

Árvores Genéricas

Prof. Luiz Gustavo Almeida Martins

Árvores genéricas

Árvore sem restrição do número de filhos

Cada nó pode ter um **número arbitrário de filhos**

Ex: árvore de diretórios

Árvores genéricas

Árvore sem restrição do número de filhos

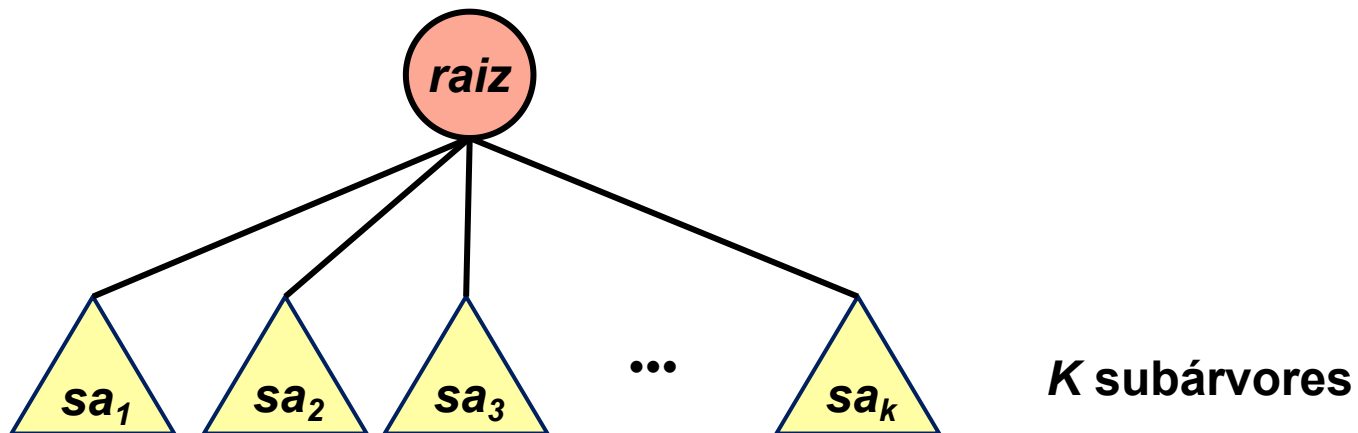
Cada nó pode ter um **número arbitrário de filhos**

Ex: árvore de diretórios

Definição: Se **A** é uma árvore genérica, então:

A é uma árvore vazia; ou

A contém um nó raiz e zero, uma ou mais subárvores



Árvores genéricas

Árvore sem restrição do número de filhos

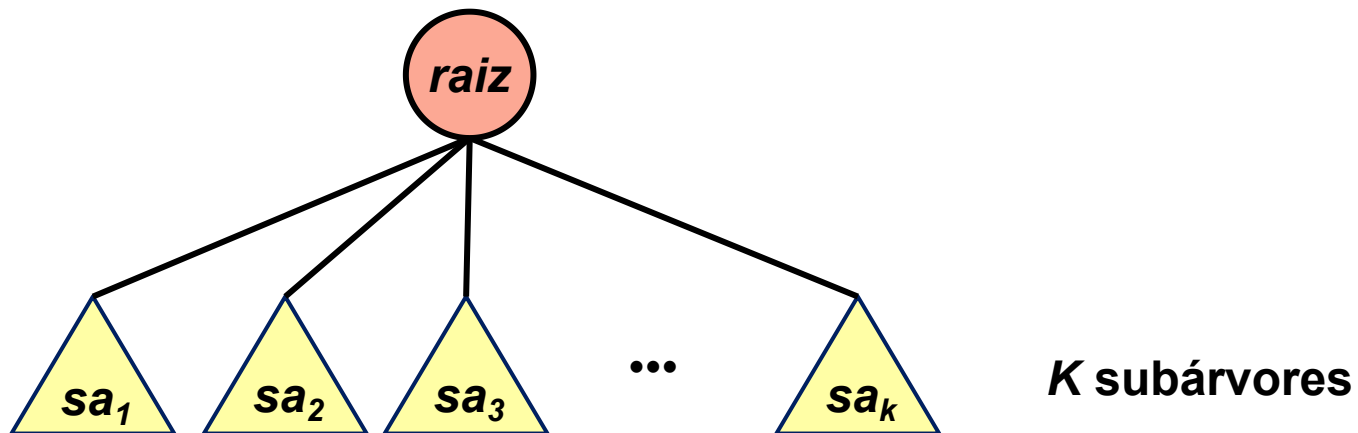
Cada nó pode ter um **número arbitrário de filhos**

