

# Aula 13 - Árvores Binárias

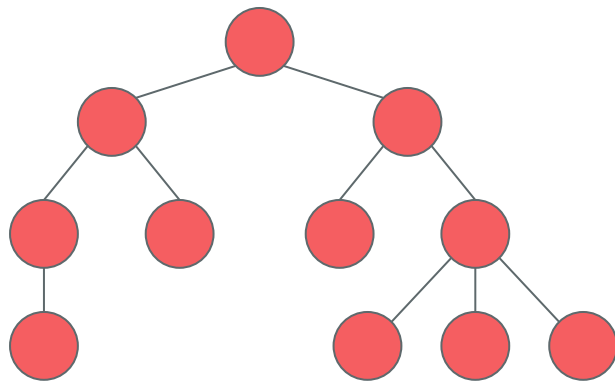
## Parte 2

Estruturas de Dados 2018/1

Prof. Diego Furtado Silva

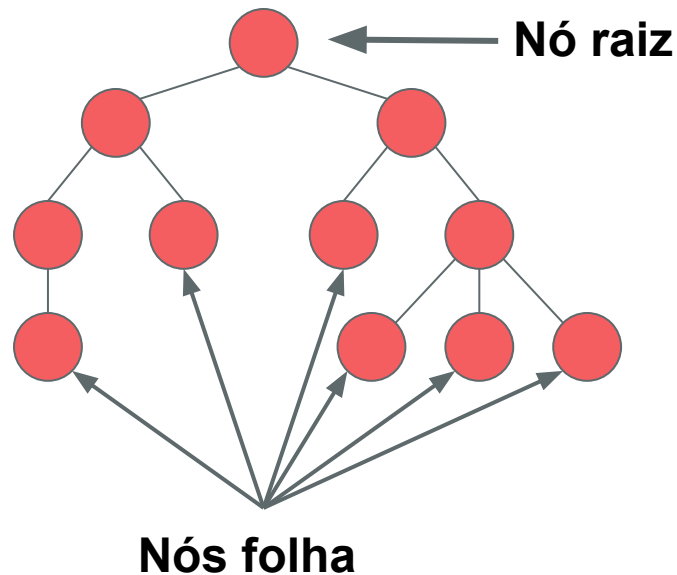
# Árvores - relembrando

Árvores utilizam estruturas hierárquicas



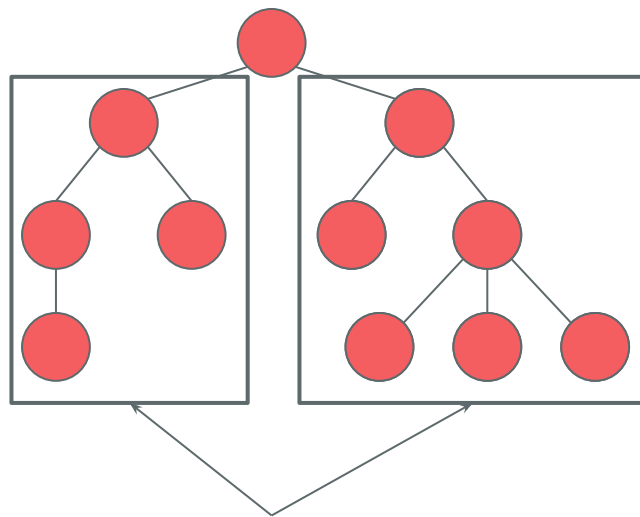
# Árvores - relembrando

## Árvores utilizam estruturas hierárquicas



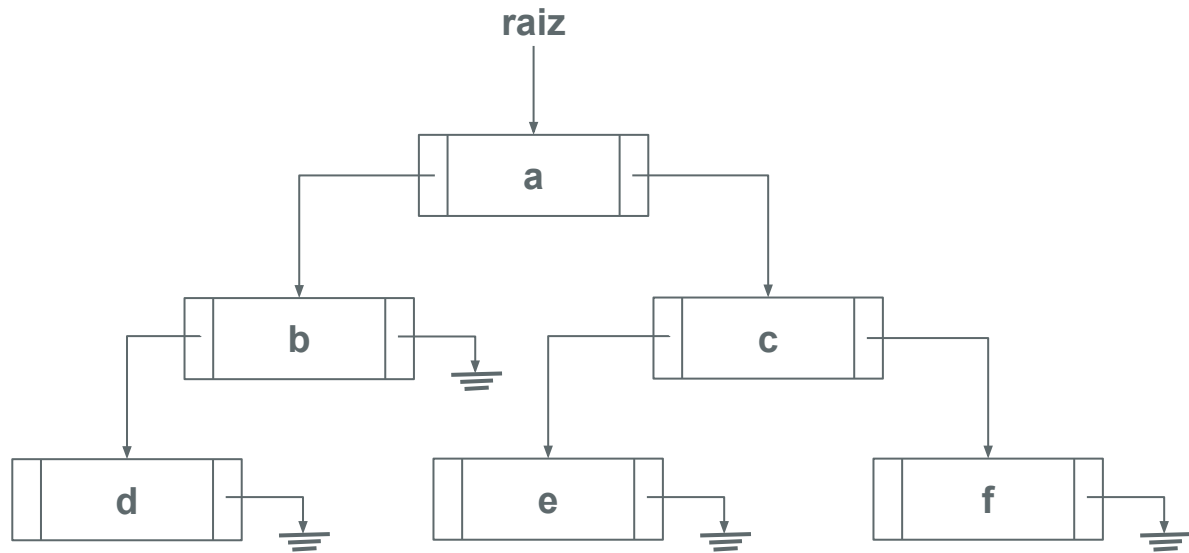
# Árvores - relembrando

Árvores utilizam estruturas hierárquicas



**Sub-árvores  
do nó raiz**

# Árvores Binárias - relembrando



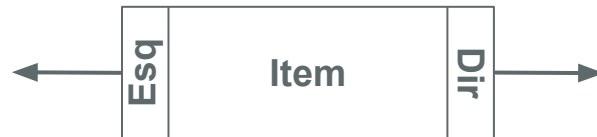
# Árvores Binárias - relembrando

```
typedef int TipoChave;

typedef struct {
    TipoChave Chave;
    /* Outros campos */
} TipoItem;

typedef struct TipoNo {
    TipoItem Item;
    struct TipoNo *esq, *dir;
} TipoNo;

typedef struct TipoNo *TipoApontador;
typedef TipoApontador TipoArvore;
```



# Árvores Binárias - Agora sim coisa nova!

## Operações básicas:

- Cria árvore
- Cria raiz
- Insere esquerda
- Insere direita

Saudades de codar?

# Árvores Binárias – Percursos

Operações para percorrer uma AB visitando cada nó uma única vez

- Visitar pode significar qualquer tipo de operação feita no nó
