor em comparações futuras. Tal procedimento é chamado de **Busca Binária**. Pede-se

- Escreva um pseudo-código **recursivo** para a **Busca Binária**
- Escreva a relação de recorrência do seu algoritmo.
- Prove que o tempo execução de pior-caso é  $\Theta(\log n)$ .

**Exercício 5.** Observe que o laço interno do **InsertionSort** usa uma busca linear para varrer (de trás para frente) o subvetor ordenado  $A[1..j-1]$ . Podemos usar a **Busca Binária** (do Exercício 4) para tentar melhorar o pior-caso do **InsertionSort** para  $\Theta(n \log n)$ ?