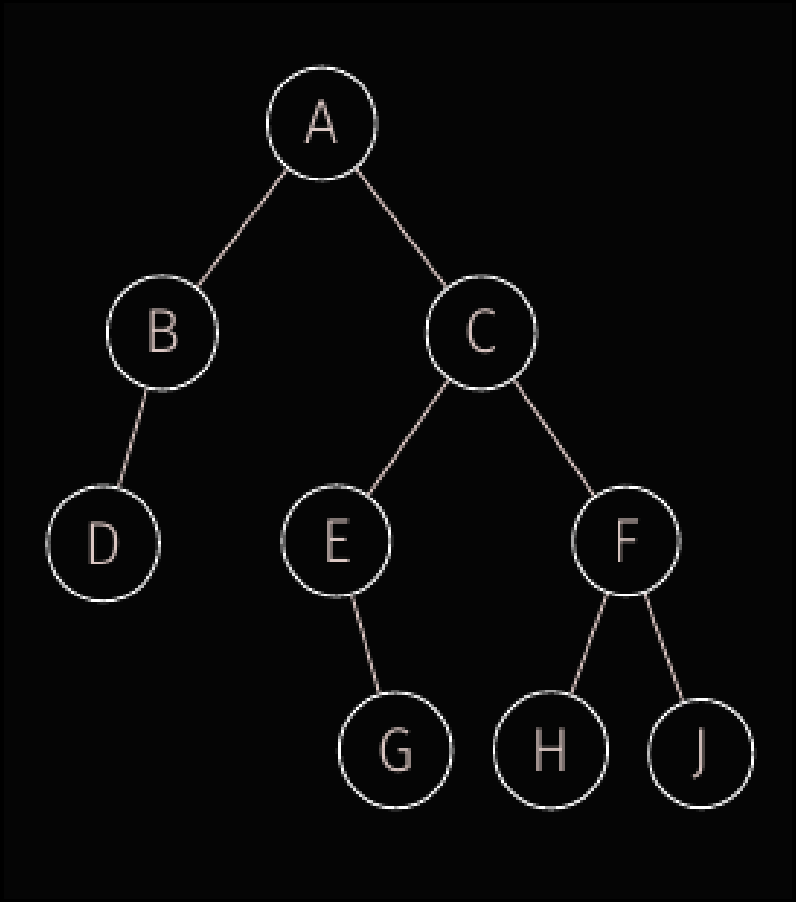
AULA 15
ESTRUTURA DE DADOS

Árvores

Árvore

Uma árvore é uma estrutura não-linear que permite modelar ramificações, escolhas.

