Universidade Federal do Espírito Santo – Departamento de Informática
Est. de Informação (INF02827) & Est. de Dados (INF01906)
3º Trabalho Prático
Período: 2008/2
Profª Patrícia Dockhorn Costa
Email: pdcosta@inf.ufes.br

Data de Entrega: 26/11/2008 ***Grupos de 2 pessoas***

Este trabalho tem como objetivo praticar o uso de tipos abstratos de dados e estruturas do tipo Árvore.

Regras Importantes

- Não é tolerado plágio. Trabalhos copiados serão penalizados com zero.
- A data de entrega é inadiável. Atrasos não serão tolerados neste trabalho.

Material a entregar

- Impresso: Documentação do trabalho, que deve conter:
  - Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
  - Implementação: descrição sobre a implementação do programa. Deve ser detalhada a estrutura de dados utilizada (de preferência com diagramas ilustrativos), o funcionamento das principais funções utilizadas incluindo pré e pós condições, o formato de entrada e saída de dados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado. Modularize o seu programa como discutido em sala de aula.
  - Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
  - Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso.

- Por email (pdcosta@inf.ufes.br):
  - O assunto da mensagem deve ser ed:trab3:<nome1>:<nome2>
    - Por exemplo: ed:trab3:<joaosilva>:<mariacosta>
  - Documentação do trabalho (em formato PDF).
  - Todos os arquivos .c e .h criados (exigido código muito bem documentado!).
  - O makefile.
  - Favor nomear os arquivos da seguinte maneira: TadArvore.h, TadArvore.c, Gerenciador.c.

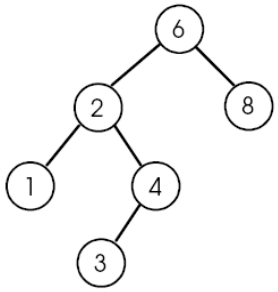
Gerenciador de Árvores Binárias de Busca

Neste trabalho é pedido um programa para gerenciar o ciclo de vida de uma Árvore Binária de Busca de valores inteiros.

Como discutimos em sala de aula, em uma Árvore binária de busca:

- o valor associado à raiz é sempre maior que o valor associado a qualquer nó da sub-árvore à esquerda (sae);
- o valor associado à raiz é sempre menor ou igual (para permitir repetições) que o valor associado a qualquer nó da sub-árvore à direita (sad);
- quando a árvore é percorrida em ordem simétrica (sae - raiz - sad), os valores são encontrados em ordem não decrescente.

Exemplo de árvore binária de busca:



A operação de busca em uma árvore binária de busca explora a propriedade de ordenação da árvore. Portanto, possui desempenho computacional proporcional à altura (O(log n) para o caso de árvore balanceada). Este desempenho não é obtido quando a árvore estiver degenerada.

Funcionalidades do Gerenciador

O seu programa gerenciador deverá:

- Criar uma árvore binária de busca;
- Incluir elementos na árvore binária de busca;
- Excluir elementos da árvore binária de busca;
- Imprimir a árvore em pré-ordem, in-ordem e em pós-ordem. Use a notação com "<r <sa> <sa> >", como vimos em sala de aula;
- Buscar um dado elemento na árvore (usando a propriedade da árvore binária de busca);
- Balancear a árvore (use algoritmo apresentado em sala de aula);
- Destruir a árvore, liberando a memória utilizada.

O seu programa (Gerenciador.c) deverá ler os dados de entrada a partir de um arquivo, cujo nome é passado como parâmetro na linha de comando (faz parte do trabalho descobrir como manipular arquivos e strings em C). Exemplo de execução do programa a partir da linha de comando:

```
Gerenciador entrada.txt
```

O arquivo de entrada é basicamente uma lista de comandos (um por linha) em formato texto. O último comando é a palavra FIM, que indica o final do arquivo. O formato a ser usado é definido a seguir.

Formato arquivo de entrada

O arquivo de entrada consiste de uma série de comandos, um em cada linha. Os possíveis comandos do arquivo de entrada, são:

- INCLUI *elem*: Inclui um elemento *elem* (inteiro) na árvore binária de busca;
- EXCLUI *elem*: Exclui o elemento elem da árvore;
- IMPRIME PREORDEM: Imprime o conteúdo da árvore em pre-ordem;
- IMPRIME INORDEM: Imprime o conteúdo da árvore em in-ordem ;
- IMPRIME POSORDEM: Imprime o conteúdo da árvore em pos-ordem;
- BUSCA elem: Busca um determinado elemento na árvore. Caso seja encontrado, imprime o elemento no arquivo de saída. Caso não seja encontrado, imprime a mensagem de erro "elemento *elem* não encontrado";
- BALANCEAR: Faz balanceamento da árvore binária de busca, usando o algoritmo apresentado em sala de aula.

Exemplo de arquivo de entrada:

```
INCLUI 10
INCLUI 9
```

```
INCLUI 8
INCLUI 5
INCLUI 4
IMPRIME PREORDEM
EXCLUI 8
IMPRIME PREORDEM
BUSCA 7
BUSCA 9
BUSCA 8
IMPRIME POSORDEM
BALANCEAR
IMPRIME PREORDEM
FIM
```

Formato arquivo de saída

O arquivo de saída (“saida.txt”) deve conter as informações de saída das operações de imprimir e de busca. Para o arquivo de entrada dado como exemplo, obtemos o seguinte arquivo de saída:

```
<10 <9 <8 <5 <4 <> <>> <>> <>> <>> <>>
<10 <9 <5 <4 <> <>> <>> <>> <>>
elemento 7 não encontrado
9
8
< < < < < <> <> 4> <> 5> <> 8> <> 9> <> 10>
<8 <5 <4 <> <>> <> > <9 <> <10 <> <>>>>
```

BOM TRABALHO!