Universidade Estadual de Maringá
Curso: Ciência da Computação
Disciplina: DIN6897 – Organização e Recuperação de Dados
Profa. Valéria D. Feltrim *03/06/2013*

Nome: _____ **RA**: _____

2ª Prova

1. (1,0) Codifique a mensagem abaixo usando código de Huffman. Construa a tabela de frequência, a árvore de Huffman e a tabela de códigos para os respectivos símbolos.

   AMO_AMAR_AMORA.

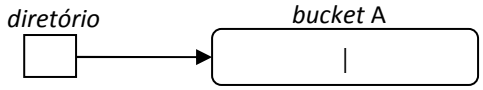
2. Suponha que uma aplicação utilize um arquivo hash com 13 endereços (índices de 0 a 12) e empregue h(x) = x mod 13 como função de hashing, em que x representa a chave do elemento cujo endereço deseja-se computar. Inicialmente, esse arquivo hash encontra-se vazio. Em seguida, a aplicação solicita uma sequência de inserções de elementos cujas chaves aparecem na seguinte ordem: 44, 46, 49, 70, 27, 71, 90, 97, 96. Com relação à aplicação descrita, faça o que se pede a seguir.

   a. (0,5) Assuma que as colisões são tratadas por meio de inspeção linear. Esboce o arquivo hash para mostrar o seu conteúdo após a referida sequência de inserções.

   b. (0,75) Assuma que as colisões são tratadas por meio de encadeamento em área separada. Esboce o arquivo hash (área primária) para mostrar seu conteúdo após a referida sequência de inserções, bem como o arquivo de colisões (área de overflow).

   c. (0,25) Demonstre qual das duas formas de tratamento de colisão ((a) ou (b)) é mais eficiente em termos do Comprimento Médio da Busca.

3. Suponha um sistema de ***hashing* extensível** em que cada *bucket* pode armazenar duas chaves. Inicialmente o sistema está vazio, como mostrado na figura abaixo.

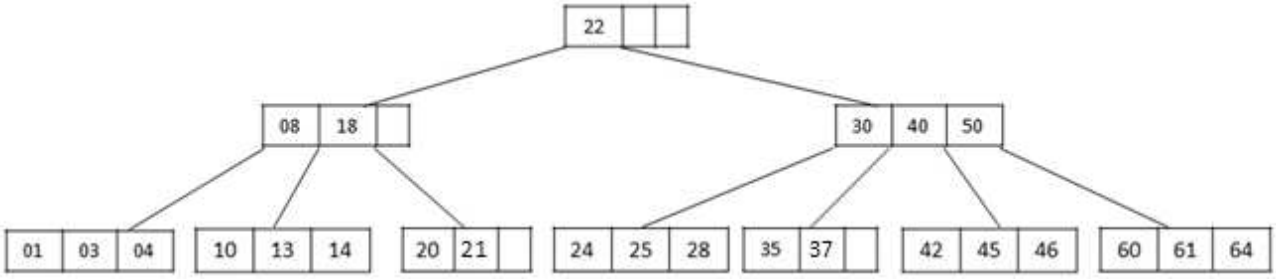


   a. (1,0) Mostre graficamente como fica o sistema (diretório e *buckets*) após a inserção de cada uma das seguintes chaves, nesta ordem: 001100, 101010, 010100, 011110, 011010, 110011, 101101 (Sugestão: redesenhe o sistema toda vez que ocorrer um *overflow*). Indique na figura qual é a profundidade global e a profundidade local de cada *bucket*.