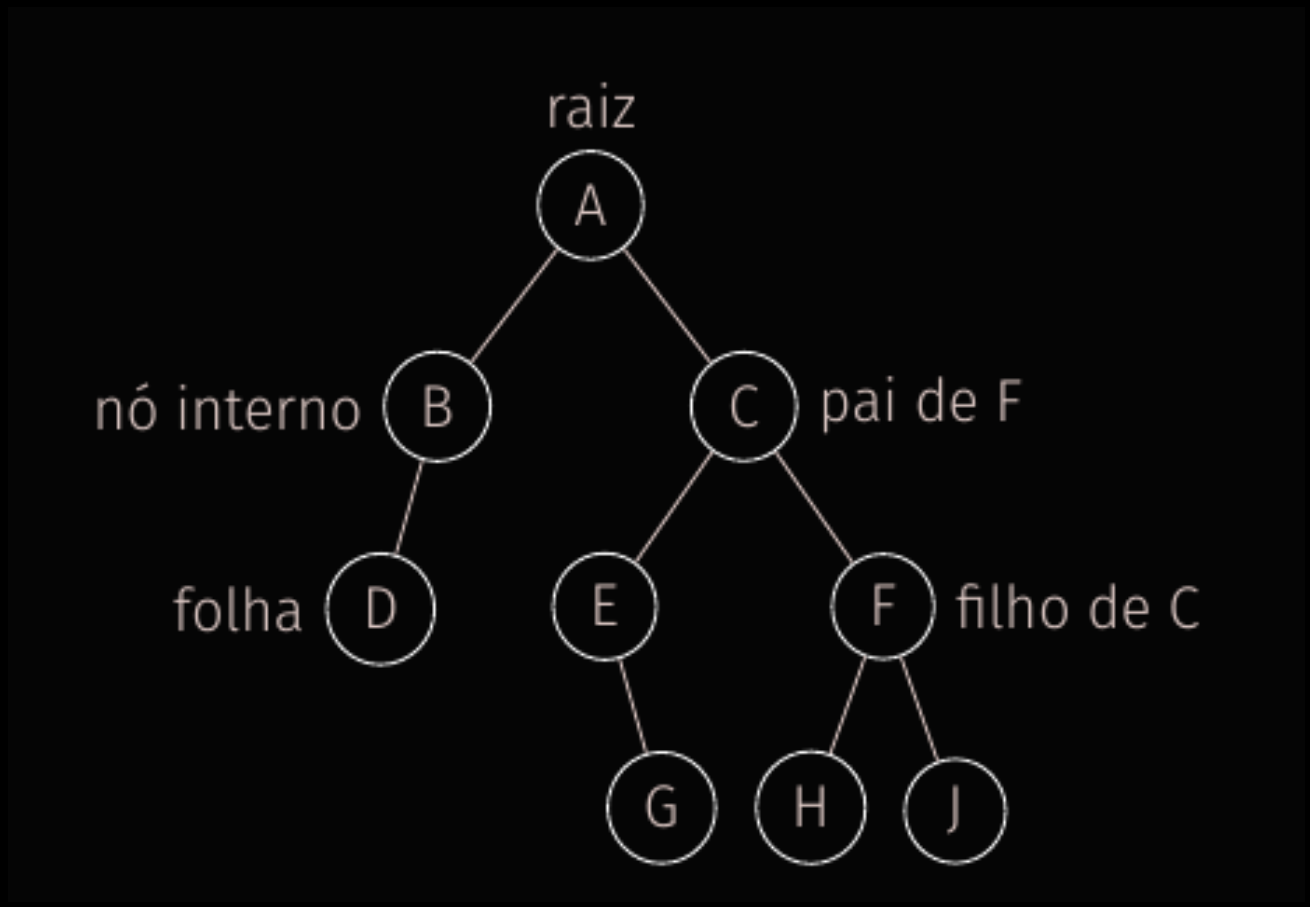
Árvore



Árvore

raiz: o primeiro elemento da árvore;
nó: representa um registro contido na árvore.
filho: um dos elementos que sucede um nó.
pai: elemento que precede um nó.
folha: elementos que não possui filhos.
interno: nó que possui filhos.

Árvore

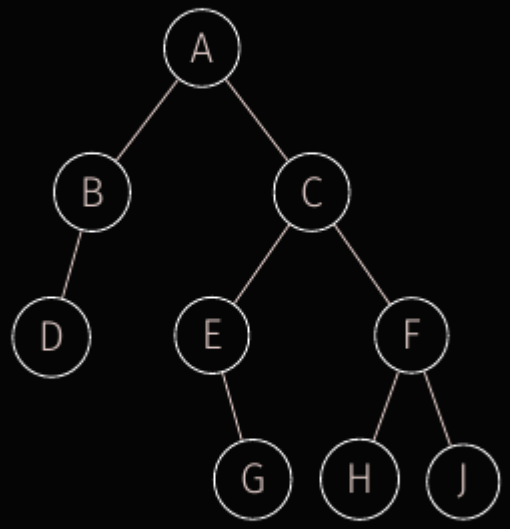


Árvore

O **Grau de um nó** é o número de sub-árvores abaixo desse nó (ou quantos filhos ele possui). Note que o grau de um nó folha é 0.

Árvore

O Grau do nó A é 2.



