

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS E INFORMÁTICA
UNIDADE EDUCACIONAL PRAÇA DA LIBERDADE
Bacharelado em Engenharia de Software

Ravi Antônio Gonçalves de Assis

2ª LISTA DE EXERCÍCIOS DE ALGORITMOS E ESTRUTURAS DE DADOS II

Belo Horizonte

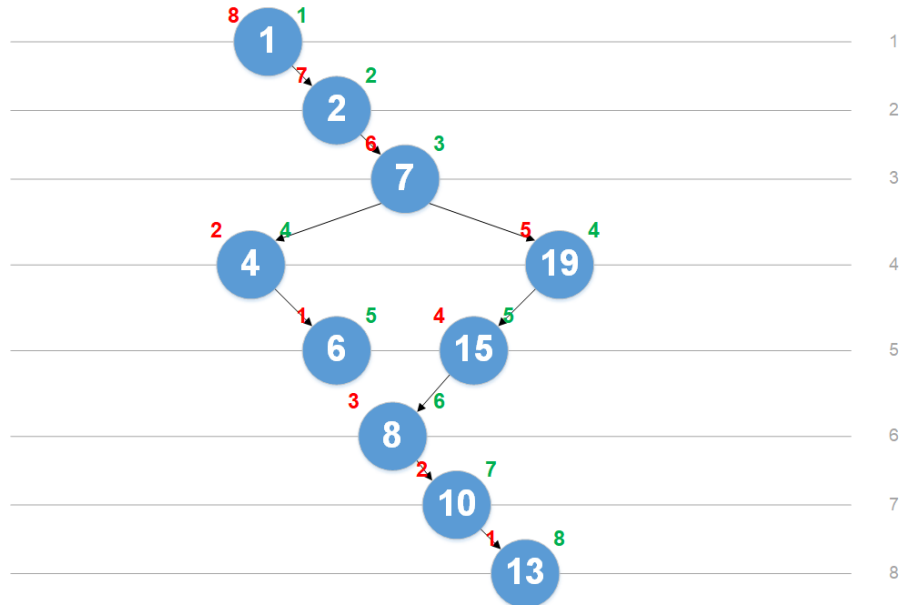
2018

1)

a)

i)

i.

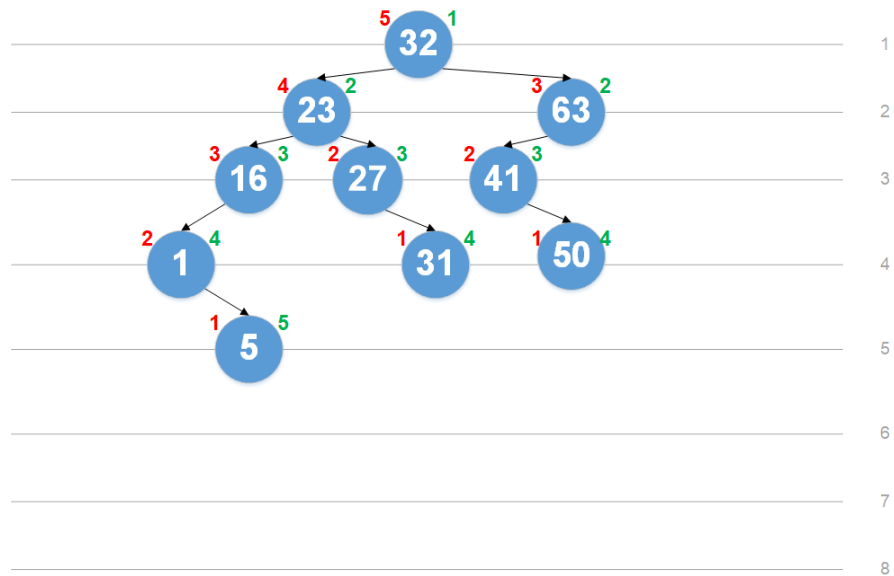


Pré-ordem: 1, 2, 7, 4, 6, 19, 15, 8, 10, 13

Pós-ordem: 6, 4, 13, 10, 8, 15, 19, 7, 2, 1

ii)

ii.

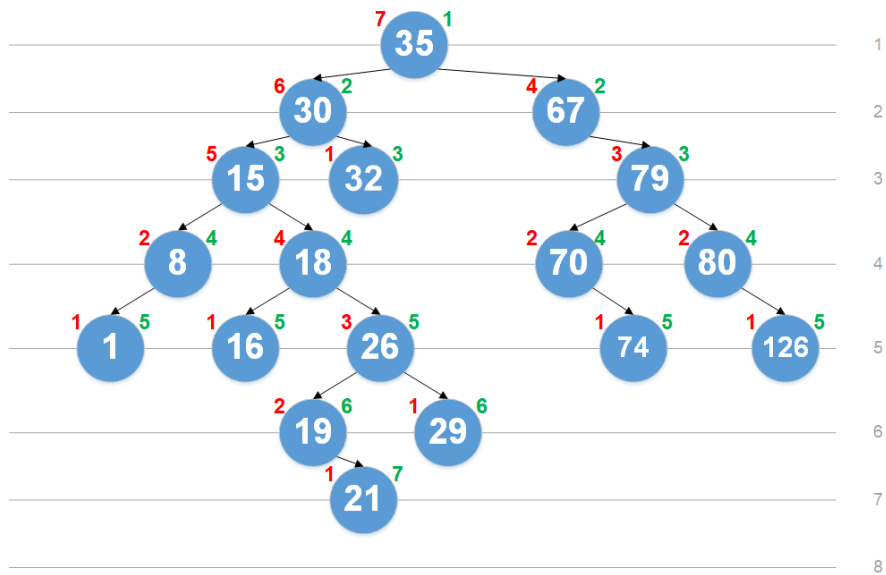


Pré-Ordem: 32, 23, 16, 1, 5, 27, 31, 63, 41, 50

Pós-ordem: 5, 1, 16, 31, 27, 50, 41, 63, 32

iii)

ii.

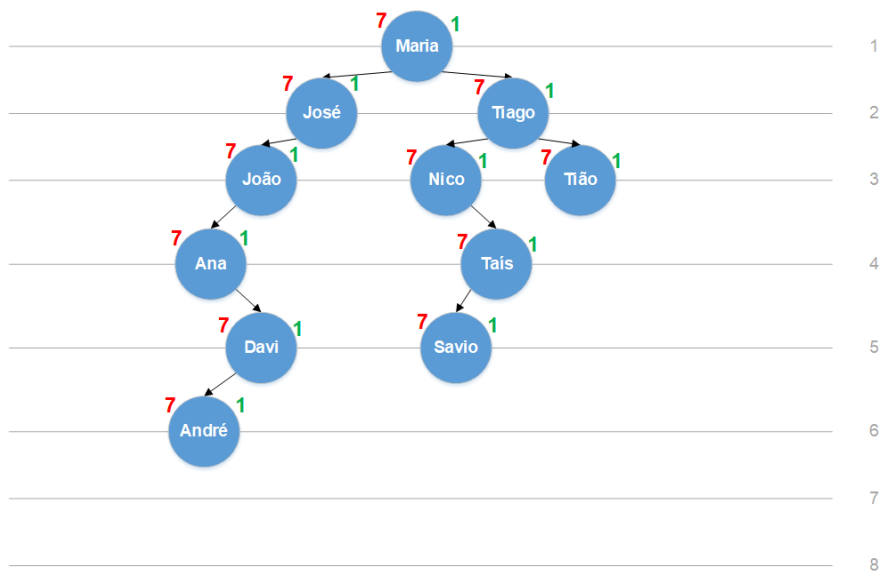


Pré-ordem: 35, 30, 15, 8, 1, 18, 16, 26, 19, 21, 29, 32, 67, 79, 70, 74, 80, 126

Pós-ordem: 1, 8, 16, 21, 19, 29, 26, 18, 15, 32, 30, 74, 70, 126, 80, 79, 67, 35

iv)

iv.

