

2) Implemente os seguintes métodos na classe CarvBin (você deve escolher 6 métodos - os marcados com * são obrigatórios)

a. O método recursivo que retorne o menor elemento de uma árvore binária e o método recursivo que retorne o maior elemento de uma árvore binária

b. O método que retorne o menor elemento de uma subárvore (dado o valor da chave do nó raiz dessa subárvore) e o método que retorne o maior elemento de uma subárvore (dado o valor da chave do nó raiz dessa subárvore)

```
/* Chama a funcao menorValor da árvore */
public int menorValorArvore() {
    return menorValor(raiz);
}

/* Retorna o menor valor em uma subárvore */
private int menorValor(CNo no) {
    return no.esquerda == null ? no.item : menorValor(no.esquerda);
}

/* Chama a funcao maiorValor da árvore */
public int maiorValorArvore() {
    return maiorValor(raiz);
}

/* Retorna o maior valor de uma subárvore */
private int maiorValor(CNo no) {
    return no.direita == null ? no.item : maiorValor(no.direita);
}
```

c. (*) O método recursivo que retorne a altura de um nó

(lembre-se que a altura de um nó-folha = 1)

```
/* Chama função que retorna a altura do nó cujo o valor foi informado */
public int altura(int valor) {
    return alturaRecursiva(retornaNo(raiz, valor));
}

/* Retorna a altura da árvore a partir de um nó */
private int alturaRecursiva(CNo no) {
    if (no == null)
        return 0;
    return 1 + max(alturaRecursiva(no.esquerda), alturaRecursiva(no.direita));
}
```

d. (*) O método recursivo que retorne a profundidade de um nó

(lembre-se que a profundidade do nó raiz = 1)

```
/* Chama a funcao de profundidade recursiva */
public int profundidade(int valor) {
    return profundidadeRecursiva(raiz, valor);
}

/* Funcao retorna a profundidade de um nó */
private int profundidadeRecursiva(CNo no, int valor) {
    if (no == null)
        return 0;
    if (no.item == valor)
        return 1;
    return 1 + (valor < no.item ? profundidadeRecursiva(no.esquerda, valor) :
    profundidadeRecursiva(no.direita, valor));
}
```

f. O método que retorne a quantidade de descendentes de um nó passado por parâmetro.

```
/* Funcao que chama a funcao de quantidade de filhos de uma arvore */
public int quantidadeFilhos(int valor) {
    return qtdFilhos(retornaNo(raiz, valor)) - 1;
}

/* Determina a quantidade de filhos de um no */
private int qtdFilhos(CNo no) {
    if (no == null)
        return 0;
    return 1 + qtdFilhos(no.esquerda) + qtdFilhos(no.direita);
}
```

j. (*) O método que imprime para cada nó as seguintes informações: seu valor, altura, profundidade, seu pai (se houver), quantidade de filhos, valor do filho à esquerda (se houver), valor do filho à direita (se houver), se o nó é intermediário ou nó folha.

```
/* Metodo que chama imprime completo */
public void imprimeCompleto() {
    imprimeCompletoStr(raiz, null);
}

/* Método imprime diversas informações sobre no */
private void imprimeCompletoStr(CNo no, CNo pai) {
    if (no != null) {
        imprimeCompletoStr(no.esquerda, no);
        System.out.println("Valor: " + no.item);
        System.out.println("Altura: " + alturaRecursiva(no));
        System.out.println("Profundidade: " + profundidadeRecursiva(raiz, no.item));
        System.out.println("Nó Pai: " + (pai == null ? "Não possui" : pai.item));
        System.out.println("Quantidade de filhos: " + quantidadeFilhos(no.item));
        System.out.println("Nó a esquerda: " + (no.esquerda == null ? "Não possui" :
no.esquerda.item));
        System.out.println("Nó a direita: " + (no.direita == null ? "Não possui" :
no.direita.item));
        System.out.println(eFolhaFinal(no) ? "É folha" : "Não é folha");
        System.out.println();
        imprimeCompletoStr(no.direita, no);
    }
}

/* Chama a funcao eFolha */
public boolean eFolha(int valor) {
    return eFolhaFinal(retornaNo(raiz, valor));
}

/* Retorna se true caso o no seja folha e false caso seja intermediario */
private boolean eFolhaFinal(CNo no) {
    return no.esquerda == null && no.direita == null;
}
```

3) É possível saber a quantidade de nós em determinado nível de uma árvore? E a quantidade máxima de nós dada a altura de uma árvore? Quais são as equações necessárias?

R: