Aula em laboratório - Árvores (20/10/2014)

Codifique as operações solicitadas para os respectivos TADs (Árvore Binária e Árvore Binária de Busca), e faça um pequeno programa que utiliza e testa as funções implementadas. Não é necessário fazer entrada de dados via teclado. Faça um programa de teste determinístico. Uma implementação parcial de outras operações do TAD está disponível na Wiki (nas aulas de 10-11 e 13-11): você deve utilizá-la para as declarações e operações necessárias para criar e imprimir a árvore.

Ao final da aula, envie todos os códigos feitos em um arquivo zip, com o título:

"Aula 17/11 - [Nº Usp]" , em que [NºUsp] é o seu número usp

Considere o TAD Árvore Binária (declaração abaixo)

```
typedef int elem;

typedef struct bloco {
    elem info;
    struct bloco *esq, *dir;
} no;

typedef no *Arvore;
```

1) Escreva o código da função que busca na árvore o pai de uma chave dada, e retorna um ponteiro para o nó pai. Protótipo: no* busca_pai(Arvore, elem);

Considere o TAD Árvore Binária de Busca (declaração abaixo):

```
typedef int elem;
typedef struct bloco {
    elem info;
    struct bloco *esq, *dir;
} no;

typedef no *ABB;
```

2) Escreva o código da função de busca em versão não recursiva.
   Protótipo: no* buscar(ABB,elem);

3) Escreva o código de uma função que busca a maior chave em uma árvore dada, **não vazia**, retornando esse elemento. Protótipo: elem busca_maior(ABB);

4) Escreva o código da função que remove uma chave da árvore (se a remoção não ocorrer, retorna um flag de erro). (Veja o texto a seguir) Protótipo: int remover(ABB*,elem);

Casos a serem tratados na remoção:

Caso 1: o nó a ser removido é folha (remover, p.ex., chave T ou chave F na árvore abaixo): remove-se o nó (libera a memória) e o ponteiro que apontava para ele deve receber NULL.

Caso 2: o nó a ser removido tem um único filho (p.ex., remover C ou G): remove-se o nó (libera a memória), e seu filho é "puxado" para o lugar do pai.

Caso 3: o nó a ser removido tem 2 filhos (p.ex., remover E, P ou K): localiza o nó com a maior chave da sub-árvore esquerda do nó, que recebe o valor dessa chave, e remove o nó aonde estava a maior chave.