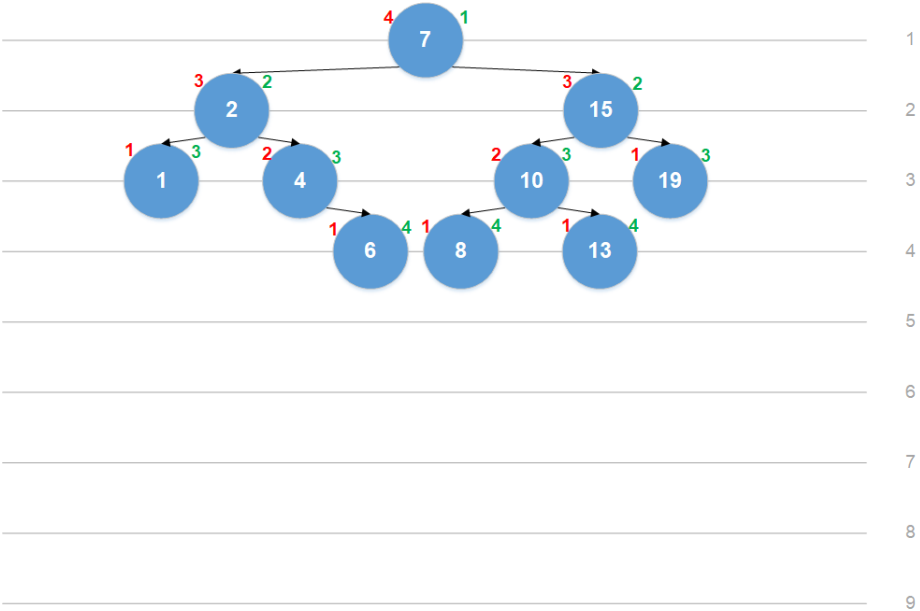


Pré-ordem: Maria, José, João, Ana, Davi, André, Tiago, Nico, Taís, Savio, Tião

Pós-ordem: André, Davi, Ana, João, José, Savio, Taís, Nico, Tião, Tiago, Maria

b)
i)

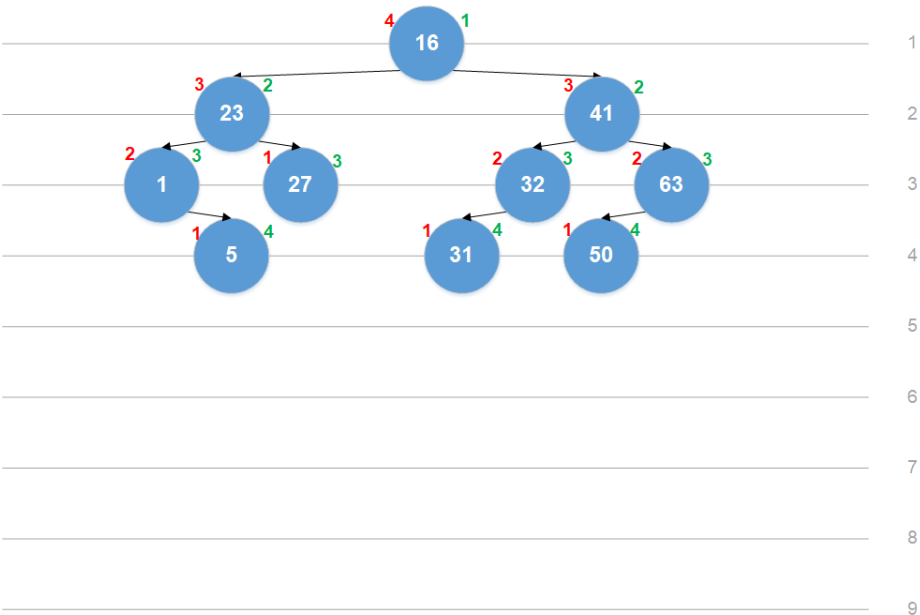
i.