Árvore

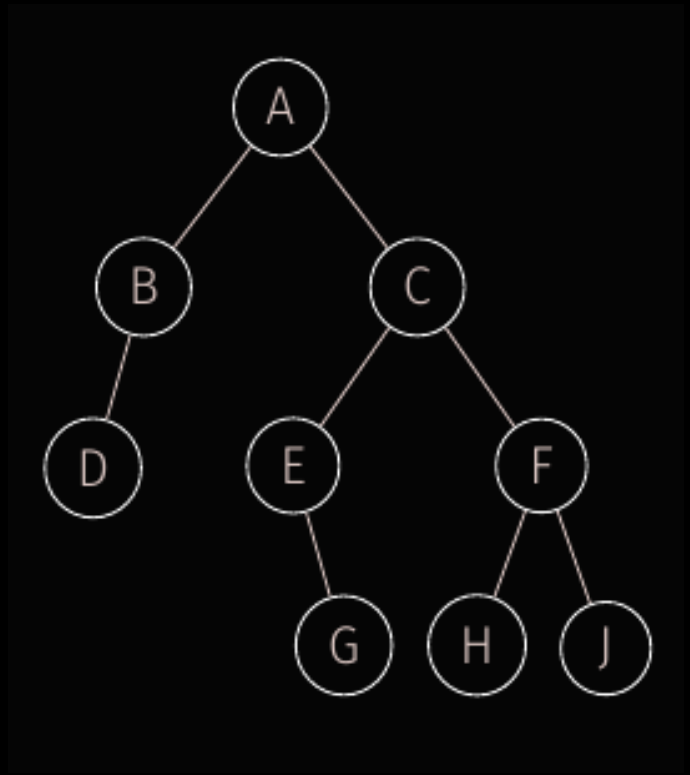
A altura de um nó é o caminho mais longo do nó até uma folha. A altura da árvore é a altura da raiz.

A profundidade de um nó é o número de arestas do nó até a raiz.

O nível de um nó é definido como 1+ o nível de seu pai, sendo o nível da raíz igual a 0.

Árvore

A altura do nó C é 2, a altura da árvore é 3, a profundidade do nó C é 1 e o nível dele é 1.



Árvore Binária

Uma árvore comum na computação é a árvore binária cujos nós possuem de 0 a 2 filhos:

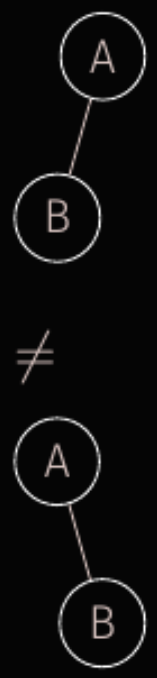
Árvore Binária

Modelagem:

```
typedef struct bintree {
        TREE_TYPE data;
        struct bintree * left;
        struct bintree * right;
} bintree;
```

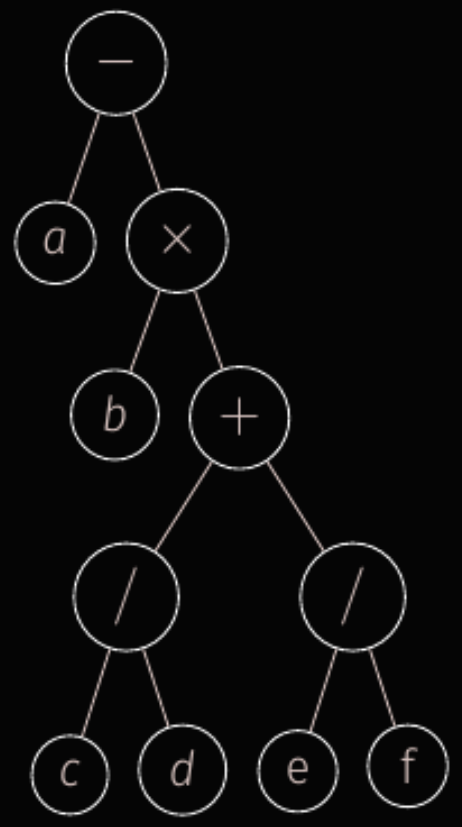
Árvore Binária

Note que as seguintes árvores são diferentes!



