



## Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Departamento de Sistemas e Computação

Graduação em Ciência da Computação

### Exercício sobre árvore binária de busca

**Objetivo:** Praticar a implementação de árvores binárias de busca.

O endereço do sistema de submissão é o <https://les.dsc.ufcg.edu.br:8443/EasyLabCorrection>.

Relembre o conceito de árvore binária de busca visto em sala de aula.

Atividades necessárias antes de iniciar o exercício:

1. Crie um projeto no Eclipse chamado LEDA, por exemplo (pode ser qualquer outro nome que lhe convier);
2. Descompacte o arquivo baixado (exceto o PDF) na pasta dos fontes (normalmente **src**) do seu projeto LEDA criado no seu workspace. O arquivo baixado tem a seguinte estrutura:
  - adt
  - bt
  - BTNode.java (CLASSE REPRESENTANDO UM NÓ DE UMA ÁRVORE BINÁRIA GÊNÉRICA)
  - BT.java (INTERFACE DE UMA ÁRVORE BINÁRIA GÊNÉRICA)
  - bst
  - BSTNode.java (CLASSE REPRESENTANDO UM NÓ DE UMA BST)
  - BST.java (INTERFACE DE UMA BST GÊNÉRICA)
  - BSTImpl.java (**IMPLEMENTAÇÃO PARCIAL DE UMA BST**)
3. No Eclipse, selecione a pasta dos fontes no projeto LEDA e faça um refresh (apertar F5). Note que deve aparecer um pacote adt.bst contendo os arquivos mencionados acima.

Agora você está pronto para começar a trabalhar nas seguintes atividades:

1. Observe a interface BT.java. Ela descreve os serviços de uma BT genérica.
2. Observe a interface BST.java. Ela descreve os serviços de uma BST genérica.
3. Observe também a existência de implementação incompleta BSTImpl. Você precisa implementar os métodos incompletos.
4. Concentre-se em implementar conforme descrito na interface e pense em cenários para testar suas implementações. Alguns cenários interessantes são: testar inserções em árvores não vazias e em árvores vazias, testar remoções em árvores não vazias e vazias, verificar se o tamanho da árvore é alterado com inserções e remoções com sucesso, verificar se a altura da árvore muda (e como muda) com inserções e remoções, etc. O importante de testar sua árvore é ter em mente que sua estrutura está correta. Isso você pode verificar com os métodos de percurso da árvore.

### Instruções para o envio

Ao terminar o exercício, faça os seguintes passos:

1. Compacte a pasta **adt** que existe nos fontes de seu projeto LEDA (**src**) e retire suas classes de teste desse arquivo compactado. A compactação DEVE ser feita a partir do diretório raiz de seus fontes de forma a preservar a estrutura de pastas que refletem a estrutura dos pacotes (package) Java. Por exemplo, voce deve ter um arquivo compactado NOME\_COMPLETO\_DO\_ALUNO.ZIP com a seguinte estrutura:  
- adt  
-- bst  
--- BSTImpl.java (**IMPLEMENTACAO PARCIAL DE UMA BST**)
2. Envie esse arquivo com sua solução para o sistema de submissão e verifique que o contador de submissões será alterado.

#### **Observações finais:**

- **A interpretação do exercício faz parte da atividade.**
- **A atividade é individual. A conversa entre alunos é proibida.**
- **É proibido coletar códigos prontos e adaptar. Implemente as questões. Isso é para seu aprendizado.**
- **Caso voce observe qualquer problema no sistema de submissão, contacte o professor imediatamente.**
- **Se voce nao compactar o arquivo seguindo a estrutura de diretórios a compilação não terá sucesso e o sistema mostrará isso. Erro de compactação serão de responsabilidade do aluno. O professor não ajudará o aluno nesse item. É só seguir as instruções deste arquivo.**