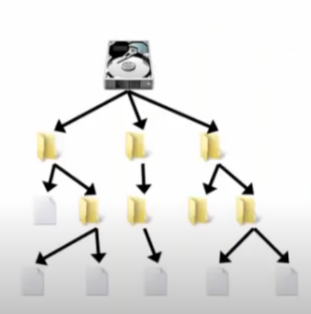
**Arvores**

* Qualquer par de vértice está conectado a apenas uma aresta
* Na arvore há apenas conexo (não há ciclo)

* **Arvore:** Conjunto não vazio de vértices
* **Vértices:** Cada uma das entidades representadas
* **Aresta:** Conexão entre dois vértices

Exemplo de arvore: Registro de diretórios do computador

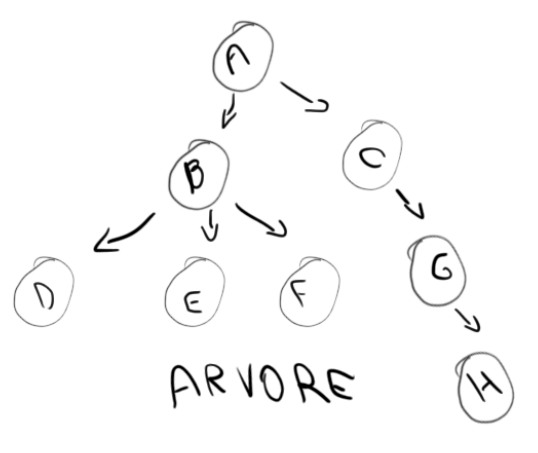
**Aplicações**

Arvores são usadas para representar estruturas **hierárquica e não lineares**

Ex: Relação de descendência: Avô, Pai, Filho...

Diagrama hierárquico: Dono, Gerente, Coordenador, Supervisor...

**Formas de representação**



**Tipos de arvores**

* Arvore de busca
* Arvore AVL
* Arvore Rubro-Negra
* Arvore B, B+, B-, B\*
* Arvore 2-3, etc.

**Propriedades das arvores**

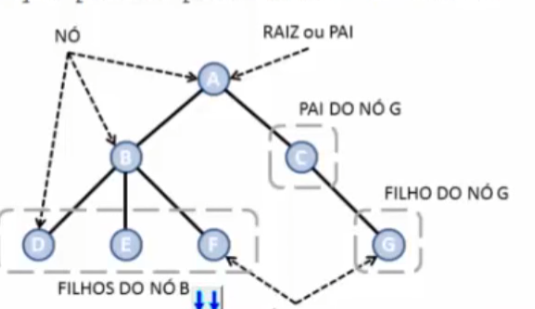
Pai: Antecessor imediato de um vértice

Filho: Sucessor imediato de um vértice

Raiz: Vértice que não possui Pai

Folha: Vértice que não possui Filho

Interno: Vértices internos que possui ao menos 1 filho



Sub-Arvores

Todo nó filho que possui filhos, se tornam raiz da sua própria sub-arvore

Altura da arvore: Comprimento que a arvore percorre do caminho mais longo da raiz a folha, neste caso = 3

**Implementação de uma Arvore**

* Criação da Arvore
* Inserção de um elemento
* Remoção de um elemento
* Acesso a um elemento
* Destruição da Arvore

O acesso pode ser feito de duas formas

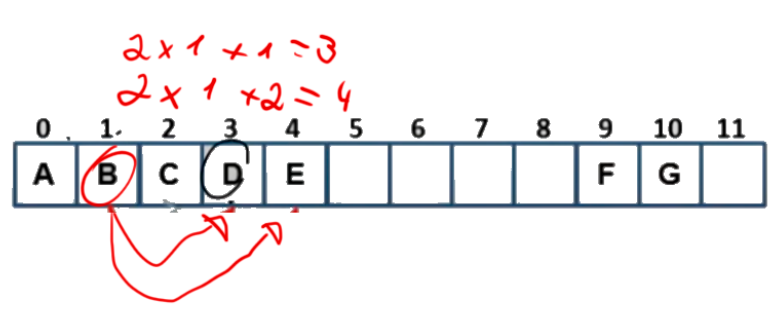
* Estática (Heap)
* Dinâmica (Lista Encadeada)

