

Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação
Disciplina de Organização de Arquivos
Profa. Dra. Cristina Dutra de Aguiar Ciferri

Lista de Exercícios

Essa lista de exercícios contém exercícios referentes às aulas ministradas com o conteúdo dos capítulos 1 a 10 do livro texto "File Structures", dos autores M. J. Folk e B. Zoellick. Apesar de bastante abrangente, essa lista de exercícios não é exaustiva e, portanto, o aluno deve basear seus estudos nas aulas ministradas, no conteúdo das transparências, no conteúdo do livro texto e também em materiais adicionais que possam complementar o aprendizado. Bons estudos!

Suponha um arquivo de dados com registros de tamanho variável, indexado por um **índice de chave primária**. Considere as seguintes operações: inserção de registro, remoção de registro e modificação de registro. Quais dessas operações são necessárias, no arquivo de índice, para

- inserção de um registro no arquivo de dados?
- remoção de um registro no arquivo de dados?
- atualização de um valor de um campo não chave no arquivo de dados?

Suponha um arquivo de dados com registros de tamanho variável, indexado por um **índice de chave secundária fortemente acoplado**. Considere as seguintes operações: inserção de registro, remoção de registro e modificação de registro. Quais dessas operações são necessárias, no arquivo de índice, para

- inserção de um registro no arquivo de dados?
- remoção de um registro no arquivo de dados?
- atualização de um valor de um campo não chave no arquivo de dados?

Suponha um arquivo de dados com registros de tamanho variável, indexado por um **índice de chave secundária fracamente acoplado**. Considere as seguintes operações: inserção de registro, remoção de registro e modificação de registro. Quais dessas operações são necessárias, no arquivo de índice, para

- inserção de um registro no arquivo de dados?
 - remoção de um registro no arquivo de dados?
 - atualização de um valor de um campo não chave no arquivo de dados?
-

Quais os problemas de se usar árvores binárias de busca e árvores AVL como estruturas de índices para grandes volumes de dados?

Explique a seguinte sentença: “B-Trees são construídas de baixo para cima, enquanto árvores binárias são construídas de cima para baixo”.

Como uma folha de uma árvore-B difere de um nó interno?

Considere uma árvore-B de ordem 4. Considere que o primeiro elemento do segundo nó é a chave promovida durante o particionamento do nó. Insira as seguintes chaves no índice:

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

Na árvore-B do exemplo anterior, insira a chave \$, sendo que $\$ < A$.

Considere uma árvore-B de ordem 4. Considere que o último elemento do primeiro nó é a chave promovida durante o particionamento do nó. Insira as seguintes chaves no índice:

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

Considere uma árvore-B de ordem 3. Considere que o primeiro elemento do segundo nó é a chave promovida durante o particionamento do nó. Insira as seguintes chaves no índice:

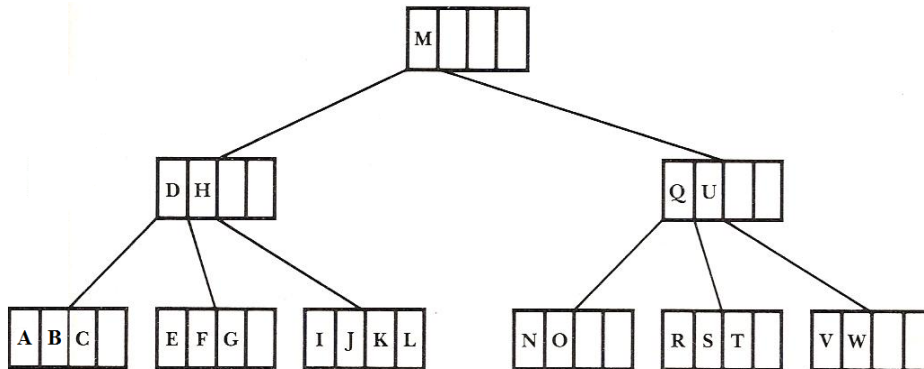
A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, X, Z

Qual o efeito da inserção das chaves em ordem alfabética? A árvore degenerou?

Quais as vantagens do uso da técnica de redistribuição quando da inserção e remoção de chaves em uma árvore-B?

Suponha que você vai remover uma chave em uma árvore-B, a qual causa *underflow* na página. Se pela página irmã do lado direito é necessária concatenação, e pela página irmã do lado esquerdo é possível redistribuição, qual opção você escolheria? Por quê?

Considere a árvore-B de ordem 5 a seguir. Usando o algoritmo de remoção apresentado em sala de aula, remova as chaves A, B, Q e R.