Ex: árvore de diretórios

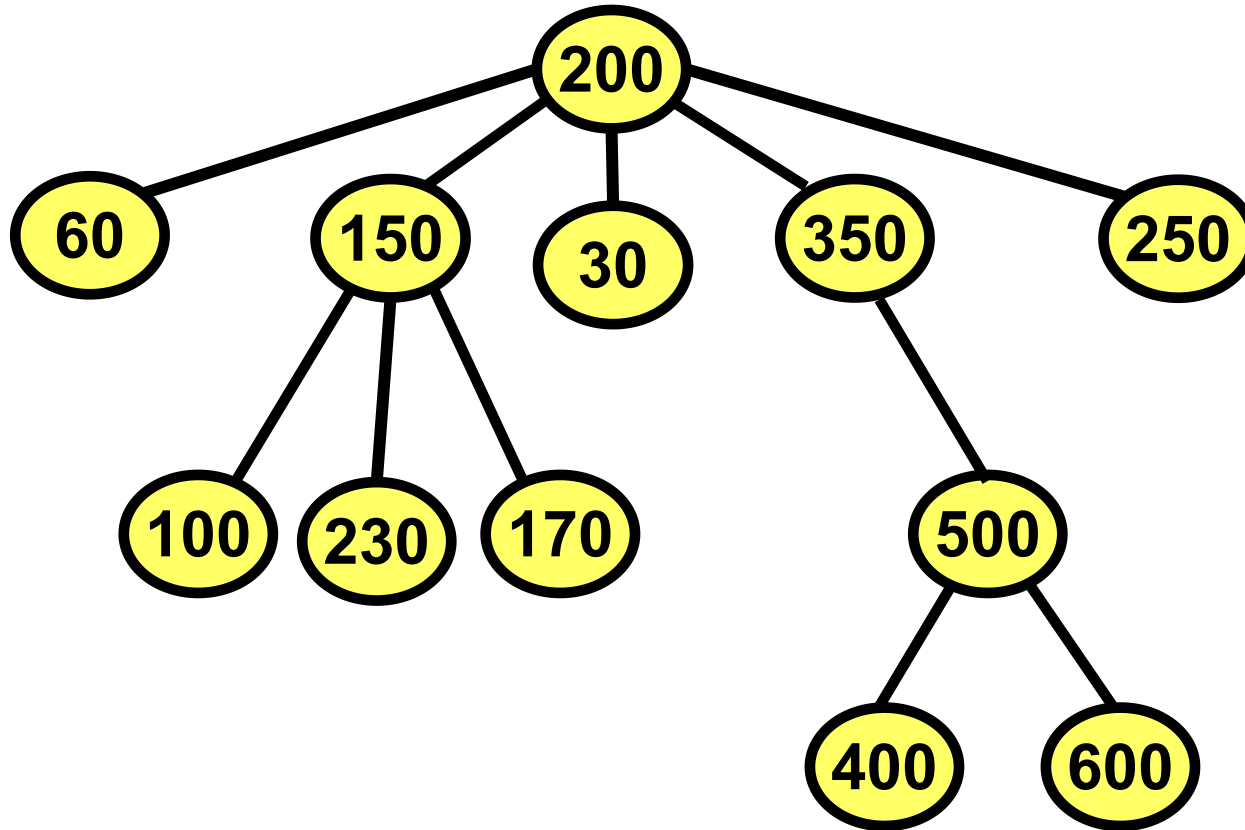
Definição: Se **A** é uma árvore genérica, então:

~~A é uma árvore vazia; ou~~ **Simplificação**

A contém um nó raiz e zero, uma ou mais subárvores



Exemplo de árvore genérica



Árvores genéricas: formas de representação

Contiguidade física:

Representação **não intuitiva**

Árvores genéricas: formas de representação

Contiguidade física:

Representação **não intuitiva**

Estratégia de representação que preserve hierarquia

Acesso precisa respeitar a hierarquia

Ex: usar **quantidade de filhos** como informação adicional

Árvores genéricas: formas de representação

Contiguidade física:

Representação **não intuitiva**

Estratégia de representação que preserve hierarquia

Acesso precisa respeitar a hierarquia

Ex: usar **quantidade de filhos** como informação adicional

Organização dos nós **por nível** ou **por profundidade**

Árvores genéricas: formas de representação

Contiguidade física:

Representação **não intuitiva**

Estratégia de representação que preserve hierarquia

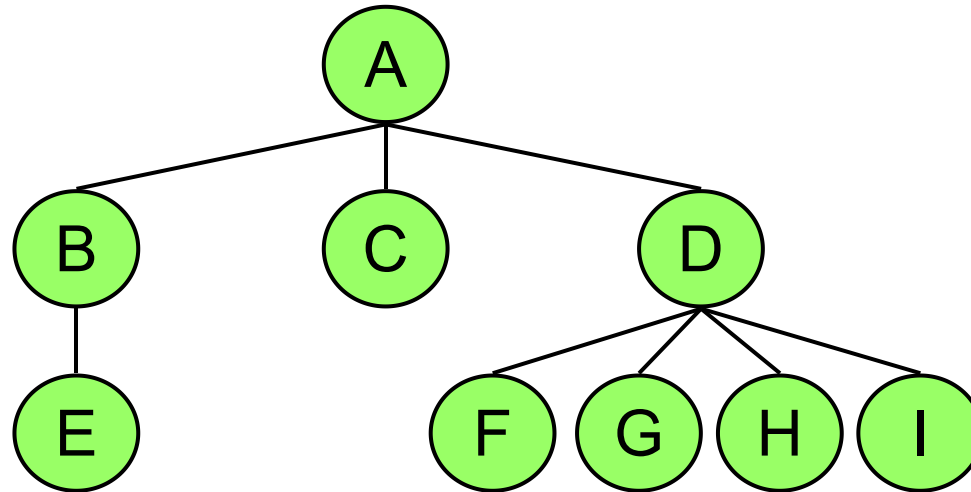
Acesso precisa respeitar a hierarquia

Ex: usar **quantidade de filhos** como informação adicional

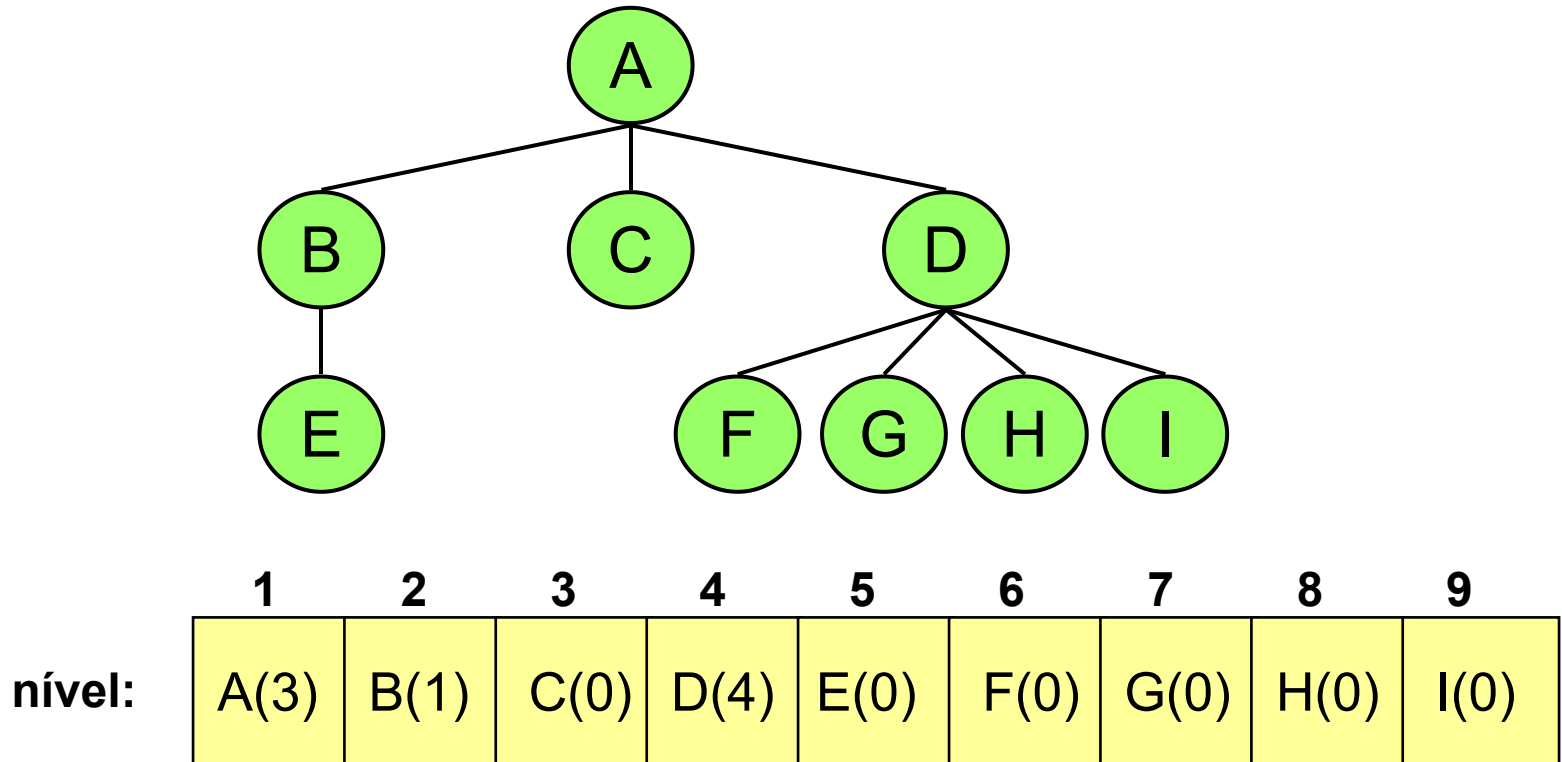
Organização dos nós **por nível** ou **por profundidade**

Inserção e remoção envolve **deslocamento de posição**

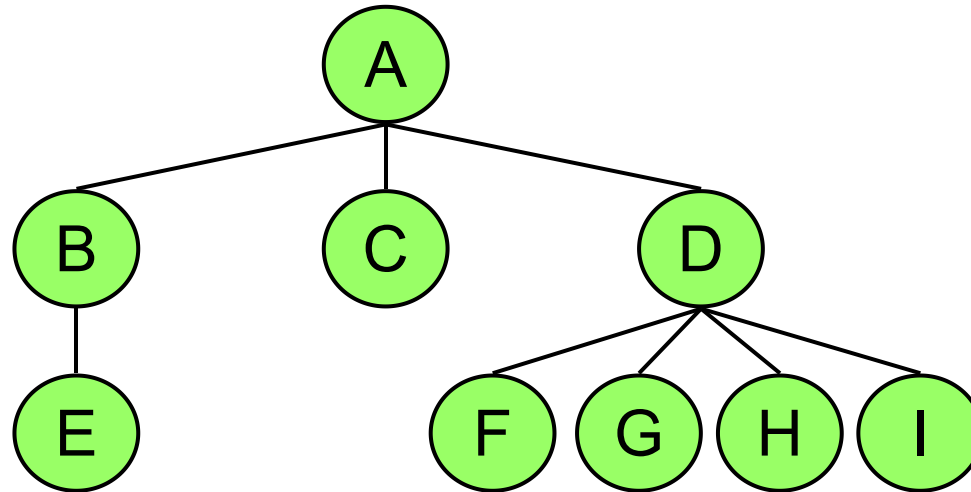
Exemplos de contiguidade física



Exemplos de contiguidade física



Exemplos de contiguidade física



	1	2	3	4	5	6	7	8	9
nível:	A(3)	B(1)	C(0)	D(4)	E(0)	F(0)	G(0)	H(0)	I(0)

	1	2	3	4	5	6	7	8	9
profundidade:	A(3)	B(1)	E(0)	C(0)	D(4)	F(0)	G(0)	H(0)	I(0)

Árvores genéricas: formas de representação

Contiguidade física:

Complexidade para seguir a hierarquia implícita na estrutura

Pode ser reduzida quando os nós têm **graus iguais** ou são **processados na mesma ordem** em que são armazenados

Movimentação de dados na inserção e remoção

Eficiente no espaço ocupado por cada nó

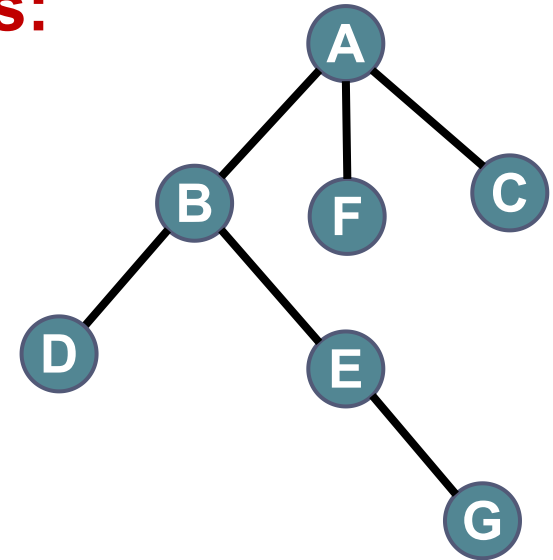
No geral, **não constitui uma boa solução**

Árvores genéricas: formas de representação

Representação baseada nos pais:

Lista com 2 campos: *info* e *pai*

	0	1	2	3	4	5	6
<i>info</i>	A	B	C	D	E	F	G
<i>pai</i>	-1	0	0	1	1	0	4

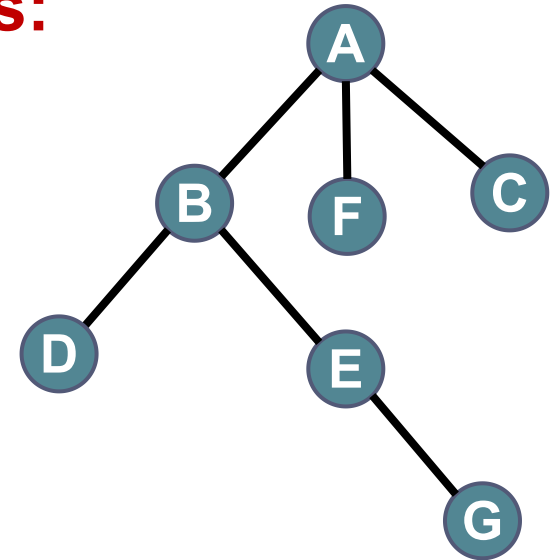


Árvores genéricas: formas de representação

Representação baseada nos pais:

Lista com 2 campos: *info* e *pai*

	0	1	2	3	4	5	6
<i>info</i>	A	B	C	D	E	F	G
<i>pai</i>	-1	0	0	1	1	0	4



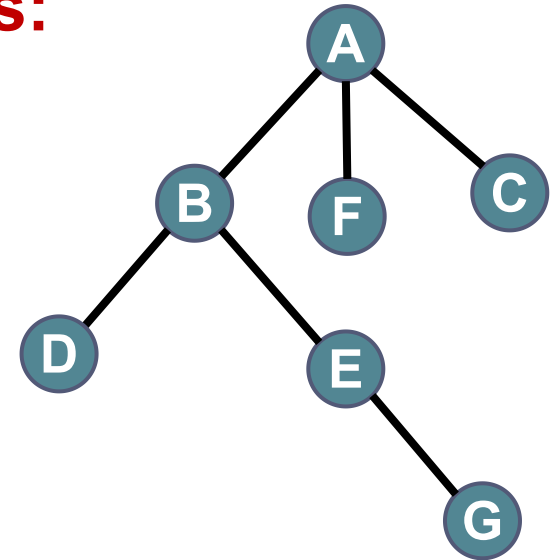
Interessante para operações que percorrem "de baixo para cima" (ex: encontrar o nó pai)

Árvores genéricas: formas de representação

Representação baseada nos pais:

Lista com 2 campos: *info* e *pai*

	0	1	2	3	4	5	6
<i>info</i>	A	B	C	D	E	F	G
<i>pai</i>	-1	0	0	1	1	0	4



Interessante para operações que percorrem "**de baixo para cima**" (**ex:** encontrar o nó pai)

Dependendo da forma de implementação:

Pode ocupar espaço desnecessário (**estática**)

Custo elevado em operações como filho e irmão (**dinâmica**)

Árvores genéricas: formas de representação

Árvore com grau fixo:

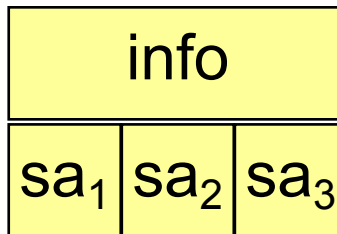
Representação dinâmica similar a das árvores binárias
Número máximo de filhos é fixado

Árvores genéricas: formas de representação

Árvore com grau fixo:

Representação dinâmica similar a das árvores binárias
Número máximo de filhos é fixado

Ex: nó de uma árvore com no máximo 3 filhos

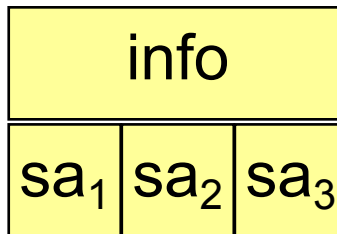


Árvores genéricas: formas de representação

Árvore com grau fixo:

Representação dinâmica similar a das árvores binárias
Número máximo de filhos é fixado

Ex: nó de uma árvore com no máximo 3 filhos



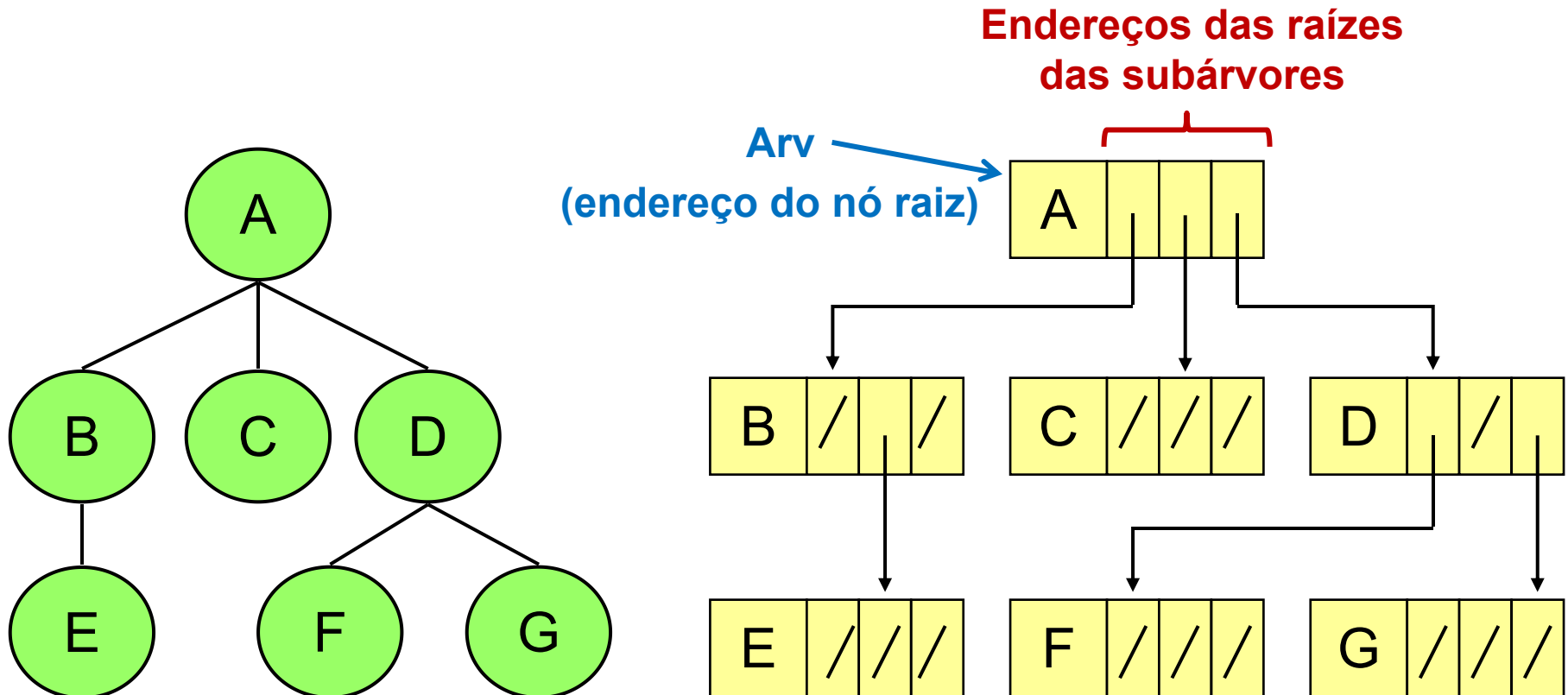
Desvantagens:

