Data de Entrega e Apresentação: 02/07/2018
Valor: 30 pontos

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

Material a entregar

Impresso: Documentação do trabalho, que deve conter:
(Valor: 5 pontos)

- 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 (principalmente o Insere e Remove), 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.

Via portal:

- O arquivo compactado contendo:
  - Documentação do trabalho (em formato PDF).
  - Todos os arquivos .c e .h criados (código bem documentado!). Favor nomear os arquivos da seguinte maneira: TadArvore.h, TadArvore.c, Gerenciador.c. (Valor: 25 pontos)

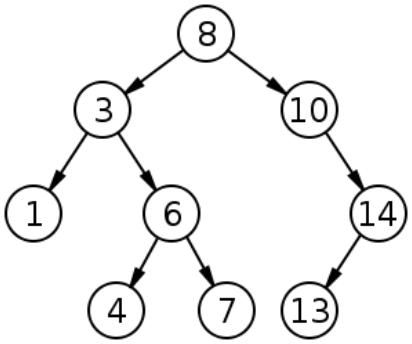
Gerenciador de Árvores Binárias de Busca

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 subárvore à esquerda (sae);
- o valor associado à raiz é sempre menor ou igual (para permitir repetições) que o valor associado a qualquer nó da subárvore 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; (1 ponto)
Incluir elementos na árvore binária de busca e mantê-la balanceada através das propriedades da AVL; (7 pontos)
Excluir elementos da árvore binária de busca e manter a árvore balanceada; (7 pontos)
Imprimir a árvore em detrimento da escolha de um dos percursos: pré-ordem, in-ordem e pós-ordem. (3 pontos)
Buscar um dado elemento na árvore (usando a propriedade da árvore binária de busca); (1 pontos)
Destruir a árvore, liberando a memória utilizada. (1 ponto)

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.
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. (2 pontos)

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".

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
IMPRIME INORDEM
FIM
```

Formato arquivo de saída

O arquivo de saída ("saida.txt") deve conter as informações de saída das operações de imprimir e de busca (3 pontos). Para o arquivo de entrada dado como exemplo, obtemos o seguinte arquivo de saída:

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

Obs.: O formato para a impressão da árvore no percurso INORDEM poderá ser modificado a fim de apresentar uma melhor visão da árvore binária **(ponto extra)**.

BOM TRABALHO!