Árvore Binária - Exemplo

$$a - b(c/d + e/f)$$



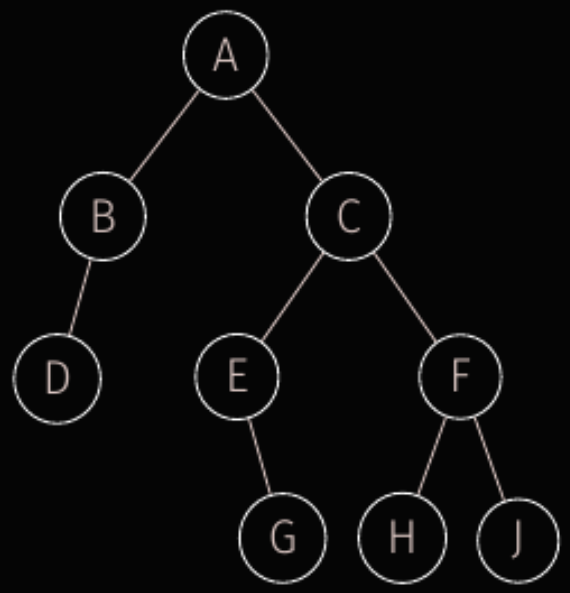
Árvore Binária - Exemplo

Para percorrer os elementos de uma árvore, a cada nó, temos que decidir qual ramo iremos explorar primeiro:

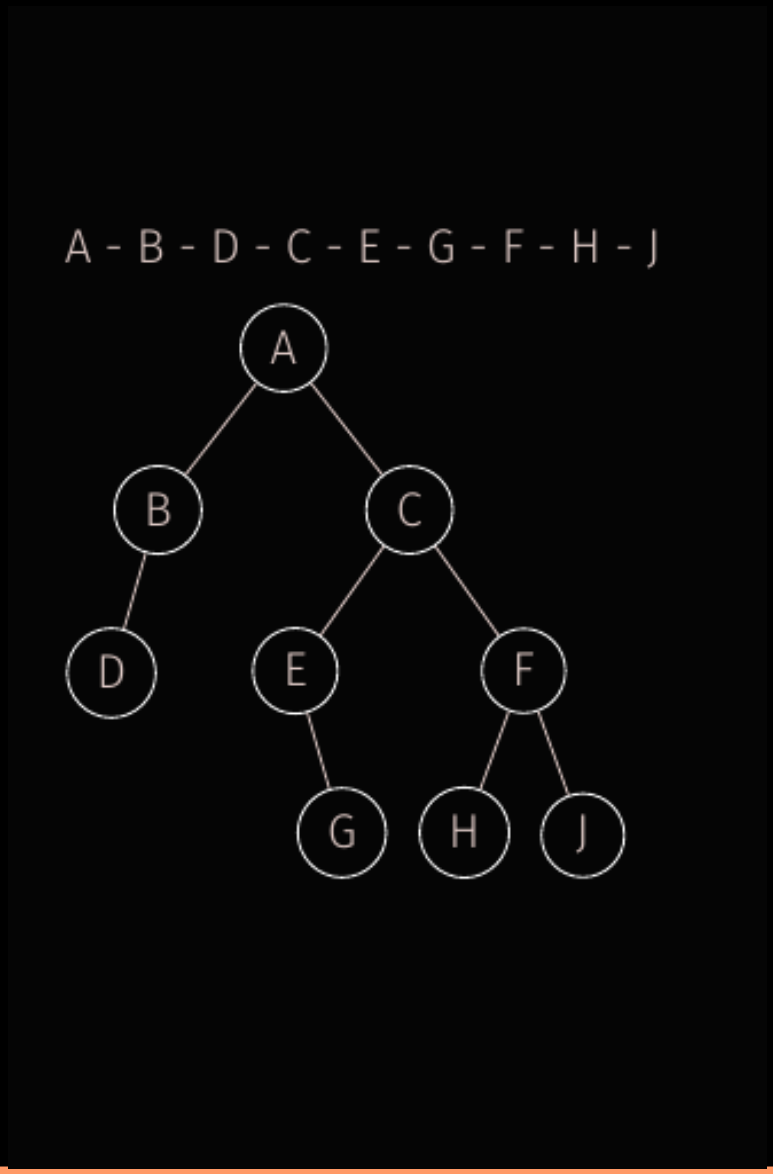
- Pré-ordem: raíz - esquerda - direita.
- Em-ordem: esquerda - raíz - direita.
- Pós-ordem: esquerda - direita - raiz.

