

Quarta Lista de Exercícios de Estrutura de Dados e Algoritmos 2/2006

1. Suponha que um arquivo de dados contém um número conhecido N de linhas, cada qual consistindo de uma chave ch e um dado d associado a mesma; e que esse arquivo já está em ordem crescente de chave. Escreva um procedimento que lê sequencialmente esse arquivo e constrói uma árvore binária de busca para esses elementos, onde todo o nível k , exceto possivelmente o último, tem 2^k elementos. O procedimento deve usar no máximo $O(N)$ operações, e no máximo $O(\log N)$ espaço adicional. *Dica:* Use um vetor para guardar as partes da árvore que podem ser construídas com os primeiros k elementos do arquivo, para $0 \leq K \leq N$.
2. Se precisamos percorrer uma lista simplesmente ligada na ordem *errada* (do último elemento para o primeiro), podemos primeiro inverter a lista, imprimir e inverter de novo. Procure adaptar a idéia para percorrer todos os nós de uma árvore binária de busca, sem usar recursão, pilha explícita, ou vetores auxiliares.
3. Suponha que temos uma estrutura de dados que armazena uma lista $F = (f_1, f_2, \dots, f_n)$, com dois apontadores, *ini* para o elemento f_1 e *fim* para o elemento f_n .
 - a) Suponha que o programa apenas insere elementos no início da lista (ponta f_1) retira elementos do fim da mesma ponta (f_n). Qual a representação de lista **encadeada mais adequada para essa estrutura, para que essas operações possam ser efetuadas da maneira mais rápida possível?**
 - b) Escreva o procedimento $Insere(F, x)$ para essa estrutura de dados.
 - c) Escreva o procedimento $x \leftarrow Retira(F)$ para essa estrutura de dados.
4. Escreva um algoritmo (e se possível, implemente em C) que crie uma árvore n -ária (inserindo e removendo nós), onde n é o número máximo de filhos que uma árvore n -ária pode ter.

5. Escreva um procedimento para remover nós de uma árvore AVL. Qual o seu custo?

OBS: Data de entrega da lista: 05/11/2006.