



**PRIMEIRA MINI-PROVA**  
**31 de janeiro de 2013**

- Esta prova contém 04 (quatro) questões.
- A duração da prova é de 01 (uma) hora.
- A detecção de cópia implicará na atribuição de nota 0 (zero) à prova.

**QUESTÃO 1**

Um algoritmo  $A$  resolve um problema em tempo  $\Omega(n^2)$  no pior caso, enquanto um outro algoritmo  $B$  resolve o mesmo problema em  $O(n^2)$  no melhor caso. Qual dos dois algoritmos é o mais rápido. Explique sucintamente.

**QUESTÃO 2**

A *altura* de uma árvore binária pode ser definida como

$$h(T) = \begin{cases} 0, & \text{se } T = \emptyset \\ 1 + \max\{h(T_L), h(T_R)\}, & \text{se } T = \begin{array}{c} r \\ \swarrow \quad \searrow \\ T_L \quad T_R \end{array}, \end{cases}$$

onde  $r$  representa um nó e  $T_L, T_R$  as sub-árvores binárias à esquerda e à direita de  $r$ .

A altura de um nó de uma árvore binária é então definida como a altura da sub-árvore nele enraizada.

Escreva um procedimento recursivo que recebe um ponteiro para a raiz de uma árvore binária e imprime a altura dos nós desta árvore *em pós-ordem*.

**QUESTÃO 3**

A remoção de um nó de uma árvore AVL faz-se como numa árvore de busca binária, seguida por rotações eventualmente necessárias para a restauração do seu balanceamento. Considere a árvore AVL  $T$  resultante da inserção dos valores (5, 3, 10, 2, 4, 7, 11, 1, 6, 9, 12, 8) nesta ordem. Pede-se:

- Desenhar  $T$
- Desenhar  $T$  logo após a remoção do valor 10 (antes das eventuais rotações)
- Desenhar  $T$  após cada rotação necessária

**QUESTÃO 4**

Considere o vetor  $V = (1, 3, 6, 5, 9, 7, 2, 8, 4)$ . Represente:

- A *max-heap* binária resultante das inserções sucessivas dos elementos de  $V$  (na mesma ordem em que aparecem no vetor);
- A *max-heap* binária resultante da construção bottom-up com a função *max-heapify*.