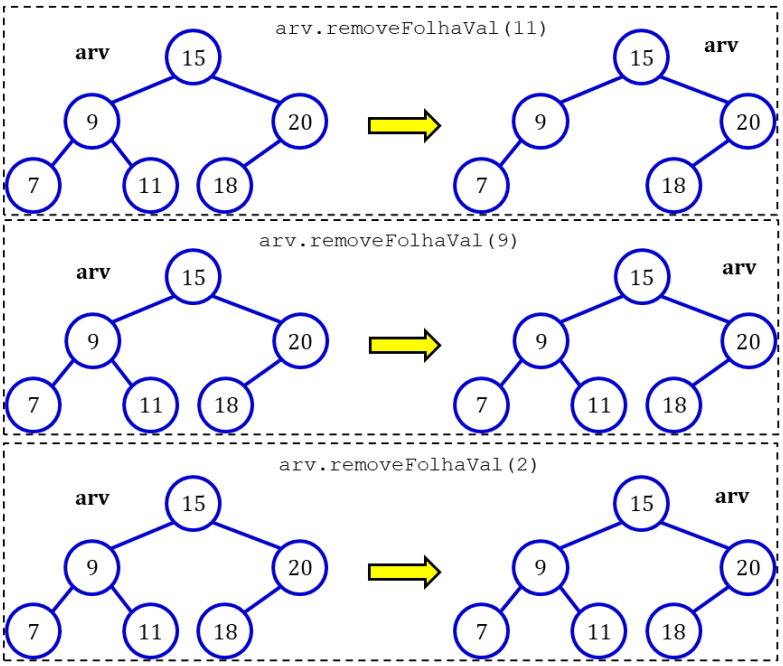
DCC013 - Estrutura de Dados / DCC107 - Laboratório de Programação II

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**.

Desenvolver uma operação **NÃO RECURSIVA** `void removeFolhaVal(int val)` para, dado (parâmetro) um inteiro positivo `val`, remover a folha com valor `val` de uma **árvore binária de busca** (ABB). Emitir uma mensagem caso o valor `val` não se encontre na ABB ou se não há folha com o valor `val`. Usar a **propriedade de ABB** para visitar a quantidade mínima de nós da ABB.
A figura a seguir exibe exemplos da operação para a ABB `arv`.




```
void ArvBinBusca::removeFolhaVal(int val)
{
```

```
    NoArv *p = raiz;
    NoArv *aux = NULL;

    while (p != NULL)
    {
        if (val < p->getInfo()) // Se val é menor, vamos para a esquerda
        {
            aux = p;
            p = p->getEsq();
        }
          else if (val > p->getInfo()) // Se val é maior, vamos para a
direita
        {
            aux = p;
            p = p->getDir();
        }
        else if (p->getInfo() == val) // Achamos o nó com info == val
        {
             if (p->getEsq() == NULL && p->getDir() == NULL) // Saber se
é folha
            {
                    (aux->getDir() == p) ? aux->setDir(removeFolha(p)) :
aux->setEsq(removeFolha(p));
                return;  // Verificação para saber se é filho a esquerda
ou direita, e removemos ele
            }
            else
            {
                cout << "Não há folha com valor val" << endl;
                  return; // Se temos um nó com info == val e ele não é
folha, não removemos
            }
        }
    }

     cout << "Valor não encontrado na ABB" << endl; // Caso não haja
nenhum nó com info == val
    return;
}
```