

ALGESD 09

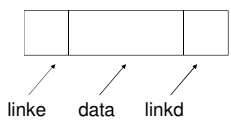
Árvores

Prof. Dr. Marcelo Duduchi

Árvores

- Árvores Binárias Ordenadas podem ser implementadas a partir de nodos iguais aos nodos das listas duplamente encadeadas onde os links apontam para os filhos à esquerda e à direita:

```
class Node {
    char data;
    Node linke, linkd;
}
```



Árvores Binárias Ordenadas

```
class ArvBinOrd {

    class Node {
        char data;
        Node linke, linkd;
    }

    Node raiz;

    ArvBinOrd(){
        raiz = null;
    }

    ...
}
```

Árvores Binárias Ordenadas

```
void incArv(char elem) {
    raiz = incAux(raiz, elem);
}

Node incAux(Node raiz, char elem){
    if(raiz == null){
        raiz = new Node();
        raiz.data = elem;
        raiz.linke = null;
        raiz.linkd = null;
    }
    else {
        if ( raiz.data > elem ) raiz.linke = incAux(raiz.linke, elem);
        else if ( raiz.data < elem ) raiz.linkd = incAux(raiz.linkd, elem);
        else System.out.println("Nome já existe!");
    }
    return raiz;
}
```

Árvores Binárias Ordenadas

```
void preOrdem(){
    System.out.print("PreOrdem: []");
    preOrdemAux(raiz);
    System.out.println("]");
}

void preOrdemAux(Node raiz){
    if(raiz != null){
        System.out.print(" "+raiz.data);
        preOrdemAux(raiz.linke);
        preOrdemAux(raiz.linkd);
    }
}
```

Árvores Binárias Ordenadas

```
void emOrdem(){
    System.out.print("EmOrdem: []");
    emOrdemAux(raiz);
    System.out.println("]");
}

void emOrdemAux(Node raiz){
    if(raiz != null){
        emOrdemAux(raiz.linke);
        System.out.print(" "+raiz.data);
        emOrdemAux(raiz.linkd);
    }
}
```

Árvores Binárias Ordenadas

```
void posOrdem(){
    System.out.print("PosOrdem: [");
    posOrdemAux(raiz);
    System.out.println("]");
}
void posOrdemAux(Node raiz){
    if(raiz != null){
        posOrdemAux(raiz.linke);
        posOrdemAux(raiz.linkd);
        System.out.print(" "+raiz.data);
    }
}
```

Árvores Binárias Ordenadas

```
boolean BuscaArv (char elem) {
    return BuscaAux(raiz, elem); }
boolean BuscaAux (Node raiz, char elem) {
    if (raiz != null) {
        if (raiz.data > elem)
            return BuscaAux (raiz.linke, elem);
        else
            if(raiz.data < elem)
                return BuscaAux (raiz.linkd, elem);
            else
                return true;
    }
    return false;
}
```

Árvores Binárias Ordenadas

```
public static void main(String args [ ]) {
    ArvBinOrd a1 = new ArvBinOrd();
    a1.incArv('L');
    a1.incArv('M');
    a1.incArv('F');
    a1.preOrdem();
    a1.emOrdem();
    a1.posOrdem();
    char x = 'M';
    System.out.println("Achou? "+BuscaArv(x));
}
}
```

Árvores Binárias Ordenadas

```
void excluiArv (char elem) {
    raiz = excluiAux(raiz, elem); }
Node excluiAux(Node raiz, char elem) {
    if (raiz != null) {
        if (raiz.data > elem) raiz.linke = excluiAux (raiz.linke, elem);
        else if (raiz.data < elem) raiz.linkd = excluiAux (raiz.linkd, elem);
        else { if (raiz.linke == null && raiz.linkd == null) return null;
            else if (raiz.linke != null && raiz.linkd != null) { raiz = raiz.linke; return raiz; }
            else if (raiz.linke != null && raiz.linkd == null) { raiz = raiz.linkd; return raiz; }
            else { Node aux = raiz.linke;
                while (aux.linkd != null)
                    aux = aux.linkd;
                raiz.data = aux.data;
                raiz.linke = excluiAux(raiz.linke, elem);
                return raiz;
            }
        }
    }
    return raiz;
}
```

Árvores Binárias Ordenadas

```
class ArvBinOrdFone {
    class Node {
        String nome;
        String fone;
        Node linke, linkd;
    }
    Node raiz;
    ArvBinOrdFone () {
        raiz = null;
    } ...
}
```

Árvores Binárias Ordenadas

```
void IncListaFone (String nm, String fn){
    raiz = IncAux(raiz, nm, fn);
}
Node IncAux (Node raiz, String nm, String fn){
    if(raiz == null){
        Node aux = new Node();
        aux.nome = nm; aux.fone = fn;
        aux.linke = null; aux.linkd = null;
        return aux;
    }
    else{ ...
}
```

Árvores Binárias Ordenadas

```
...
if(raiz.nome.compareTo(nm)>0){
    raiz.linke = IncAux(raiz.linke, nm, fn);
    return raiz;
}
else
    if(raiz.nome.compareTo(nm)<0){
        raiz.linkd = IncAux(raiz.linkd, nm, fn);
        return raiz;
    }
...

```

Árvores Binárias Ordenadas

```
...
else{
    System.out.println("Nome já existe!");
    return raiz;
}

}
}

```

Árvores Binárias Ordenadas

```
void preOrdem(){
    System.out.print("PreOrdem: []");
    preOrdemAux(raiz);
    System.out.println("]");
}
void preOrdemAux(Node raiz){
    if(raiz !=null){
        System.out.print(" "+raiz.nome);
        preOrdemAux(raiz.linke);
        preOrdemAux(raiz.linkd);
    }
}

```

Árvores Binárias Ordenadas

```
void emOrdem(){
    System.out.print("EmOrdem: []");
    emOrdemAux(raiz);
    System.out.println("]");
}
void emOrdemAux(Node raiz){
    if(raiz !=null){
        emOrdemAux(raiz.linke);
        System.out.print(" "+raiz.nome);
        emOrdemAux(raiz.linkd);
    }
}

```

Árvores Binárias Ordenadas

```
void posOrdem(){
    System.out.print("PosOrdem: []");
    posOrdemAux(raiz);
    System.out.println("]");
}
void posOrdemAux(Node raiz){
    if(raiz !=null){
        posOrdemAux(raiz.linke);
        posOrdemAux(raiz.linkd);
        System.out.print(" "+raiz.nome);
    }
}

```

Árvores Binárias Ordenadas

```
String BuscaAux (Node raiz, String nm) {
    if(raiz == null) {
        return "não encontrou";
    }
    else {
        if(raiz.nome.compareTo(nm)>0) {
            return BuscaAux(raiz.linke, nm);
        }
        else
            if(raiz.nome.compareTo(nm)<0) {
                return BuscaAux(raiz.linkd, nm);
            }
        else {
            return raiz.fone;
        }
    }
}

```

Árvores Binárias Ordenadas

```
public static void main(String args [ ]) {  
    ArvBinOrdFone a1 = new ArvBinOrdFone( );  
    a1.IncListaFone("Lidia","011 99999999");  
    a1.IncListaFone("Marcio","011 88888888");  
    a1.IncListaFone("Marcelo","011 77777777");  
    a1.preOrdem( );  
    a1.emOrdem( );  
    a1.posOrdem( );  
    String nm = "Marcelo";  
    String fn = l1.BuscaFone(nm);  
}  
}
```