Pré-ordem: 7, 2, 1, 4, 6, 15, 10, 8, 13, 19
Pós-ordem: 1, 6, 4, 2, 8, 13, 10, 19, 15, 7

ii)

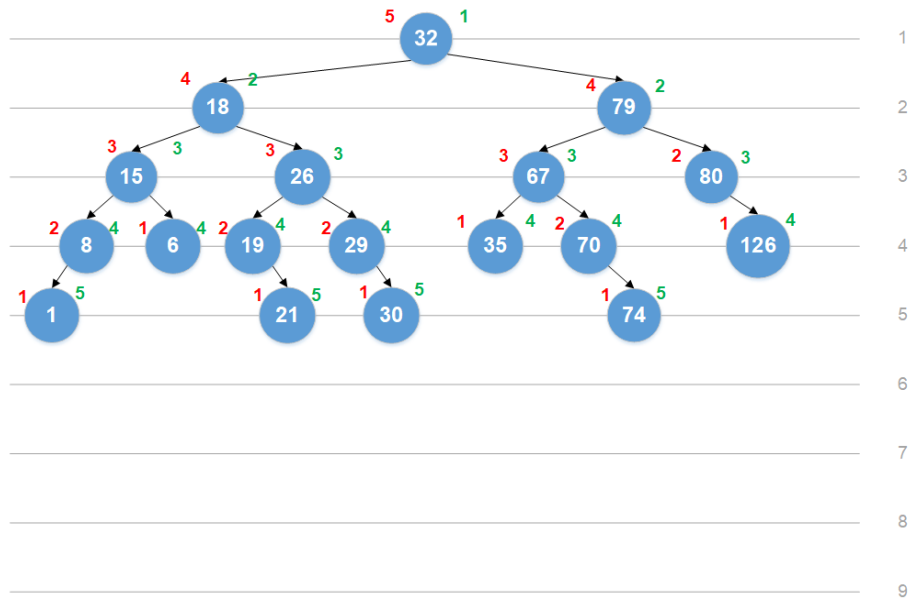
ii.



Pré-ordem: 16, 23, 1, 5, 27, 41, 32, 31, 63, 50
Pós-ordem: 5, 1, 27, 23, 31, 32, 50, 63, 41, 16

iii)

iii.

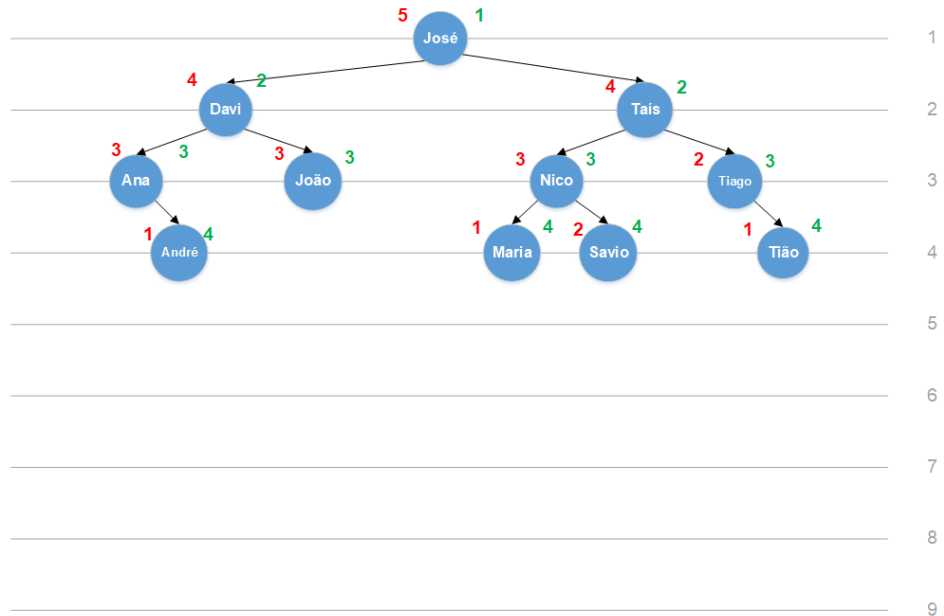


Pré-ordem: 32, 18, 15, 8, 1, 6, 26, 19, 21, 23, 30, 79, 67, 35, 70, 74, 80, 126

Pós-ordem: 1, 8, 6, 15, 21, 19, 30, 29, 26, 18, 35, 74, 70, 67, 126, 80, 79, 32

iv)

iv.



