DCC013 - Estrutura de Dados / DCC107 - Laboratório de Programação II

Árvore binária de busca

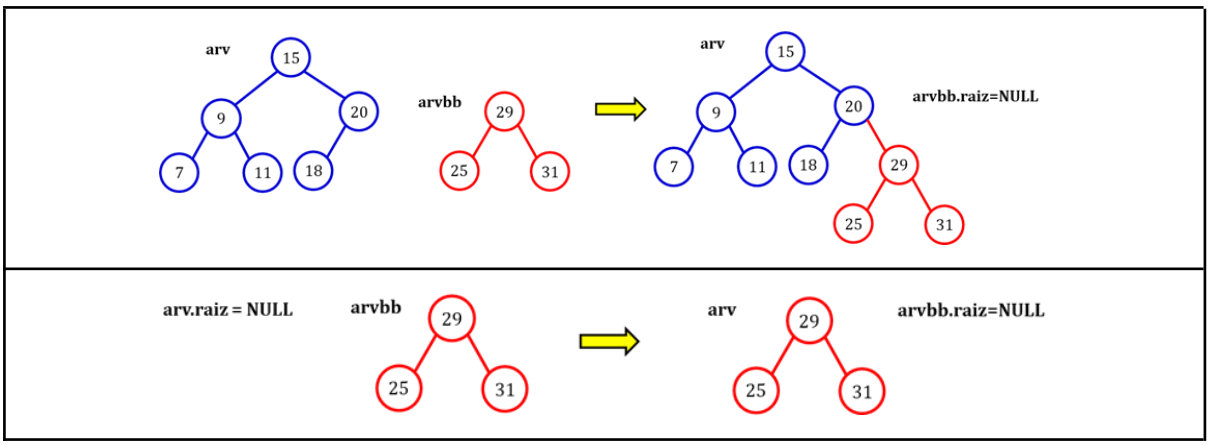
Você pode utilizar qualquer ambiente de programação para desenvolver sua atividade. Ao final, **copie e cole o seu código-fonte com a resposta aqui mesmo neste documento**, dentro dos espaços indicados para isso e **preservando a identação do código**. **Depois que terminar sua avaliação, não se esqueça de entregar sua atividade!** Fique atento ao relógio, pois as atividades entregues com atraso não serão aceitas.

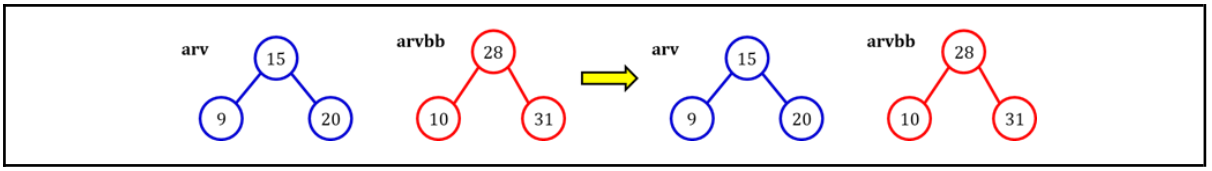
Para resolver esta atividade, **clique aqui para baixar o projeto da aula de laboratório de programação 2, que contém a implementação do TAD ArvBinBusca para árvore binária de busca de números inteiros.** Na sua solução para a questão abaixo, você pode utilizar/chamar qualquer uma das operações que estejam disponíveis no projeto (exatamente do jeito que ele se encontra no site da disciplina). Outras operações que você venha a criar para resolver o seu exercício, **inclusive as operações auxiliares**, devem ser copiadas para sua resposta neste documento. **Não cole código desnecessariamente**.

A) Desenvolver uma operação `bool menores(int val)` para, dado o inteiro `val`, retornar `true` se `val` é menor que todos os valores dos nós da árvore binária de busca (ABB) ou se a ABB é vazia. Retornar `false` caso contrário.

B) Dado um ponteiro `arvbb` para uma ABB, usando a operação `menores()`, desenvolver a operação `void insereAbbDireita(ArvBinBusca* arvbb)` para inserir `arvbb` como subárvore à direita do nó mais à direita da ABB, se possível (continuar sendo ABB). Se a `arvbb` for adicionada, ela deve tornar-se vazia no fim da operação. **Não usar a operação `auxInsere`** implementada na ABB.

Usar a **propriedade de ABB** para visitar a quantidade mínima de nós. A figura a seguir exibe 3 exemplos da operação `arv.insereAbbDireita(arvbb)` para 2 ABB.






```
// Cole aqui sua resposta
```