  - Imprimir, modificar seu valor, etc
- Um percurso gera uma sequência linear de nós visitados
  - Portanto, agora temos o conceito de sucessor e predecessor de um nó

```
void visita(TipoArvore A) {  
    printf("%c ", A->Item.Chave);  
}
```



# Árvores Binárias – Percursos

Há, claro, mais de uma maneira de percorrer uma árvore (binária ou não)

- Diferentes percursos podem ser realizados, dependendo da aplicação
- Imprimir uma árvore, buscar por um item, etc

3 percursos comuns para ABs podem ser feitos com o mesmo algoritmo base

- Pré-ordem (*pre-order*)
- Em-ordem (*in-order*)
- Pós-ordem (*post-order*)

# Árvores Binárias – Percursos

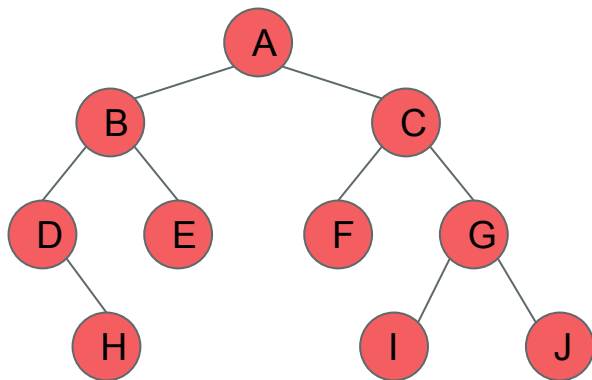
Pré-ordem (*pre-order*), em-ordem (*in-order*), pós-ordem (*post-order*):

- Diferença básica está na ordem em que os nós são visitados
- Percorre-se a AB recursivamente
  - Pré-ordem: visita o nó **antes** de acessar qualquer um dos seus filhos
  - Em-ordem: visita o nó **entre** o acesso a cada um de seus dois filhos
  - Pós-ordem: visita o nó somente **depois** de acessar seus filhos

# Árvores Binárias – Percursos

Algoritmo base é a busca em profundidade. Codemos!