Necessidade de **estimar previamente o grau máximo**

Quantidade de filhos por nó pode variar muito

📖 Gera **grande desperdício**

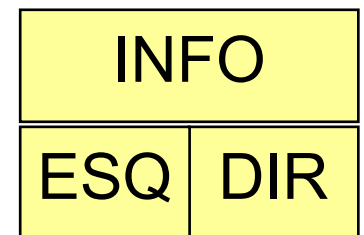
Exemplo de árvore com grau fixo



Árvores genéricas: formas de representação

Representação filho à esquerda / irmão à direita:

Cada nó aponta para a 1ª subárvore descendente (**filho à esquerda**) e para a subárvore vizinha (**irmão à direita**)



Árvores genéricas: formas de representação

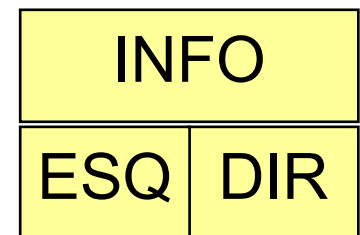
Representação filho à esquerda / irmão à direita:

Cada nó aponta para a 1ª subárvore descendente (**filho à esquerda**) e para a subárvore vizinha (**irmão à direita**)

Transforma a árvore genérica em uma **árvore binária**

Árvore vazia não é considerada na definição padrão

- Restrição **simplifica as implementações**
- Não limita a utilização** da estrutura em aplicações reais



Árvores genéricas: formas de representação

Representação filho à esquerda / irmão à direita:

Cada nó aponta para a 1ª subárvore descendente (**filho à esquerda**) e para a subárvore vizinha (**irmão à direita**)

Transforma a árvore genérica em uma **árvore binária**

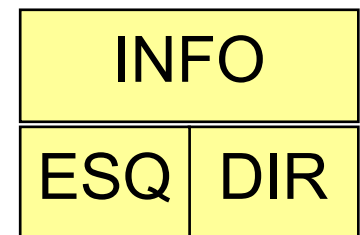
Árvore vazia não é considerada na definição padrão

- Restrição **simplifica as implementações**
- Não limita a utilização** da estrutura em aplicações reais

