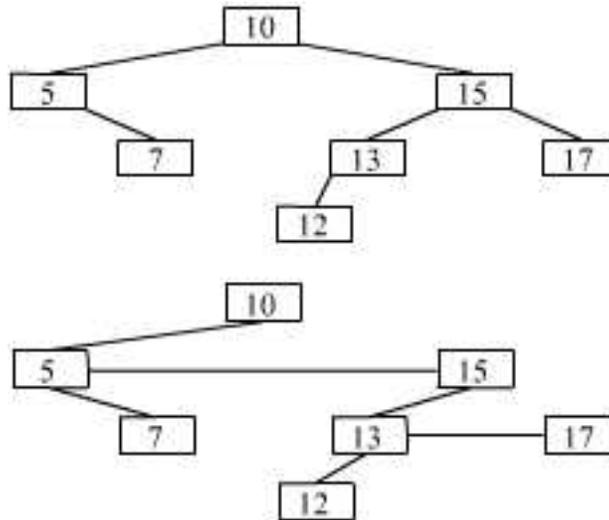


MAC 323 - Estruturas de Dados

Primeiro semestre de 2007

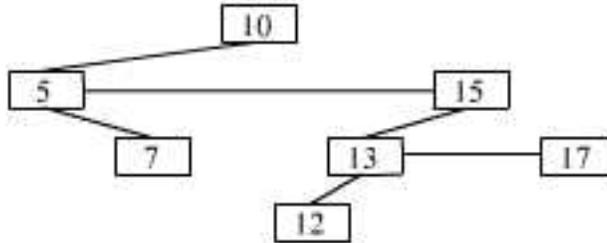
Segunda Lista de Exercícios – Entrega: 28 de maio de 2007

1. Crie uma função que receba um ponteiro para a raiz de uma árvore binária de busca, e dois valores. A função deve retornar o ancestral comum mais próximo dos dois valores. Caso algum dos valores não esteja na lista, retorne NULL.
2. Crie uma função que receba um ponteiro para a raiz de uma árvore binária de busca, e, a partir dela, construa uma árvore arbitrária (primeiro filho/irmão), devolvendo um ponteiro para sua raiz.



3. Crie uma função que receba um ponteiro para a raiz de uma árvore binária de busca e um índice i entre $0 \dots n - 1$, onde n é o número de elementos da árvore, e devolva o i -ésimo menor elemento da árvore.
4. Suponha que você precise executar o algoritmo da questão 3 várias vezes. Quais mudanças você faria na estrutura do árvore para que conseguisse a resposta mais rapidamente? Sugira também mudanças nas funções de inserção e remoção de elementos na árvore.
5. Crie uma função que receba um ponteiro para a raiz de uma árvore binária de busca e torne-a uma árvore binária de busca com fios. Considere que a estrutura dos nós já contém os ponteiros *próximo* e *anterior*, e que eles inicialmente apontam para NULL.

6. Escreva funções que imprimam os valores de uma árvore arbitrária (primeiro filho/irmão) em pré-ordem e pós-ordem. Considere que você tem uma função `IMPRIMEVALOR(No * p_no)` que faça a impressão.



Saída pré-ordem: 10-5-7-15-13-12-17

Saída pós-ordem: 7-5-12-13-17-15-10

Algumas notas finais:

- Para todos os problemas você deve alocar os nós através da função `ALOCANO` e liberá-los com a `LIBERANO`. Considere que você sempre consegue alocar os nós.
- Em momento algum será necessário fazer `MALLOCS`, ou tratamento de erro de entrada.
- Se precisar usar alguma outra estrutura de dados como pilha, fila, fila dupla ou listas, não precisa implementá-la.
- Coloque na resposta somente a função pedida, e as auxiliares que ela possa precisar. Não é necessário o código de todo o programa.
- Faça comentários em seu código para explicar o que não for óbvio. Exemplo: você não precisa explicar quando está alocando um novo nó. Mas é bom que explique quando está procurando por algo na lista.
- Evite entregar exercícios em branco. Se não conseguir resolver por completo, faça um *pseudo-rascunho-código* do que pensou.
- Tente resolver todos os exercícios da maneira mais eficiente possível. Todos têm soluções curtas e rápidas.
- Entregue sua lista preferencialmente em papel, em vez de usar a submissão online.