

PCC104 – Prova 2

Universidade Federal de Ouro Preto

1. Resolva a seguinte recorrência:

$$T(n) = 2T(n - 1) + n, \text{ para } n > 0, \text{ com } T(0) = 1.$$

2. Apresente um algoritmo para o qual a expressão da Questão 1 represente o número de operações.
3. Considere a sua implementação do *Recursive Maximum Finder*, `find_max(L)` (Lista 4, Recursion, Ex 1). Mostre a árvore de chamadas da sua implementação para `find_max([1,5,3,9,2])`
4. Apresente a análise de complexidade completa para a sua implementação do método `reverse_string(S)` (Lista 3, Stacks and Queues, Ex. 2)
5. Considere uma matriz `M`, de dimensão $n \times m$, ordenada. Apresente um algoritmo baseado em busca binária, `binary_search(M, x)`, que retorna `True` se `x` estiver em `M` e `False`, caso contrário. (Sugestão: Faça uma busca binária para encontrar a linha em que o número provavelmente está e, em seguida, execute a busca binária comum nessa linha.)

Exemplos:

(a)

$$M = \begin{bmatrix} 1 & 2 & 5 \\ 6 & 7 & 8 \\ 11 & 15 & 16 \\ 20 & 22 & 26 \\ 28 & 30 & 31 \end{bmatrix} x = 16$$

>> True

(b)

$$M = \begin{bmatrix} 11 & 15 & 16 \\ 20 & 22 & 26 \end{bmatrix} x = 3$$

>> False