PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS E INFORMÁTICA
DPTO DE CIÊNCIA DA COMPUTAÇÃO – CIÊNCIA DA COMPUTAÇÃO
PROJETO E ANÁLISE DE ALGORITMOS – PROF. SILVIO JAMIL F. GUIMARÃES
2025/1 (EXERCÍCIO 1)

Aluno: (não vale ponto)

QUESTION

a) Defina notação **O** discorrendo também sobre funções assintóticas. Dê exemplos de funções de complexidade com suas respectivas ordens de complexidade em termos de **O**

b) Indique se as alternativas abaixo são verdadeiras ou falsas justificando suas respostas por meio da definição da notação **O**

   a) $2^{n-1} = \mathbf{O}(2^n)$
   b) $2^{n+1} = \mathbf{O}(2^n)$
   c) $2^{3n} = \mathbf{O}(4^n)$
   d) $10000n^2 + 10000n + nlogn = \mathbf{O}(n^3)$
   e) $10000n^2 + 10000n + nlogn = \mathbf{O}(nlogn)$

c) Prove que:

   a) Seja $h_1(n) = \mathbf{O}(f(n))$ e $h_2(n) = \mathbf{O}(g(n))$. Mostre que $h_1(n) + h_2(n) = \mathbf{O}(max\{f(n), g(n)\})$
   b) Seja $h_1(n) = \mathbf{O}(f(n))$ e $h_2(n) = \mathbf{O}(g(n))$. Mostre que $h_1(n) \times\ h_2(n) = \mathbf{O}(f(n) \times\ g(n))$
   c) Seja $h_1(n) = \mathbf{O}(f(n))$ e c uma constante positiva. Mostre que $h_1(n) \times\ c = \mathbf{O}(f(n))$

d) Analise as assertivas a seguir, assinalando V, se verdadeiras, ou F, se falsas.

   ( ) Seja um vetor com n elementos. Seja um algoritmo A para encontrar um elemento neste vetor usando uma estratégia de busca de binária. A relação de recorrência deste algoritmo será

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

   ( ) Seja um vetor com n elementos. Seja um algoritmo A para encontrar o maior elemento do vetor. Este algoritmo A, com ordem de complexidade O(n), pode possuir a seguinte a relação de recorrência

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

   ( ) Seja um vetor com n elementos. Seja um algoritmo A para encontrar o maior elemento do vetor. Este algoritmo A, com ordem de complexidade O(log n), pode possuir a seguinte a relação de recorrência

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

   ( ) Sejam dois vetores A e B contendo m e n elementos, respectivamente. Seja o seguinte algoritmo: (i) selecione o menor elemento de A e compare com o menor elemento de B; (ii) copie o menor elemento encontrado para um vetor C; (iii) repita os procedimentos (i) e (ii) enquanto houver elemento ainda não copiado para C. O custo computacional deste algoritmo é O(mn).

   ( ) Se $f'(n) = O(g'(n))$ e $f''(n) = O(g''(n))$, então $f'(n) + f''(n) = O(g'(n) + g''(n))$.

   ( ) $f(n) = O(g(n)$ se $\exists c > 0, n_0 \geq 0$ tal que $g(n) \leq cf(n), \forall n \geq n_0$.