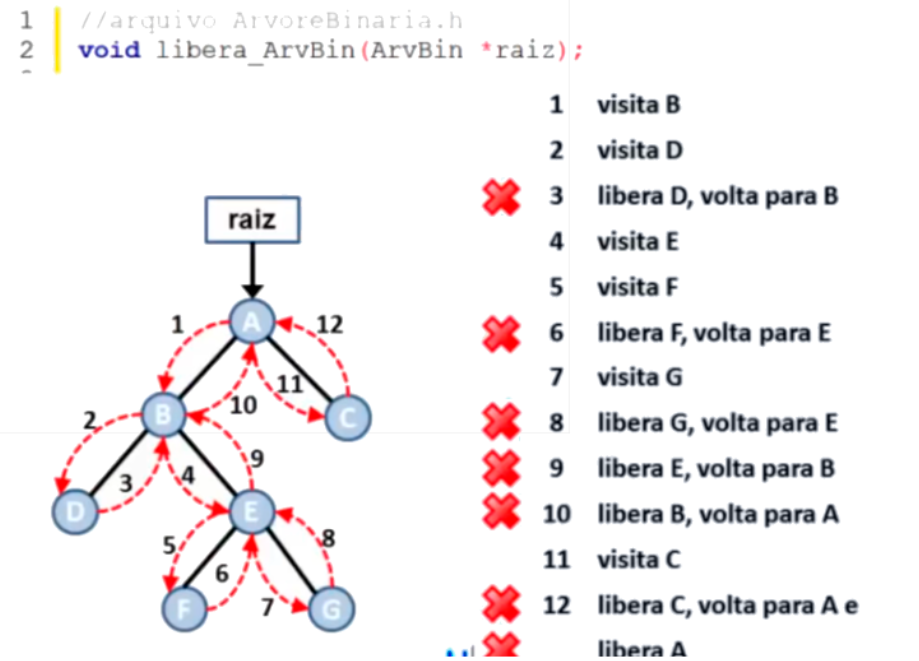
Alocação estática

Usa um Array (Vetor)

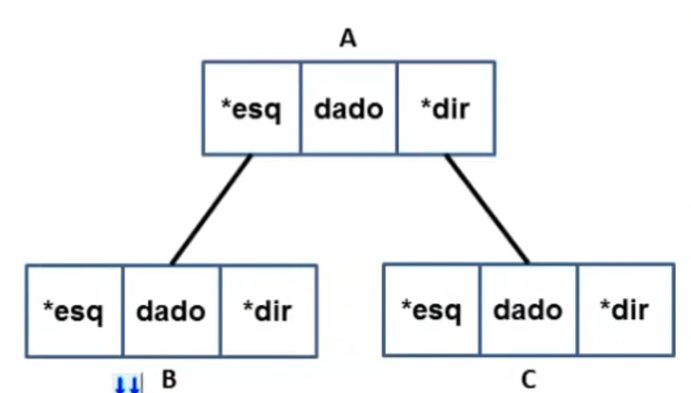
Usa duas posições para retornar à posição de Filhos a esquerda e a direita em relação ao Pai

Filho\_Esq(Pai) = 2 \* Pai + 1; Filho\_Dir(Pai) = 2 \* Pai + 2;