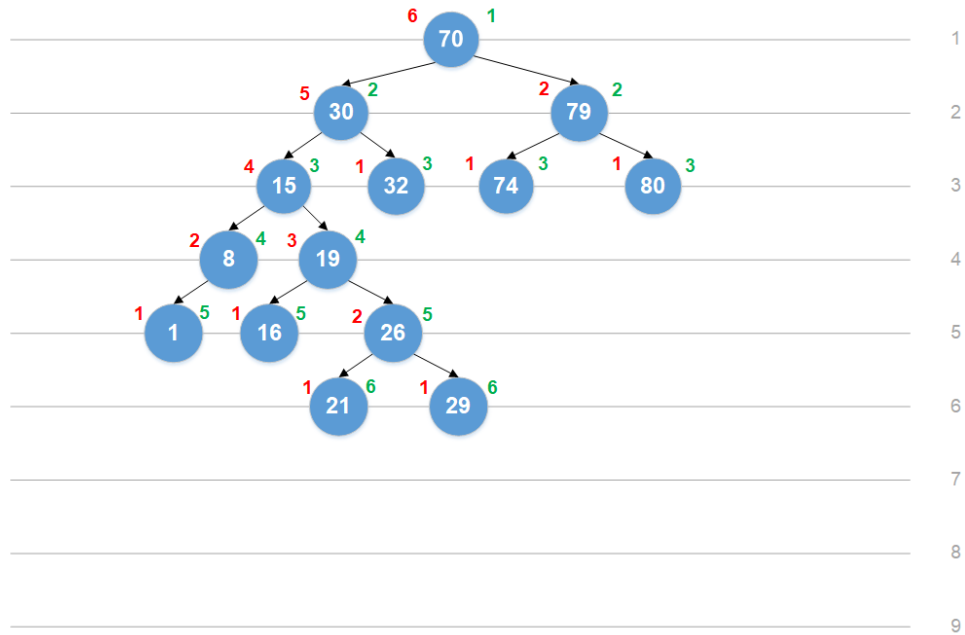
Pré-ordem: José, Davi, Ana, André, João, Taís, Nico, Maria, Savio, Tiago, Tião

Pós-ordem: André, Ana, João, Davi, Maria, Nico, Savio, Tião, Tiago, Taís, José

2)

a)

2-a



3)

```
public void preOrdemComFilhos(CNo no) {
    if (no != null) {
        int filhos = 0;
        if (no.esq != null) filhos++;
        if (no.dir != null) filhos++;
        System.out.println("Chave= " + no.item + " Filhos="
            + filhos);
        preOrdemComFilhos(no.esq);
        preOrdemComFilhos(no.dir);
    }
}
```

4)

```
/*Ex4*/
public void emOrdemComFilhos(CNo no) {
    if (no != null) {
        int filhos = 0;
        if (no.esq != null) filhos++;
        if (no.dir != null) filhos++;
        emOrdemComFilhos(no.esq);
        System.out.println("Chave= " + no.item + " Filhos="
            + filhos);
        emOrdemComFilhos(no.dir);
    }
}
```

5)

```
/*Ex5*/
public void posOrdemComFilhos(CNo no) {
    if (no != null) {
        int filhos = 0;
        if (no.esq != null) filhos++;
        if (no.dir != null) filhos++;
        posOrdemComFilhos(no.esq);
        posOrdemComFilhos(no.dir);
        System.out.println("Chave= " + no.item + " Filhos=
" + filhos);
    }
}
```