Árvore Binária

Qual a ordem dos nós ao fazer o percurso pré-ordem?

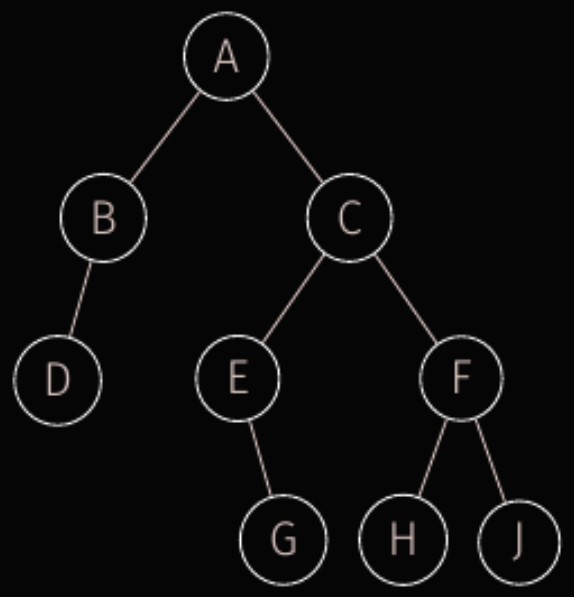


Árvore Binária

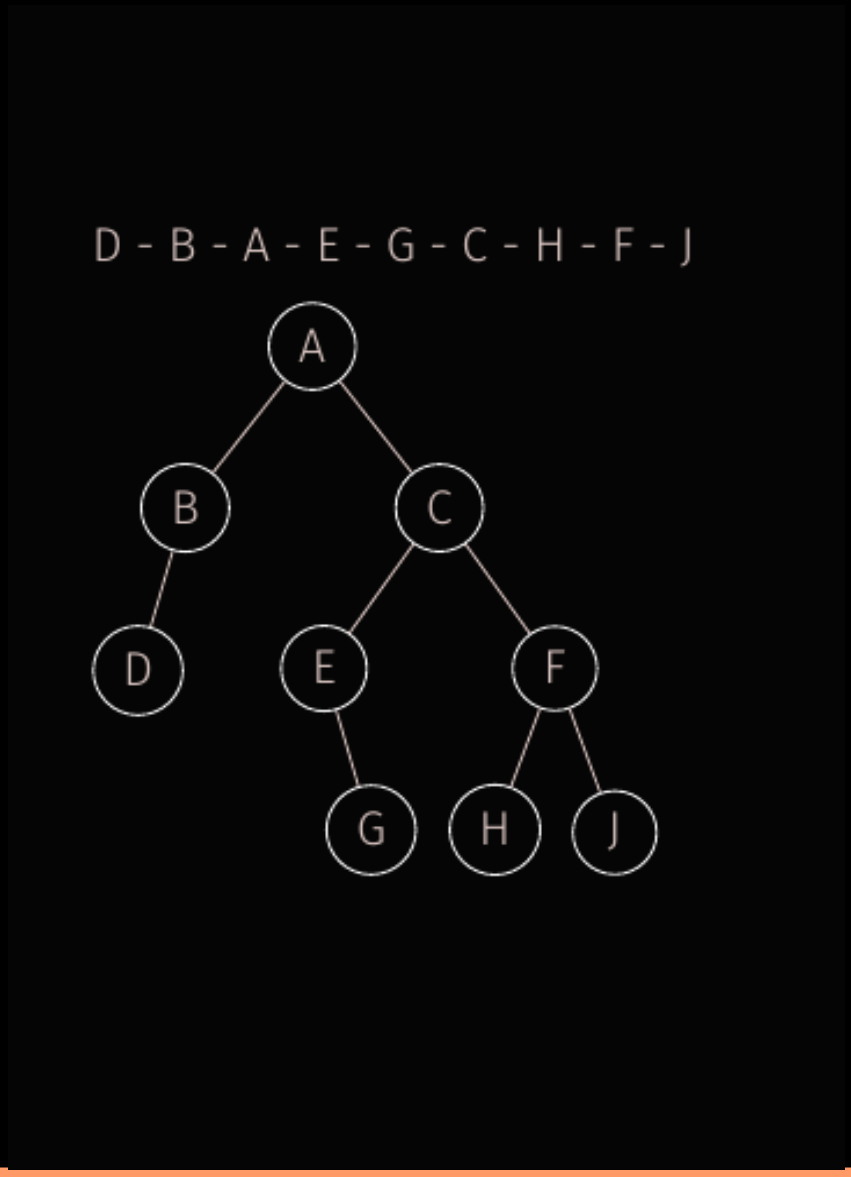


Árvore Binária

Qual a ordem dos nós ao fazer o percurso em-ordem?