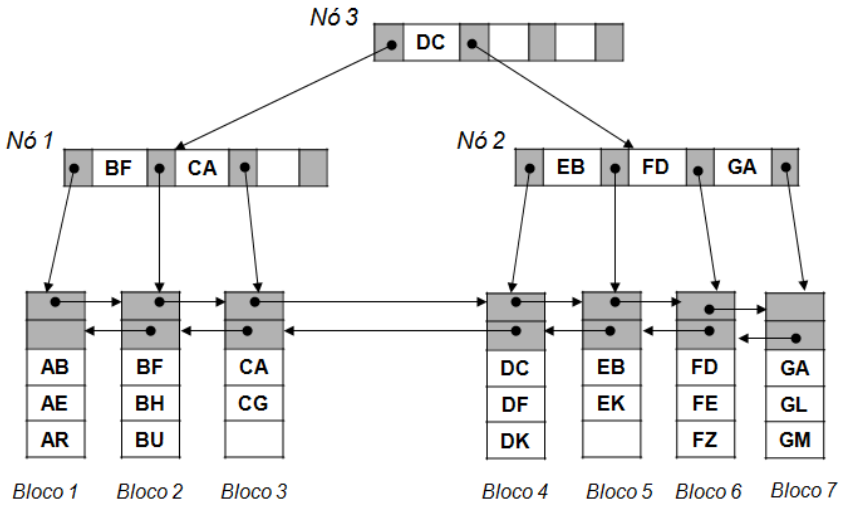
   b. (0,25) Dê exemplos de duas chaves que causem *overflow*, sendo (i) uma que cause expansão do diretório e (ii) uma que cause a criação de um novo *bucket*, mas sem precisar expandir o diretório.

   c. (0,25) Na estrutura resultante de (a), indique quais são os *buckets* "amigos" (*buddy buckets*). Justifique sua resposta.

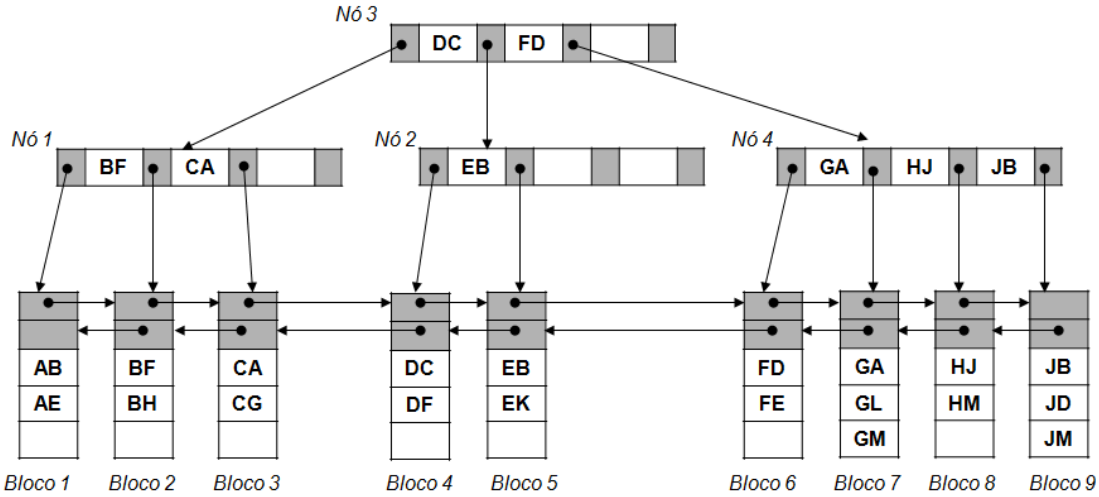
4. (1,0) Dada a árvore-B* de ordem 4 mostrada abaixo, faça a inserção das chaves 43, 48, 02, nesta ordem, representando a árvore resultante após cada operação de inserção. Quando for possível dividir usando tanto a irmã direita quanto a irmã esquerda, escolha a irmã direita.



5. (1,0) Dada a árvore-B+ abaixo, em que os nós de índice têm ordem 4 e blocos de registros têm fator de bloco 3, indique o seu estado após cada uma das seguintes operações de inserção, pela ordem: FG, FA, AM. Nos blocos de registros, faça uso de redistribuição na inserção como estratégia para minimizar a fragmentação. No entanto, o índice deve ser tratado como uma árvore-B comum, ou seja, sem uso de redistribuição na inserção.



6. (1,0) Dada a árvore-B+ abaixo, em que os nós de índice têm ordem 4 e blocos de registros têm fator de bloco 3, indique o seu estado após cada uma das seguintes operações de remoção, pela ordem: AB, FE, EB. Blocos com apenas um registro estão em situação de *underflow*, diferente dos nós de índice, que podem conter apenas uma chave.



Boa Prova!