

UNIVERSIDADE FEDERAL DO MARANHÃO

DEPARTAMENTO DE INFORMÁTICA

CURSO DE CIÊNCIA DA COMPUTAÇÃO

DISCIPLINA ESTRUTURAS DE DADOS

II LISTA DE EXERCÍCIOS (2005.1)

<COMPLEXIDADE DE ALGORITMOS>

- 1 Prove, por definição, que a função O é transitiva.
- 2 Para resolver um problema, foram apresentados dois algoritmos A_1 e A_2 com complexidades $f(n) = n^2$ e $g(n) = 10.n.\log_{10}(n)$, respectivamente. Qual desses algoritmos você escolheria, considerando na sua justificativa o tamanho do problema em questão.
- 3 Seja $f(n)$ a função de complexidade relacionada ao número de atribuições realizadas em um algoritmo. Apresente 3 versões A_1 , A_2 e A_3 , com complexidades $O(1)$, $O(n)$ e $O(n^2)$ respectivamente.
- 4 Prove as seguintes afirmativas:
 - a. $\log n = O(n^{2/3})$;
 - b. $n^{1/3} = \Theta(n)$
 - c. $n = \Omega(\log n)$
- 5 Qual a função de complexidade dos seguintes procedimentos vistos ao longo deste curso (considere o melhor e pior casos e as estruturas envolvidas no algoritmo – pilha, fila, etc.)
 - a. **VerificaEscopo(A):**
Entrada: Expressão infixa A com N símbolos;
Saída: V ou F se a expressão A contendo $\{[(e)]\}$ for válida ou não
 - b. **AvaliaPosfixa(B):**
Entrada: Expressão posfixa B com N símbolos;
Saída: resultado da expressão
 - c. **ConverteInfixaPosfixa(C):**
Entrada: Expressão infixa C com N símbolos;
Saída: Expressão posfixa equivalente
 - d. As 3 primitivas elementares de uma fila de prioridades implementada através de N (máximo de prioridades) filas comuns;
 - e. As 3 primitivas elementares de uma fila de prioridades implementada através de uma lista encadeada simples (N = máximo de prioridades);