

## Lista de Exercícios AEDII – 03 – Profº Helton

**BoaNoche Development Group™**

### **1. Explique o que são buckets e porque eles são utilizados quando se trabalha com hash em disco?**

Buckets são estruturas que permitem que um Hash armazene mais de um registro por chave, esse tipo estrutura é utilizada em hash em disco com o intuito de otimizar a estrutura, ou seja não gasta-se recursos operacionais para tratar as colisões.

### **2. Diferencie as técnicas de hashing extensível e hashing linear.**

Ambos utilizam dos buckets, a diferença está na duplicação do diretório. Na técnica de hashing extensível o diretório é duplicado em um único passo, já no hashing linear os buckets vão sendo duplicados gradualmente. Outra diferença é que no extensível a função de hash não varia. Já no linear, há variação da função hash.

### **3. Para que serve a área extra quando se usa hashing linear (também conhecida como área de overflow)?**

A área de overflow é utilizada quando uma chave é inserida em um bucket cheio, e não é a vez dele se duplicar. Dessa maneira, um novo bucket é ligado por um ponteiro a esse bucket já cheio, e a chave é inserida nesse novo cesto, que é a tal área extra.

### **4. Qual o número máximo de comparações necessárias para se localizar um registro numa árvore AVL com 1.000.000 de chaves? Esse é um valor aceitável para pesquisa em memória secundária?**

O número máximo de comparações necessárias para se localizar um registro numa árvore AVL é de, aproximadamente,  $\log_2(n)$ . Nesse exemplo são necessárias, aproximadamente, 20 comparações, o que é inaceitável para pesquisa em memória secundária. (até 6 comparações, aceitável)

### **5. Explique o conceito de árvore binária paginada. Qual a sua principal desvantagem?**

É uma árvore binária em que ao invés de se ler um registro por vez, lê-se uma grande quantidade de registros sequencialmente a um custo pequeno. A principal desvantagem é sua construção “Top-Down” assim é difícil garantir que as chaves da raiz serão boas separadoras e também impedir agrupamento de chaves que não deveriam estar na mesma página.

### **6. Explique quais as principais características de uma árvore B de ordem n.**

As principais características de uma árvore B de ordem n: a ocupação mínima por nó é de  $((n/2)-1)$  e a ocupação máxima é de n-1 registros. Cada nó pode ter até n ponteiros “filhos”.

### **7. Descreva como ocorrem os processos de promoção e subdivisão durante a inserção de chaves numa árvore B.**

Quando um nó está cheio e um novo registro deve ser inserido no mesmo, acontecem os processos de subdivisão e promoção. Esse nó que já tinha  $2t-1$  elementos é subdividido em dois nós com t elementos cada. Um elemento, ou o mais a esquerda do nó à direita ou o mais a direita do nó à esquerda, é “promovido”, ou seja, sobe de forma ordenada para o nó pai. Assim, o nó pai recebe um novo elemento e um novo filho.

**8. Quais os casos possíveis durante a remoção de uma chave de uma árvore B?**

O elemento a ser removido estar em um nó interno ou em uma folha.

**9. Descreva com suas palavras os passos necessários para a remoção de uma chave de uma árvore B.**

Primeiro verifica-se se a chave está em um nó interno ou em uma folha. No primeiro caso, o sucessor da chave (próximo elemento sequencialmente) será movido para a posição eliminada e o processo de eliminação procede com a eliminação de um valor de nó folha, que é o seguinte: checa-se se após a remoção o nó terá um número de registros iguais ou maiores à ocupação mínima, se sim, remova o registro. Caso contrário, verificam-se os irmãos à esquerda e à direita do nó do qual fora removido um elemento, se em algum deles há mais elementos do que a ocupação mínima, a chave  $k$  que separa os dois nós desce para o nó de onde foi removida uma chave e a chave do irmão é promovida no lugar desta que desceu (caso seja o irmão à esquerda, a chave mais à direita. Caso seja o irmão à direita, a chave mais à esquerda). Agora no caso dos dois irmãos terem exatamente a ocupação mínima, o nó de onde foi retirada uma chave se funde com um de seus irmãos mais a chave separadora do pai. Se o pai tiver também a ocupação mínima de um nó deve-se seguir os passos da verificação dos irmãos e proceder recursivamente.

**10. Argumente porque a altura de uma árvore B com  $n$  chaves é  $h \leq \log_t (n+1) / 2$ . Sendo que o número mínimo de elementos por nó é  $t - 1$ .**

A seguinte fórmula pode ser deduzida da seguinte maneira:

Como em uma Btree cada nó possui  $t-1$  chaves, logo uma árvore de altura 1 possui 2 nós, já no nível 2 terá  $2t$  nós e no nível 3 terá  $2t^2$ , e na altura  $h$  terá  $2t(t^{h-1}-1)$

Logo o número de nós será :

$$n \geq 1 + (t-1) \sum_{i=1}^h 2t^{i-1} = 1 + 2(t-1) \frac{t^h - 1}{t - 1} = 2t^h - 1$$

Logo a altura será  $\log_t (n+1)/2$

**11. Explique qual o conceito e quais as vantagens de uma árvore B+.**

A Btree+ tem o mesmo princípio que a Btree, porém dado um nó, o seu filho mais a direita tem o endereço do nó-irmão do mesmo nível, isso traz uma enorme vantagem para realizar percursos em largura ou processos sequenciais, aumentando o desempenho através da diminuição do acesso ao disco.

**12. Como funcionam e onde são utilizados os processos de concatenação e redistribuição de chaves em uma árvore B+?**

**13. Porque é interessante escolher separadores ao invés de chaves nas árvores B+? E como pode ser feita a escolha deste separador?**

**14. Porque a ocorrência de overflow ou underflow no conjunto de índices não acompanha necessariamente o mesmo fato no conjunto de sequências?**