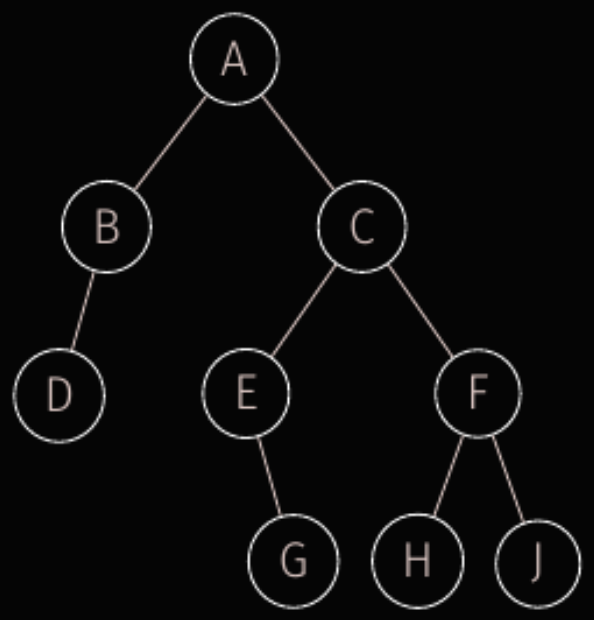


Árvore Binária

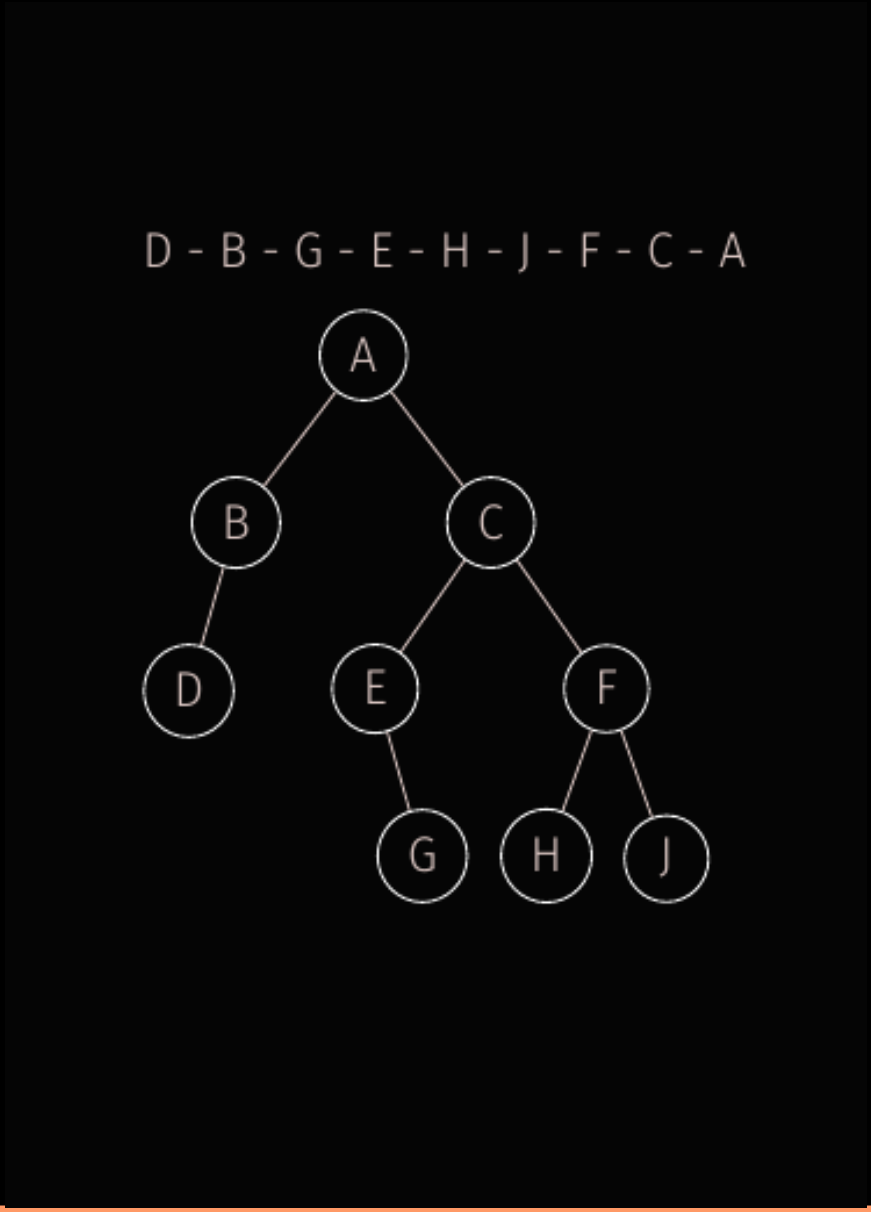


Árvore Binária

Qual a ordem dos nós ao fazer o percurso pós-ordem?



Árvore Binária



Árvore Binária

Para criar uma árvore precisamos estabelecer o critério de onde inserir cada novo nó.

Árvore Binária

Vamos adotar o critério de uma árvore ordenada.

Para um nó n,todos os nós a esquerda possuem valor menor do que ele e todos osnós a direita valores maiores ou iguais.

Seguindo esse critério, um novo nó será inserido ao encontrarmos um nó folha.

Árvore Binária

```
bintree * create_node (int x) {

}
```

Árvore Binária

```
bintree * create_node (int x) {
        bintree * node = malloc(sizeof(bintree));
        node->left = node->right = NULL;
        node->x = x;
        return node;
}
```

Árvore Binária

```
bintree * insert_sorted (bintree * t, bintree * node)
 {

 }
```

Árvore Binária

```