Qual a diferença entre uma árvore-B e uma árvore-B*? Quais as vantagens e as desvantagens da árvore-B*?

O que é uma árvore-B virtual? Quais as vantagens e as desvantagens da árvore-B virtual?

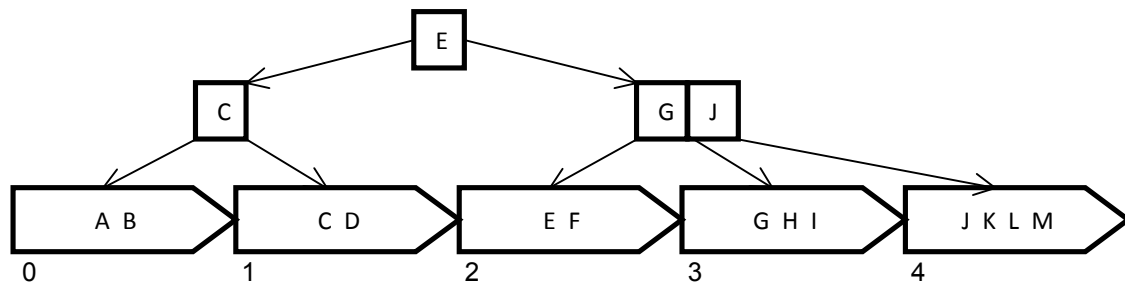
O que é uma árvore-B⁺? Quando ela é necessária?

O que são separadores? Qual a diferença entre separadores e chaves?

Considere uma árvore-B⁺ de ordem 3. Considere que primeiro elemento do segundo nó é a chave promovida durante o particionamento do nó. Considere *sequence sets* (ou seja, blocos) que contenham 3 registros por nó. Insira as seguintes chaves no índice:

A, B, C, D, E, F, G, H, I e J

Considere a árvore-B+ de ordem 3 a seguir. Considere *sequence sets* (ou seja, blocos) que contenham no mínimo 2 registros por nó e no máximo 4 registros por nó. Remova as seguintes chaves do índice: L, M, K, A



Considere a árvore-B+ de ordem 3 a seguir. Considere *sequence sets* (ou seja, blocos) que contenham no mínimo 2 registros por nó e no máximo 4 registros por nó.

a) Quais os separadores dos blocos a seguir?

bloco 1	ADAMS ... BAIRD ... BIXBY ... BOONE ...
bloco 2	BYNUM ... CART ... COLE ... DAVES ...
bloco 3	DENVER ... ELLIS ... FOLK ... FRANK ...

b) Construa a árvore sobre os blocos acima já existentes

c) Realize as seguintes operações

inserção de CARTER

inserção de DRAG

remoção de BIXBY

remoção de COLE

Para aplicações reais, juntamente com cada chave da árvore-B é armazenada uma referência para o registro do arquivo de dados que contém os dados que são associados à chave (abordagem 1). Outra solução consiste em, juntamente com cada chave da árvore-B, armazenar o registro completo de dados que contém os dados que são associados à chave (abordagem 2). Compare essas duas abordagens em termos de vantagens e desvantagens.

Em sala de aula, considerou-se que os campos da árvore-B que armazenam as chaves de busca são campos de tamanho fixo (abordagem 1). Entretanto, a árvore-B pode ser implementada considerando campos de tamanho variável para armazenar as chaves de busca (abordagem 2). Compare essas duas abordagens em termos de vantagens e desvantagens.

Qual a diferença entre *merging* e *matching*? O que essas diferenças acarretam nos algoritmos de *merging* e *matching*?

Considere as listas ordenadas a seguir, sendo que $k = 10$. Mostre o resultado do merging dessas listas, usando o algoritmo k multiway merging. Ilustre a árvore de decisão passo a passo.

13, 15, 69

1, 34, 73

3, 9, 27

7, 10, 78

19, 36, 47

79, 80, 81

5, 54

17, 28, 51

21, 43, 98

8, 24, 50

Considere um *heap* com 9 posições. Insira as seguintes chaves no *heap*, na seguinte sequência: K, L, A, M, X, T, W, V, U. Desenhe tanto o vetor do *heap* quanto a árvore binária correspondente.

Considere um arquivo onde os seguintes valores são chaves para uma função de *hash*: 2369, 3760, 4692, 4871, 5659, 1821, 1074, 7115, 1620, 2428, 3943, 4750, 6975, 4981, 9208. O arquivo de *hash* usa oito *buckets*, numerados de 0 a 7. Cada *bucket* é um bloco de dois registros. Carregue esses registros no arquivo na ordem dada, usando a função de *hash* $h(K) = K \bmod 8$.
