

Tarefa 3

SCC-601 – Introdução à Ciência da
Computação II

Tarefa 3 (para entrega no fim da aula)

a) Resolva a seguinte recorrência:

$$F(n) = F(n - 1) + n \quad \text{para todo } n \geq 2.$$

$$F(1) = 1$$

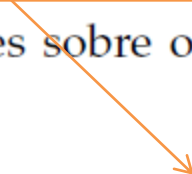
b) Considere que um algoritmo recursivo resolve um dado problema seguindo $F(n)$. Qual a complexidade do algoritmo em termos da notação Big-Oh (ou Big-Theta)?

Resolução

A recorrência pode ser facilmente “desenrolada”: $F(n) = F(n-1) + n = F(n-2) + (n-1) + n = \dots = F(1) + 2 + 3 + \dots + (n-1) + n = 1 + 2 + 3 + \dots + (n-1) + n$. Concluimos assim que

$$F(n) = \frac{1}{2}(n^2 + n) \quad (3.4)$$

para todo $n \geq 1$. (Não temos informações sobre o valor de $F(0)$.) Portanto, F está em $\Theta(n^2)$.



Tente provar, por indução, que essa expressão é válida

Fonte: <http://www.ime.usp.br/~pf/livrinho-AA/>