6)

```
/*Ex6*/
public void preOrdemComFilhos2(CNo no) {
    if (no != null) {
        System.out.print("Chave= " + no.item);
        if (no.esq != null) System.out.print(" Filho
esquerdo= " + no.esq.item);
        if (no.dir != null) System.out.print(" Filho
esquerdo= " + no.dir.item);
        System.out.println();
        preOrdemComFilhos2(no.esq);
        preOrdemComFilhos2(no.dir);
    }
}
```

7)

```
/*Ex7*/
public void emOrdemComFilhos2(CNo no) {
    if (no != null) {
        emOrdemComFilhos2(no.esq);

        System.out.print("Chave= " + no.item);
        if (no.esq != null) System.out.print(" Filho
esquerdo= " + no.esq.item);
        if (no.dir != null) System.out.print(" Filho
esquerdo= " + no.dir.item);
        System.out.println();

        emOrdemComFilhos2(no.dir);
    }
}
```

8)

```
/*Ex8*/
public void posOrdemComFilhos2(CNo no) {
    if (no != null) {
        posOrdemComFilhos2(no.esq);
        posOrdemComFilhos2(no.dir);
        System.out.print("Chave= " + no.item);
        if (no.esq != null) System.out.print(" Filho
esquerdo= " + no.esq.item);
        if (no.dir != null) System.out.print(" Filho
esquerdo= " + no.dir.item);
        System.out.println();
    }
}
```

9)

```
/*Ex9*/
public int altura(int item) {
    return altura(item, retornaNo(item, this.raiz), 1);
}

private int altura(int item, CNo no, int alt) {
    if(this.vazia()) return 0;
    else if (no == null) return alt;
    else {
        int altEsq = altura(item, no.esq, alt+1);
        int altDir = altura(item, no.dir, alt+1);
        return (altEsq <= altDir) ? altEsq : altDir;
    }
}
```

10)

```
/*Ex10*/
public int profundidade(int item) {
    return profundidade(item, this.raiz, 1);
}
private int profundidade(int item, CNo no, int prof) {
    // TODO Auto-generated method stub
    if(this.vazia() || no == null) return 0;
    else if (no.item == item) return prof;
    else if (item < no.item) return profundidade(item, no.esq,
prof+1);
    else if (item > no.item) return profundidade(item, no.dir,
prof+1);
    return 0;
}
```


11)

```
private CNo retornaNo(int item, CNo no) {
    if(this.vazia() || no == null) return null;
    else if (no.item == item) return no;
    else if (item < no.item) return retornaNo(item, no.esq);
    else if (item > no.item) return retornaNo(item, no.dir);
    return null;
}

public int fb(int item) {
    CNo subArv = retornaNo(item, this.raiz);
    if(this.vazia() || subArv == null) return 0;
    else {
        int altE, altD;
        altE = altD = 0;
        if(subArv.esq != null) altE =
altura(subArv.esq.item);
        if(subArv.dir != null) altD =
altura(subArv.dir.item);
        return altE - altD;
    }
}
```

12)

```
public void imprimeInfoNo() {
    imprimeInfoNo(this.raiz, 1);
}

private void imprimeInfoNo(CNo no, int nivel) {
    if (no != null) {
        int filhos = 0;
        filhos = (no.esq != null) ? filhos+1 : filhos;
        filhos = (no.dir != null) ? filhos+1 : filhos;
        System.out.print("* Nó " + no.item + " - Nível " +
nível + " - " + filhos + " filho(s) ");
        if( no.esq != null && no.dir != null)
            System.out.println(": " + no.esq.item + " e "
+ no.dir.item);
        else if (no.esq != null)
            System.out.println(": " + no.esq.item);
        else if (no.dir != null)
            System.out.println(": " + no.dir.item);
        else System.out.println();
        imprimeInfoNo(no.esq, nivel+1);
        imprimeInfoNo(no.dir, nivel+1);
    }
}
```