

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL

Algoritmos e Estruturas de Dados I – 2023/2

Trabalho III

Entrega em 02/12/2023 até às 23:00, via Moodle

Prof. Marcio Sarroglia Pinho

Balanceamento de Árvores Binária de Pesquisa

Instruções e Objetivo

O objetivo deste trabalho, que deve ser **feito em DUPLA**, é o estudo, o entendimento e a implementação do balanceamento em árvores binárias de pesquisa, com base no critério de árvores AVL. Cada dupla deverá **fazer todas as tarefas solicitadas a seguir conforme as orientações**.

Descrição

Inicialmente, deverá ser feito um estudo do funcionamento da árvore de pesquisa AVL. A ideia é criar árvores através da inserção de nodos (método *add*), como no exemplo a seguir.

O conteúdo sobre árvores AVL pode ser encontrado nos livros indicados na bibliografia da disciplina, lembrando que o livro "Estruturas de dados e algoritmos em Java", de Goodrich e Tamassia, está disponível online e aborda este conteúdo. Pesquisas na Web em sites confiáveis, tais como sites de universidades, podem ajudar no entendimento do conteúdo.

Para desenvolver o trabalho **devem ser usados os códigos de Árvores Binárias de Pesquisa** disponíveis no Moodle.

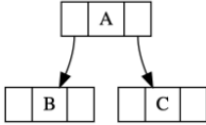
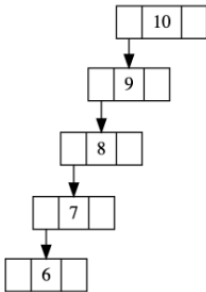
O programa deve aplicar o balanceamento **em um nodo específico**, passado por parâmetro para um método chamado **ApplyBalancing**.

```
ABP B;  
B.add(10);  
B.add(6);  
B.add(12);  
B.add(11);  
for(int i=15;i<25;i++)  
    B.add(i);  
  
B.GeraDOT();  
B.ApplyBalancing(6);  
B.GeraDOT();
```

Antes e depois do processo de balanceamento, devem ser geradas imagens da árvore através do método **GeraDot** que gera o arquivo no formato DOT, do GraphViz, conforme visto em aula. Este arquivo permite a exibição das árvores através de ferramentas como:

- <https://dreampuf.github.io/GraphvizOnline>
- <http://www.webgraphviz.com/>
- <http://viz-js.com/>

Veja a seguir exemplos de exibições geradas a partir destas ferramentas.

Código	Imagem
<pre> digrph g { node [shape = record,height=.1]; nodeA[label = "<esq> A <dir> "]; nodeB[label = "<esq> B <dir> "]; nodeC[label = "<esq> C <dir> "]; "nodeA":esq -> "nodeB"; "nodeA":dir -> "nodeC"; } </pre>	
<pre> digrph g { node [shape = record,height=.1]; node6[label = "<esq> 6 <dir> "] node7[label = "<esq> 7 <dir> "] node8[label = "<esq> 8 <dir> "] node9[label = "<esq> 9 <dir> "] node10[label = "<esq> 10 <dir> "] "node7":esq -> "node6" "node8":esq -> "node7" "node9":esq -> "node8" "node10":esq -> "node9" } </pre>	

Ao realizar o balanceamento de um nodo o método deve informar qual o tipo de balanceamento deve ser realizado no nodo.

Caso o balanceamento seja Direita-Direita ou Esquerda-Esquerda, basta a execução de um passo do balanceamento.

Caso o balanceamento seja Direita-Esquerda ou Esquerda-Direita, este passo deve ser seguido da a execução de um passo de balanceamento Direita-Direita ou Esquerda-Esquerda, conforme o caso.

Após a realização de um destes passos deve ser possível chamar novamente o método **ApplyBalancing** para qualquer um dos nodos da árvore.

Elaboração do Trabalho

O desenvolvimento do trabalho consta das seguintes etapas:

1. Estudar a árvore de pesquisa AVL;
2. Implementar o balanceamento de um nodo da árvore de acordo com a interface fornecida;

Após o entendimento do funcionamento da árvore AVL, é preciso implementá-la conforme o código disponível no Moodle. Portanto, deve ser implementada uma árvore de pesquisa para armazenar números inteiros usando o código passado e incluindo também os seguintes métodos:

- *applyBalancing(Integer e)*: para fazer o balanceamento do nodo da árvore que contém 'e';

Entrega

- **O trabalho deve ser entregue no Moodle através de um arquivo zip contendo:**
 - Arquivos com o código fonte;
 - Relatório com a execução de testes de balanceamento da árvore, contendo a árvore antes e a depois do processo de balanceamento. Este relatório deve conter o link para um vídeo de, no máximo, 2 minutos com a explicação do funcionamento do trabalho.

Observações:

- Os trabalhos que **NÃO FOREM ENTREGUES** através do Moodle seguindo as regras estabelecidas, até os dias e horários especificados, não serão considerados!
- As implementações que apresentarem erro de compilação não serão consideradas.
- Trabalhos que apresentarem cópias das soluções de outros colegas ou de outras fontes resultarão em nota zero para todos os alunos envolvidos.
- Todos os alunos devem apresentar o trabalho e estarem aptos a responder às perguntas sobre os algoritmos implementados.
- Cada dupla é responsável pela execução de cada tarefa. **Não serão considerados trabalhos que não cumprirem as regras estabelecidas, ou se forem disponibilizados arquivos errados ou corrompidos.**