Análise e Técnicas de Algoritmos
Terceira Lista de Exercícios

Aluno(a):

1. Resolva as seguintes relações de recorrência:
   (a) $T(n) = T(n-1) + 1$.
   (b) $T(n) = T(n/2) + 1$.
   (c) $T(n) = 2.T(n-1) + 1$.
   (d) $T(n) = 2.T(n/2) + 1$.
   (e) $T(n) = T(n-1) + n$.
   (f) $T(n) = T(n/2) + n$.
   (g) $T(n) = 2.T(n-1) + n$.
   (h) $T(n) = 2.T(n/2) + n$.
   (i) $T(n) = 4.T(n/2) + n^3$.
   (j) $T(n) = 4.T(n/2) + n^2 + n$.
   (k) $T(n) = 2.T(n/2) + n.logn$.
   (l) $T(n) = T(n/2) + T(n/3) + n$.
   (m) $T(n) = 9.T(n/3) + n^2$.
   (n) $T(n) = T(\sqrt{n}) + 1$.

2. Defina Bigh-oh, Big-Theta e Big-Omega. Dê exemplos.

3. Prove que $log(n!) \in O(n. \log n)$.

4. Suponha que $f_1(n) = O(g_1(n))$ e $f_2(n) = O(g_2(n))$. Então, $f_1(n)+f_2(n) = O(g_1(n) + g_2(n))$. Se esta conclusão está correta prove-a. Se está errada, mostre um contra-exemplo.

5. Responda verdadeiro (V) ou falso (F) e justifique sua escolha para cada um dos segiuntes itens:
   (a) ( ) $2^n + 3^n + n^{64} = O(n^{64})$.
   (b) ( ) $2^{n+1} = O(2^n)$.

(c) ( ) $2^{2n} = O(2^n)$.

(d) ( ) Se $f(n) = n^2$, então $3f(n) + 2n = O(f(n))$.

(e) ( ) Se $T(1) = d$ e, para $n > 1$, $T(n) = 2T(n/2)$ então $T(n) = O(n)$.

(f) ( ) Quando provamos a corretude de algoritmos iterativos com laços aninhados, começamos provando a corretude do laço m e então partimos para o laço mais interno.

(g) ( ) Se voce tem um algoritmo que *roda* em te $O(n)$ e um algoritmo que *roda* em tempo $O(n^3)$, o primeiro algo é mais rápido na prática do que o segundo.