bintree * insert_sorted (bintree * t, bintree * node)
 {
        if (t==NULL) return node;
        if (node->x < t->x)
        t->left = insert_sorted(t->left, node);
        if (node->x > t->x)
        t->right = insert_sorted(t->right, node);
        return t;
 }
```

Árvore Binária

```
bintree * insert_sorted (bintree * t, bintree * node)
 {
        if (t==NULL) return node;
        if (node->x < t->x)
        t->left = insert_sorted(t->left, node);
        if (node->x > t->x)
        t->right = insert_sorted(t->right, node);
        return t;
 }
```

Árvore Binária - Inserção

```
insert_sorted(root, create_node(5));
```

5

Árvore Binária - Inserção

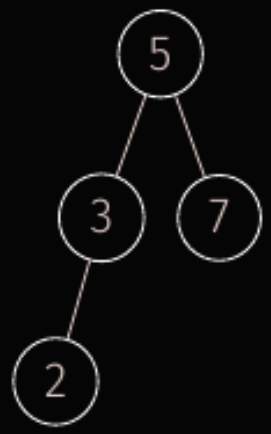
```
insert_sorted(root, create_node(3));
```

Árvore Binária - Inserção

```
insert_sorted(root, create_node(7));
```

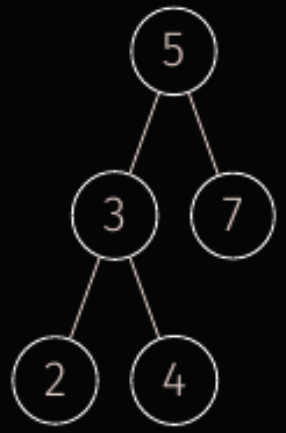
Árvore Binária - Inserção

```
insert_sorted(root, create_node(2));
```



