



PROVA FINAL (2a.CHAMADA) — 14 de Março de 2014

- Esta prova contém 04 (quatro) questões dividida em duas partes.
- A duração da prova é de 2h00min.

---

PARTE I

---

**QUESTÃO 1** (2,5pt)

Considere uma árvore-B de grau mínimo  $t = 3$ . Represente a inserção das chaves

$G, S, Q, K, C, L, H, T, V, W, M, R, N, P, A, B, X, Y, D, Z, E, U, F$

nesta ordem. Para tal, exiba a configuração da árvore *apenas* imediatamente *após* a inserção das chaves que provocarem divisões de nós e desenhe *também* a configuração final da árvore.

**QUESTÃO 2** (2,5pt)

Considere uma tabela de dispersão de tamanho  $m = 7$  que utiliza a política de *hashing fechado* com resolução de colisões por *double hashing* com funções de dispersão  $h_0(k) = k \bmod m$  e  $h_1(k) = (2k - 1) \bmod m$ . Represente a configuração da tabela após a inserção das chaves 24, 76, 20, 38, 9, 33, 63, nesta ordem.

---

PARTE II

---

**QUESTÃO 3** (2,5pt)

Escreva um algoritmo guloso para, dados um grafo  $G = (V, E)$  e um vértice  $z$  em  $V$ , encontrar a quantidade de caminhos mínimos entre o vértice  $z$  e os demais vértices do grafo  $G$ . O algoritmo deve ser escrito numa pseudo-linguagem estruturada, sem supor a existência de qualquer função. O algoritmo deve supor que o grafo já está numa estrutura de Listas de Adjacências.

**QUESTÃO 4** (2,5pt)

Escreva um algoritmo baseado na técnica de backtracking para encontrar uma 3-coloração dos vértices de um grafo não direcionado  $G = (V, E)$  dado. A coloração deve dar prioridade às cores na seguinte ordem:

Cor 1  $\ll$  Cor 2  $\ll$  Cor 3,

onde  $A \ll B$  significa que a cor A tem prioridade sobre a cor B. Se for possível fazer uma 3-coloração no grafo, a coloração deve ser colocada em um array chamado COR\_DO\_VERTICE, de tamanho  $n = |V|$ , onde cada entrada  $i$  conterá 1, 2 ou 3, de acordo com a cor designada para o vértice  $i$ . Caso o grafo da entrada não seja 3-colorível, então o algoritmo deve imprimir a frase: "Não é possível 3-colorir o grafo dado". O algoritmo deve ser escrito numa pseudo-linguagem estruturada, sem supor a existência de qualquer função. O algoritmo deve supor que o grafo já está numa estrutura de Listas de Adjacências.