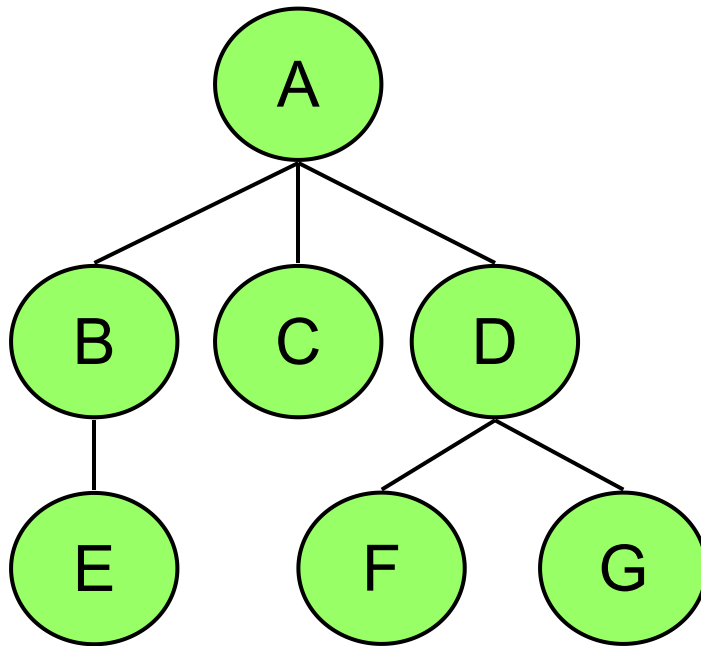
Nó raiz: irmão à direita = **NULL**

Nós folha: filho à esquerda = **NULL**

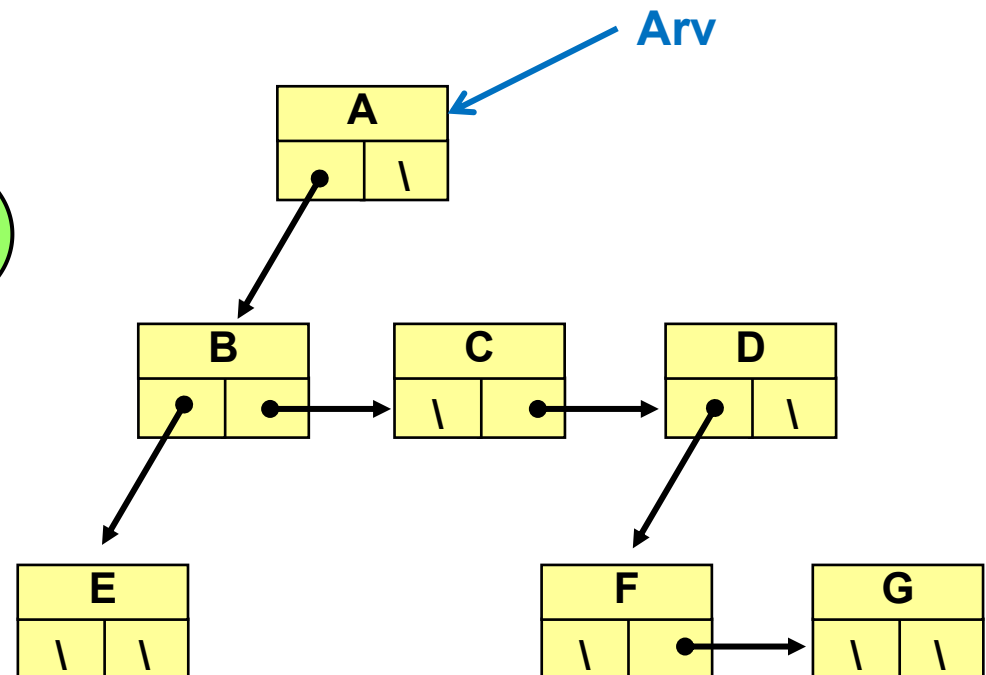
Nós de derivação: filho à esquerda \neq **NULL**



Exemplo de árvore genérica



Representação filho à esquerda e irmão à direita



Árvore genérica: estrutura de representação

Estrutura do **nó**:

Campo **INFO**: informações do nó raiz (**qualquer tipo**)

Campo **ESQ**: endereço do 1º filho à esquerda (sa_1)

Campo **DIR**: endereço do irmão à direita (sa_2 do nó pai)

ESQ	INFO	DIR
NO *	TIPO	NO *

Estrutura Nó

Árvore genérica: estrutura de representação

Estrutura do nó:

Campo **INFO**: informações do nó raiz (**qualquer tipo**)

Campo **ESQ**: endereço do 1º filho à esquerda (sa_1)

Campo **DIR**: endereço do irmão à direita (sa_2 do nó pai)

Implementação em C:

// Estrutura de um nó

```
struct no {  
    tipo info;  
    struct no *esq,  
    struct no *dir;  
};
```

// Árvore

```
typedef struct no * Arv;
```

ESQ	INFO	DIR
NO *	TIPO	NO *

Estrutura Nó

Árvore genérica: operações básicas

Arv *cria_arvore*(elem): cria uma nova árvore apenas com o nó raiz

int *insere*(A, sa): insere a subárvore *sa* como filha do nó *A*

void *exibe_arvore*(A): percorre a árvore *A*, elemento por elemento, apresentando o campo *info* de cada nó

int *busca*(A, elem): verifica se *elem* pertence à árvore *A*

void *libera_arvore*(&A): libera a memória alocada para a árvore *A*

Árvore genérica: implementação

Arv **cria_arvore** (int elem)

Aloca um novo nó;

SE falha na alocação **ENTÃO**

*retorna **NULL**;*

FIM_SE

*Campo **info** do novo nó = elem;*

*Campo **esq** do novo nó = **NULL**;*

*Campo **dir** do novo nó = **NULL**;*

retorna endereço do novo nó;

FIM

Árvore genérica: implementação

int **insere** (Arv A, Arv sa)

SE árvore vazia **ENTÃO** // A = **NULL**

retorna 0;

FIM_SE

*Irmão à direita (**dir**) do nó raiz de **sa** = filho à esquerda (**esq**) do nó **A**;*

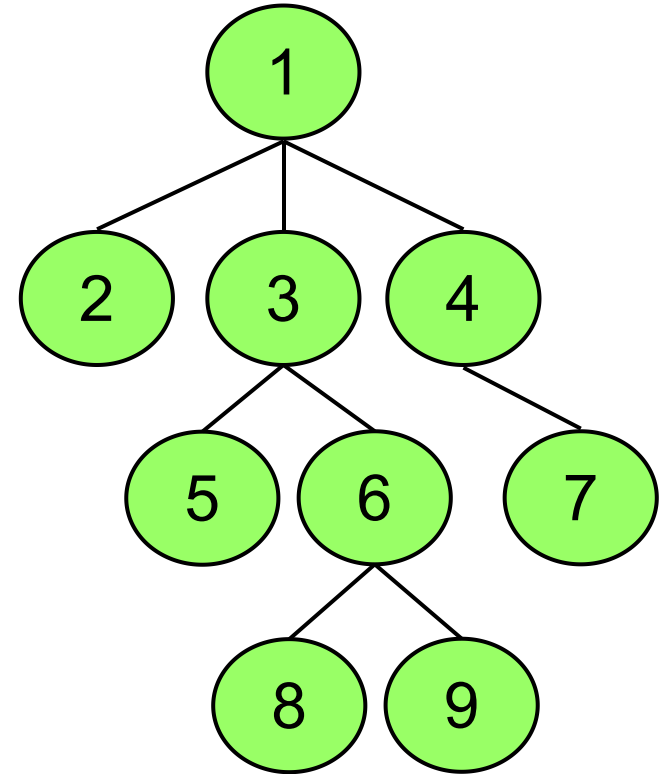
*Filho à esquerda (**esq**) do nó **A** = nó raiz de **sa**;*

retorna 1;

FIM

Exemplos de utilização das funções

Criação da árvore genérica:



Exemplos de utilização das funções

Criação da árvore genérica:

// Criação dos nós raízes das subárvores

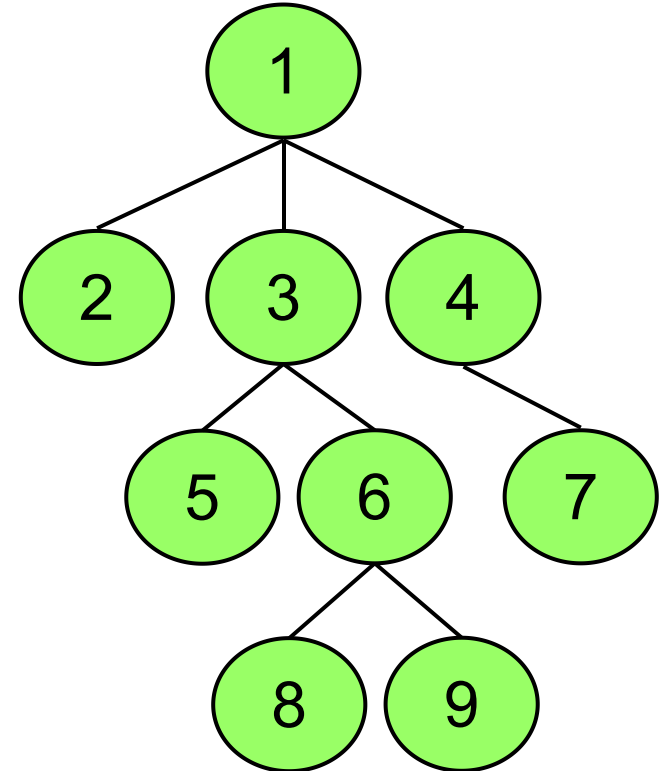
A = cria_arvore(1); B = cria_arvore(2);

C = cria_arvore(3); D = cria_arvore(4);

E = cria_arvore(5); F = cria_arvore(6);

G = cria_arvore(7); H = cria_arvore(8);

I = cria_arvore(9);



Exemplos de utilização das funções

Criação da árvore genérica:

// Criação dos nós raízes das subárvores

A = cria_arvore(1); B = cria_arvore(2);

C = cria_arvore(3); D = cria_arvore(4);

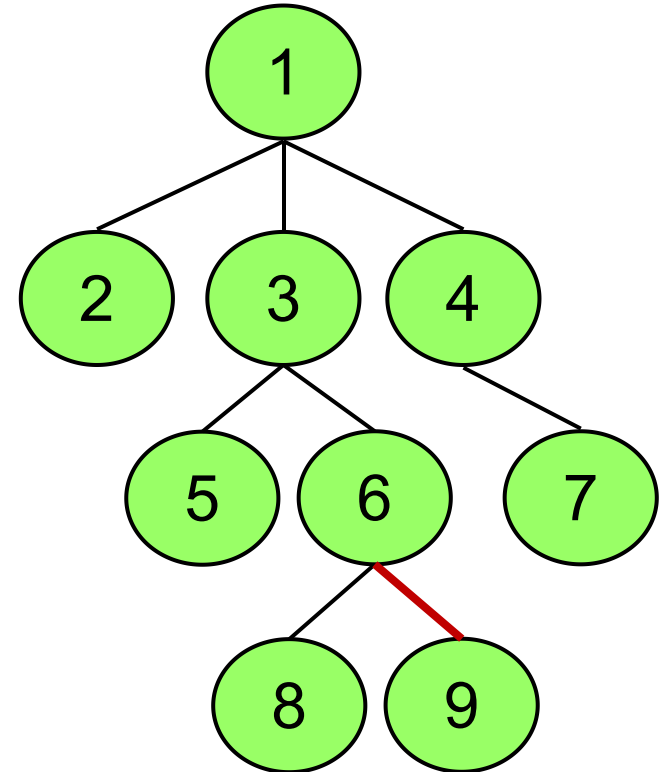
E = cria_arvore(5); F = cria_arvore(6);

G = cria_arvore(7); H = cria_arvore(8);

I = cria_arvore(9);

// Montagem da árvore

insere(F,I);



Exemplos de utilização das funções

Criação da árvore genérica:

// Criação dos nós raízes das subárvores

A = cria_arvore(1); B = cria_arvore(2);

C = cria_arvore(3); D = cria_arvore(4);

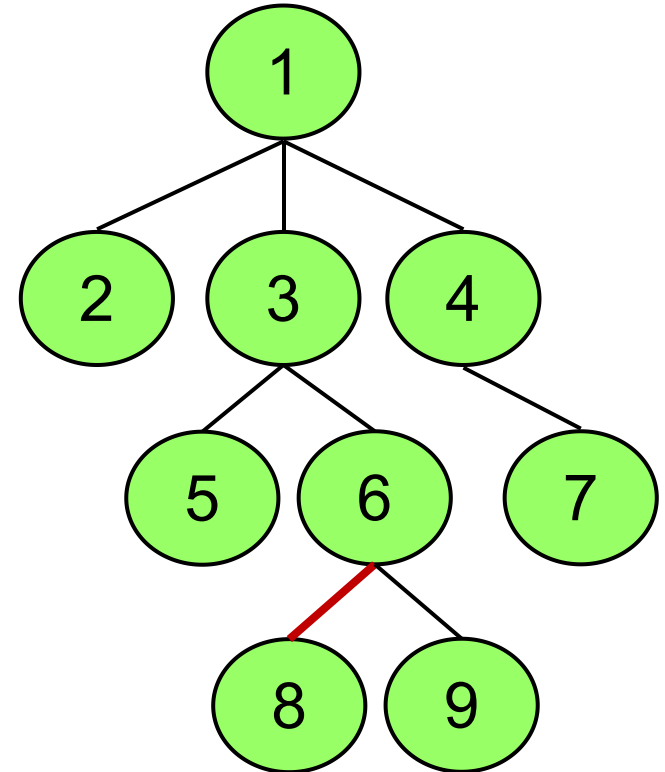
E = cria_arvore(5); F = cria_arvore(6);

G = cria_arvore(7); H = cria_arvore(8);

I = cria_arvore(9);

// Montagem da árvore

insere(F,I); insere(F,H);



Exemplos de utilização das funções

Criação da árvore genérica:

// Criação dos nós raízes das subárvores

A = cria_arvore(1); B = cria_arvore(2);

C = cria_arvore(3); D = cria_arvore(4);

E = cria_arvore(5); F = cria_arvore(6);

