

28/03/2019

Exercícios de Revisão

1. Sejam $f(n)$, $g(n)$ duas funções assintóticas positivas e a e b . Prove que as afirmativas abaixo são verdadeiras ou falsas, usando para isso as definições das notações assintóticas ou contraexemplos.
 - a) $2^{n+1} = O(2^n)$
 - b) $2^{2n} = O(2^n)$
 - c) $f(n) + g(n) = O(\max(f(n), g(n)))$
 - d) A notação θ é simétrica, ou seja, $f(n) = \theta(g(n))$ se e somente se $g(n) = \theta(f(n))$
2. Resolva a seguinte questão sobre recursividade:
 - a. Escreva uma **função recursiva** `int Palindromo(int esq, int dir, char palavra[])` que testa se uma determinada palavra é um palíndromo e retorna 1 em caso positivo e 0 em caso negativo. Um palíndromo é uma palavra que é lida da mesma forma da esquerda para direita ou da direita para esquerda (ex. ovo, arara). A palavra é passada para o função através de um vetor de caracteres limitada pelos os índices `esq` e `dir`, por exemplo: `Palindromo(0, 4, "arara")`
 - b. Calcule qual é a **função de complexidade** para o número de comparações de caracteres da sua função no melhor caso e no pior caso. Para isso, **determine e resolva** a equação de recorrência dessa função recursiva. Qual é a **ordem de complexidade** de sua função?
 - c. Qual seria a complexidade de uma implementação não recursiva dessa mesma função? Qual das duas implementações vocês escolheria? Justifique.
3. Vários algoritmos em computação usam a técnica de “Dividir para Conquistar”: basicamente eles fazem alguma operação sobre todos os dados, e depois dividem o problema em sub-problemas menores, repetindo a operação. Uma equação de recorrência típica para esse tipo de algoritmo é mostrada abaixo. Resolva essa equação de recorrência usando o Teorema Mestre.

$$T(n) = 2T(n/2) + n;$$

$$T(1) = 1;$$