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.

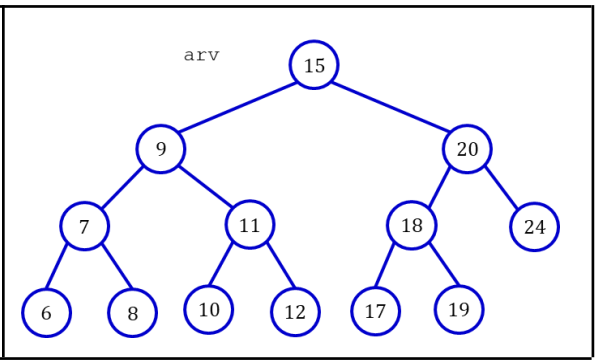
Para resolver esta atividade, **clique aqui para baixar o projeto da aula de laboratório de programação 2, que contém a implementação do TAD ArvBin Busca para árvore binária de busca de números inteiros.** Na sua solução para a questão abaixo, você pode utilizar/chamar qualquer uma das operações que 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**.

O inteiro antecessor de `val` corresponde ao nó da ABB com o maior valor menor que `val`. Desenvolver uma operação `int antecessor(int val)` para, dado (parâmetro) um inteiro positivo `val`, determinar e retornar o valor do nó de uma **árvore binária de busca** (ABB) que corresponda ao antecessor de `val`.

Caso a ABB não possua o antecessor de `val`, retornar `-1`. Usar a **propriedade de ABB** para visitar a quantidade mínima de nós. Observar que `val` pode não ser valor de nenhum nó da ABB mas o antecessor tem que ser (se existir).
Dica: usar um parâmetro por referência para saber o valor do último nó visitado. Seu valor inicial deve ser `-1`.

A figura a seguir exibe exemplos da operação para a ABB `arv` (mesma do projeto no site).

```
-  arv.antecessor(5) = -1
-  arv.antecessor(16) = 15
-  arv.antecessor(11) = 10
-  arv.antecessor(6) = -1
```




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