# Árvores Binárias - Percursos

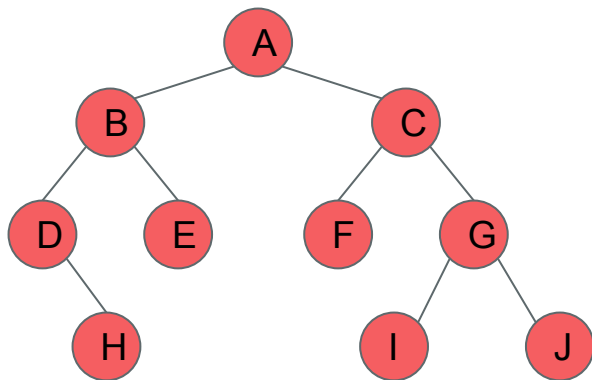


Pré-ordem

```
void visita(TipoArvore A) {  
    printf("%c ", A->Item.Chave);  
}  
  
void preOrdem(TipoArvore A) {  
    if (A != NULL) {  
        visita(A);  
        preOrdem(A->esq);  
        preOrdem(A->dir);  
    }  
}
```

Percurso: A, B, D, H, E, C, F, G, I, J

# Árvores Binárias - Percursos

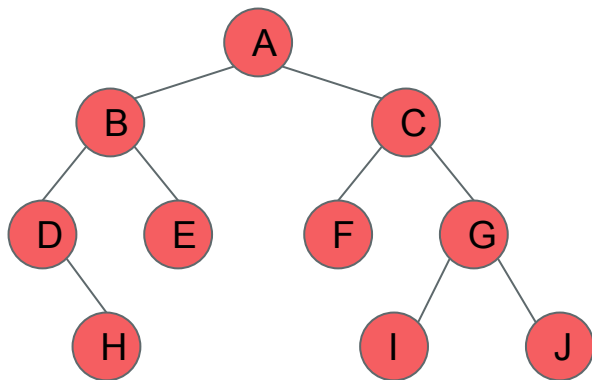


Em-ordem

```
void emOrdem(TipoArvore A) {  
    if (A != NULL) {  
        emOrdem(A->esq);  
        visita(A);  
        emOrdem(A->dir);  
    }  
}
```

Percurso: D, H, B, E, A, F, C, I, G, J

# Árvores Binárias - Percursos



Pós-ordem

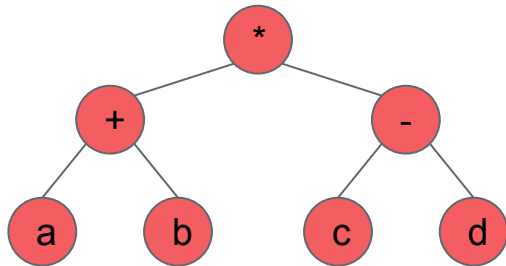
```
void posOrdem(TipoArvore A) {  
    if (A != NULL) {  
        posOrdem(A->esq);  
        posOrdem(A->dir);  
        visita(A);  
    }  
}
```

Percurso: H, D, E, B, F, I, J, G, C, A

# Árvores Binárias - Exercícios

Como representar uma expressão em uma árvore binária? Crie um algoritmo para isso. Dica: na notação pós-fixa é mais fácil.

Ex:  $(a + b) * (c - d) \rightarrow a b + c d - *$



# Árvores Binárias – Exercícios

Qual é a relação entre os percursos apresentados e as notações matemáticas infixa, pós-fixa e pré-fixa? Relembrando:

Notação infixa: o **operador** é escrito **entre dois operandos**

Notação pós-fixa: primeiro escrevemos os **operandos e depois o operador**

Notação pré-fixa: o **operador** é escrito **antes dos operandos**

Infixa	Pós-fixa	Pré-fixa
$a + b$	$a b +$	$+ a b$
$a + b * c$	$a b c * +$	$+ a * b c$
$(a + b) * (c - d)$	$a b + c d - *$	$* + a b - c d$



# Árvores Binárias – Exercícios

Escreva procedimentos recursivos que:

- 1) apaga (todos os nós de) uma AB;
- 2) calcula a altura de uma AB;
- 3) calcula o nível de um dado nó de uma AB;
- 4) verifica a existência de um nó com uma chave determinada pelo usuário.

# Árvores Binárias – Exercícios

Escreva um procedimento que faça um percurso por nível/altura da árvore.

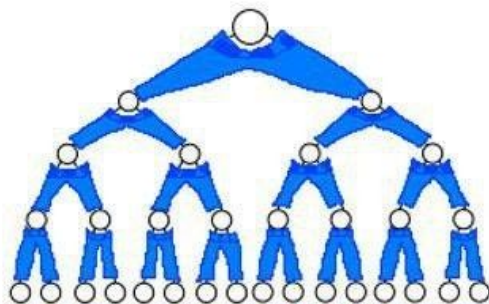
Dica: use uma fila (e lembre-se do *handshake distance*)

# Árvores Binárias - Exercícios

Responda à seguinte questão:

- Se uma árvore binária usasse calças, ela as vestiria

assim



ou assim?