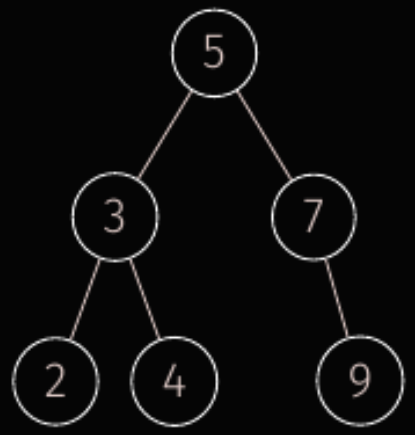
Árvore Binária - Inserção

```
insert_sorted(root, create_node(4));
```



Árvore Binária - Inserção

```
insert_sorted(root, create_node(9));
```



Árvore Binária

Para imprimir a sequência de nós no percurso pré-ordem, fazemos:

```
void pre_order(bintree * t) {

}
```

Árvore Binária

Para imprimir a sequência de nós no percurso pré-ordem, fazemos:

```
void pre_order(bintree * t) {
        if(t==NULL) return;
        printf("%d ", t->x);
        pre_order(t->left);
        pre_order(t->right);
}
```

Árvore Binária

Exercício: Implementar as funções pos_ordem e em_ordem;

Árvore Binária de Busca

Uma árvore binária de busca é uma árvore ordenada cujo desempenho para encontrar um elemento é equivalente ao de uma busca binária.

Árvore Binária de Busca

Sua característica principal é a de que dado um nó n, todos os nós a esquerda possuem um valor menor ou igual a ele e todos os nós a direita possuem um valor maior ou igual a ele.

Definição: Uma árvore binária é completa se todos os níveis, exceto o último, está completamente preenchida.

Árvore Binária de Busca

Essa árvore é ordenada conforme nosso algoritmo de inserção.

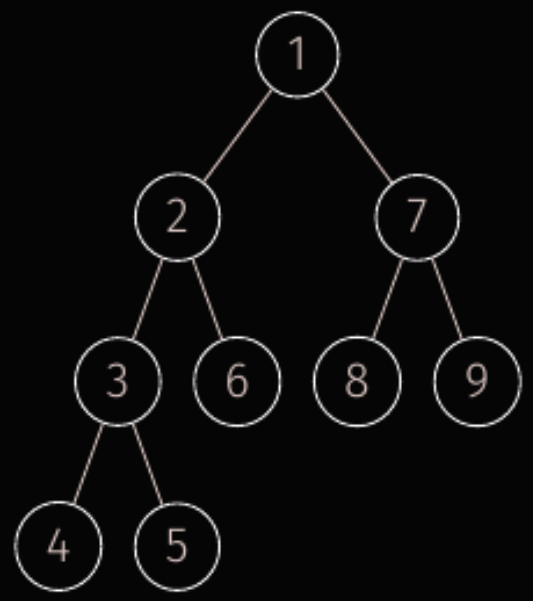
Para um desempenho ótimo, ela deve ser completa.

Árvore Binária de Busca

Uma árvore binária completa pode ser representada como uma lista sequencial de tal forma que o pai do índice i está na posição $\lfloor (i - 1)/2 \rfloor$ e os filhos de i estão na posição $2i + 1$ e $2i + 2$.

Árvore Binária de Busca

0	1	2	3	4	5	6	7	8
1	2	7	3	6	8	9	4	5



Árvore Binária de Busca

```
bintree * search (bintree * t, int k) {

}
```

Árvore Binária de Busca

```
bintree * search (bintree * t, int k) {
        bintree * p = t;

        while(p!=NULL){
                if (p->x == k) return p;
                if (k < p->x) { p = p->left;}
                else { p = p->right;}
        }
        return NULL;
}
```

AULA 15
ESTRUTURA DE DADOS

Árvores