

Estrutura de Dados II

Lista de Exercícios

Exercícios sobre Tabela Hash, árvore B e árvore B+

Questão 1: (POSCOMP 2011) Ao usar o cálculo de endereço ou hashing, geralmente é necessário o uso de um método de tratamento de colisões. Sobre esse método, é correto afirmar:

- a) O tratamento de colisões é necessário apenas quando a tabela está cheia e se necessita inserir mais uma chave.
- b) O tratamento de colisões é necessário para determinar o local da chave no momento da inserção na tabela.
- c) O tratamento de colisões é necessário quando a tabela está vazia, pois não é possível calcular o endereço diretamente nesse caso.
- d) O tratamento de colisões é necessário quando a chave inserida ainda não existir na tabela de endereçamento.
- e) O tratamento de colisões é necessário, pois o hashing gera repetição de endereço para diferentes chaves.

Questão 2: Demonstre a inserção das chaves 5, 28, 19, 15, 20, 33, 12, 7 e 10 numa tabela de hash com colisões resolvidas por lista encadeada. Considere a tabela com $m = 9$ posições e a função hash como sendo $h(k) = k \% m$. Reconstrua a tabela para $m = 11$ (primo) e comente os resultados.

Questão 3: Utilizando um vetor estático de tamanho $m = 101$ para a criação de uma tabela hash, entradas k de tipo inteiro e função de hash $h(k) = k \% m$, faça o que se pede:

- a) Implemente as funções de inserção, busca e remoção para uma tabela hash utilizando encadeamento como método para tratamento de colisões. (utilize alguma de suas implementações anteriores de lista encadeada).
- b) Implemente as funções de inserção, busca e remoção para uma tabela hash utilizando o endereçamento aberto para tratar colisões. Implemente o método de *quadratic probing* para tratar colisões.

Questão 4: Sobre árvore B, responda:

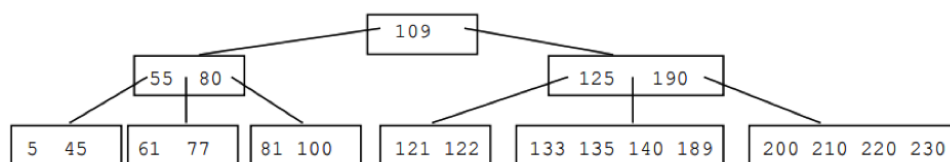
- a) Por que as árvores B são melhores do que as árvores binárias para pesquisa em disco?
- b) Explique quando ocorre a redução da altura de uma árvore B. Desenhe um exemplo para ilustrar a explicação.

- c) No processo de remoção, por que é preferível a redistribuição à concatenação?
- d) Explique a seguinte sentença: "Árvores B são construídas de baixo para cima, enquanto árvores binárias são construídas de cima para baixo".
- e) Como uma folha de uma árvore B difere de um nó interno? Quais são as partes necessárias a uma folha?
- f) Mostre os resultados de inserir as chaves a seguir em uma árvore B de ordem 2 inicialmente vazia: 14, 39, 1, 6, 41, 32, 8, 38, 43, 3, 36, 56, 65, 67, 89, 78. Após essas inserções, mostre os resultados de remover as chaves ímpares nessa estrutura.
- g) Descreva as características das páginas de uma árvore B com n chaves de altura máxima.
- h) Descreva as características das páginas de uma árvore B com n chaves de altura mínima.
- i) Descreva o caso em que ocorre cisão de uma página em uma árvore B. Descreva o processo de cisão.
- j) Descreva o caso em que ocorre a concatenação em uma árvore B. Descreva o processo de concatenação.
- k) A concatenação ocorre apenas nas folhas? Justifique.
- l) Descreva o caso em que ocorre a redistribuição em uma árvore B. Descreva o processo de redistribuição.
- m) A redistribuição ocorre apenas nas folhas? Justifique.

Questão 5: Mostre a Árvore-B de ordem 2 que resulta de carregar os seguintes conjuntos de chaves em ordem:

- a) C G J X
- b) C G J X N S U O A E B H I
- c) C G J X N S U O A E B H I F
- d) C G J X N S U O A E B H I F K L Q R T U W Z

Questão 6: Considere a árvore B de ordem 2 abaixo: 109 55 80 125 190 5 45 61 77 81 100 121 122 133 135 140 189 200 210 220 230.



Realize as seguintes operações, utilizando sempre a árvore resultante da operação anterior. Redesenhe a árvore a cada passo, indicando os nós que sofrem modificações, bem como a ocorrência de CISÃO, REDISTRIBUIÇÃO ou CONCATENAÇÃO:

- a) inserção de 150;
- b) remoção de 189;

- c) remoção de 80;
- d) inserção de 65;
- e) inserção de 66;
- f) remoção de 109.

Use o mesmo critério para remoção dos nós.

Questão 7: Considere as seguintes estruturas para a implementação de uma árvore B de ordem D na memória:

```
#define dado int
#define D 2
typedef struct Celula celula;
typedef struct Pagina pagina, *arvore;
struct Celula{
    dado info;
    int chave;
    pagina *filho;
};
struct Pagina {
    celula *pag;
    celula *pai; //Opcional
};
```

Implemente as funções de **criação de página**, **inserção** e **remoção** de nós, lembre-se de tratar os casos de cisão, redistribuição e concatenação adequadamente e que uma página da árvore b possui $2D + 1$ células, onde na primeira célula, apenas o campo do ponteiro será utilizado e apontará para a página que contém chaves menores que a chave da página atual (se a atual não for uma folha).

Questão 8: (POSCOMP 2010) Em uma Árvore-B de ordem m , temos que: (i) cada nó contém no mínimo m registros (e $m + 1$ descendentes) e no máximo $2m$ registros (e $2m + 1$ descendentes), exceto o nó raiz que pode conter entre 1 e $2m$ registros; (ii) todas os nós folha aparecem no mesmo nível. Sobre Árvores-B, é correto afirmar:

- a) O particionamento de nós em uma Árvore-B ocorre quando um registro precisa ser inserido em um nó com $2m$ registros.
- b) O particionamento de nós em uma Árvore-B ocorre quando um registro precisa ser inserido em um nó com menos de $2m$ registros.
- c) O particionamento de nós em uma Árvore-B ocorre quando a chave do registro a ser inserido contém um valor (conteúdo) intermediário entre os valores das chaves dos registros contidos no mesmo nó.
- d) O particionamento de nós ocorre quando é necessário diminuir a altura da árvore.
- e) Em uma Árvore-B, aumenta em um nível sua altura, toda vez que ocorre o particionamento de um nó.

Questão 9: Considere uma árvore B+ de ordem 2 e responda as questões a seguir:

- a) Mostre a árvore resultante da inserção da seguinte sequência de chaves: 20, 40, 60, 80, 13, 90, 100, 85, 87, 65, 10, 5, 50, 55, 45.
- b) Considerando a árvore obtida no item anterior, apresente a árvore resultante da remoção das chaves: 13, 55, 60.