Alocação dinâmica

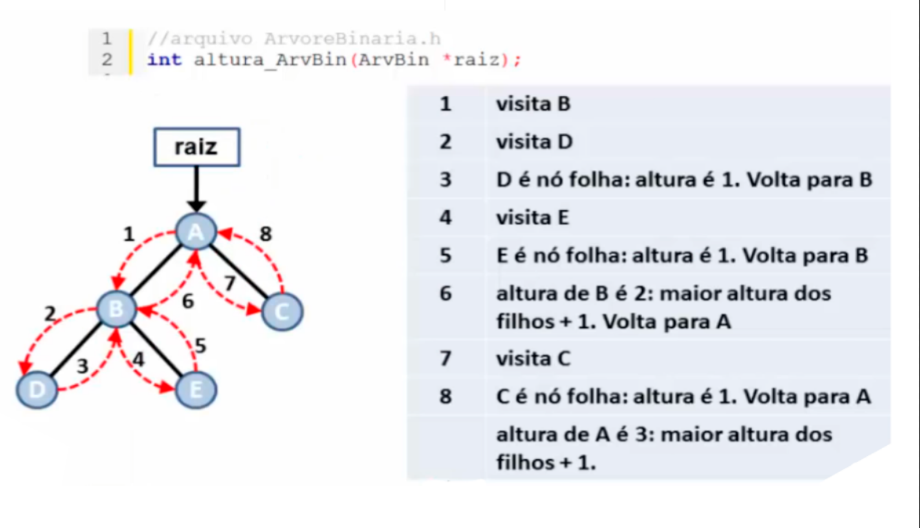
Cada nó da arvore é um Ponteiro alocado dinamicamente



Para guardados os dados usamos um ponteiro de ponteiros

**Implementação de uma arvore**

ArvoreBinaria.h: Define:

* Os protótipos das funções
* O tipo de dado armazenado na árvore
* O ponteiro da Arvore

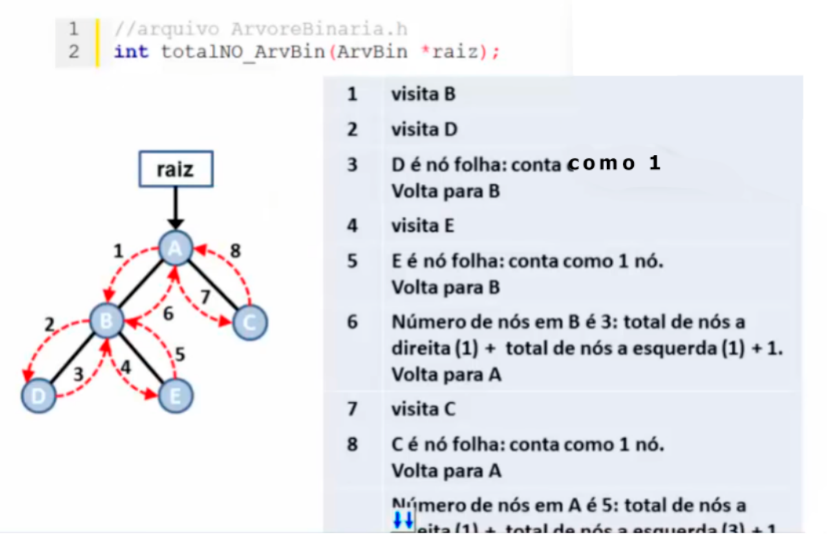
ArvoreBinaria.c: Define:

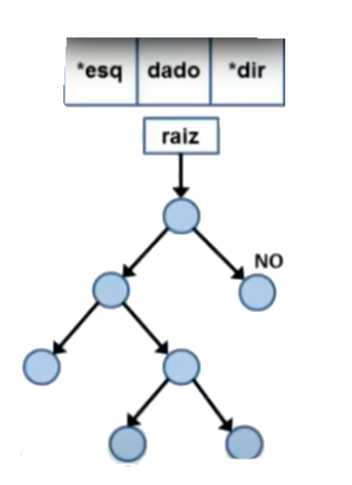
* O tipo de dado da Arvore
* Implementa suas funções

Criação de uma arvore binária

Cria a raiz da arvore e aponta para o primeiro elemento da arvore.

Destruição de uma arvore binária

Percorre todos os nos da arvore, liberando a memória individual de cada elemento.



Informações básicas sobre Arvore

**Está Vazia** – Auto Explicativo

**Altura da Arvore** – Explicação Codigo + foto

**Total de nos** – Explicação Codigo + foto

**Percorrendo uma arvore**

É a operação mais utilizada

Percorre todos os nós uma única vez

3 formas de percorrer

PRÉ ORDEM

Visita a raiz, o filho da esquerda e após, o filho da direita.

EM ORDEM

Visita o filho da esquerda, a raiz e após, o filho da direita.

PÓS ORDEM

Visita o filho da esquerda, o filho da direita e após, a raiz.