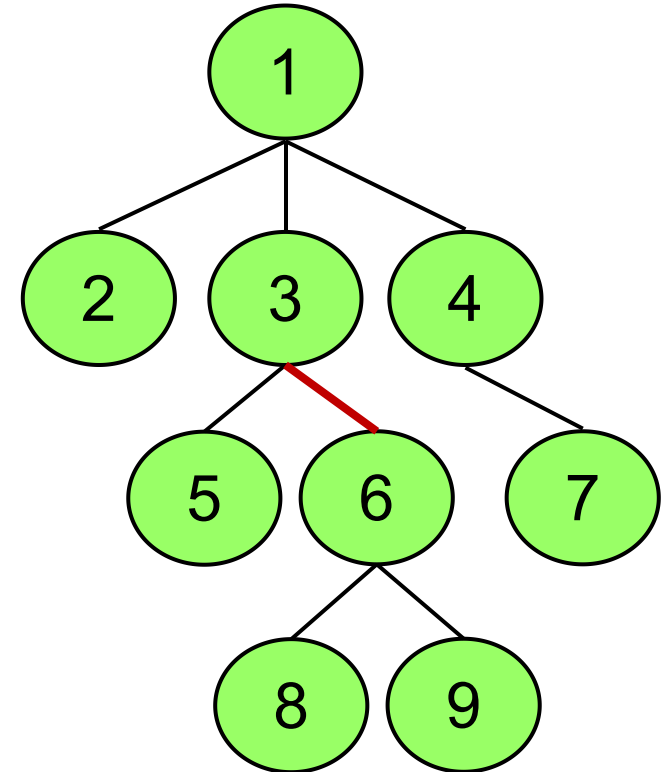
G = cria_arvore(7); H = cria_arvore(8);

I = cria_arvore(9);

// Montagem da árvore

insere(F,I); insere(F,H);

insere(C,F);



Exemplos de utilização das funções

Criação da árvore genérica:

// Criação dos nós raízes das subárvores

A = cria_arvore(1); B = cria_arvore(2);

C = cria_arvore(3); D = cria_arvore(4);

E = cria_arvore(5); F = cria_arvore(6);

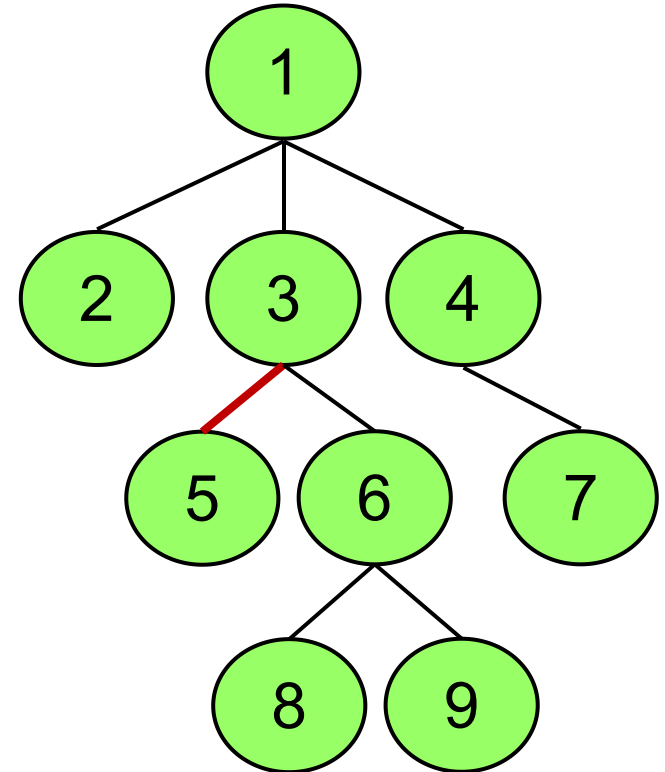
G = cria_arvore(7); H = cria_arvore(8);

I = cria_arvore(9);

// Montagem da árvore

insere(F,I); insere(F,H);

insere(C,F); insere(C,E);



Exemplos de utilização das funções

Criação da árvore genérica:

// Criação dos nós raízes das subárvores

A = cria_arvore(1); B = cria_arvore(2);

C = cria_arvore(3); D = cria_arvore(4);

E = cria_arvore(5); F = cria_arvore(6);

G = cria_arvore(7); H = cria_arvore(8);

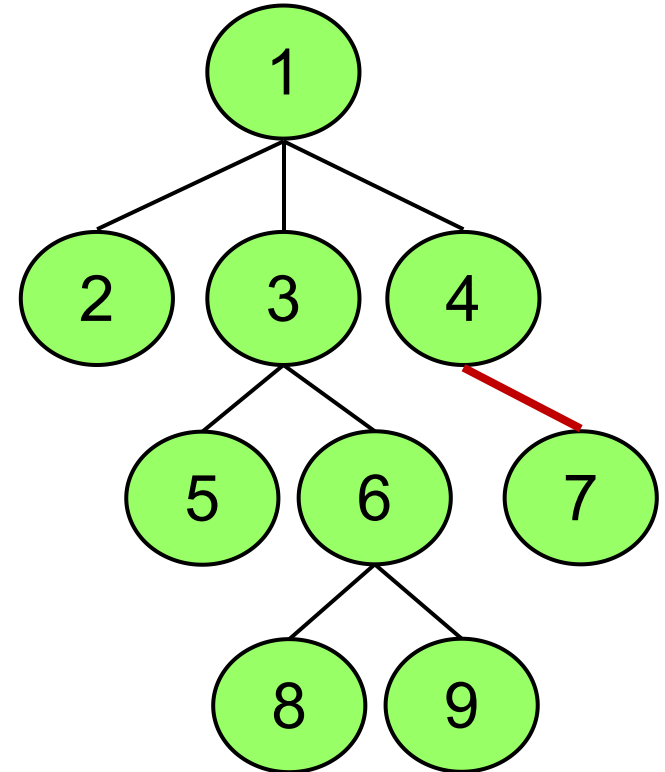
I = cria_arvore(9);

// Montagem da árvore

insere(F,I); insere(F,H);

insere(C,F); insere(C,E);

insere(D,G);



Exemplos de utilização das funções

Criação da árvore genérica:

// Criação dos nós raízes das subárvores

A = cria_arvore(1); B = cria_arvore(2);

C = cria_arvore(3); D = cria_arvore(4);

E = cria_arvore(5); F = cria_arvore(6);

G = cria_arvore(7); H = cria_arvore(8);

I = cria_arvore(9);

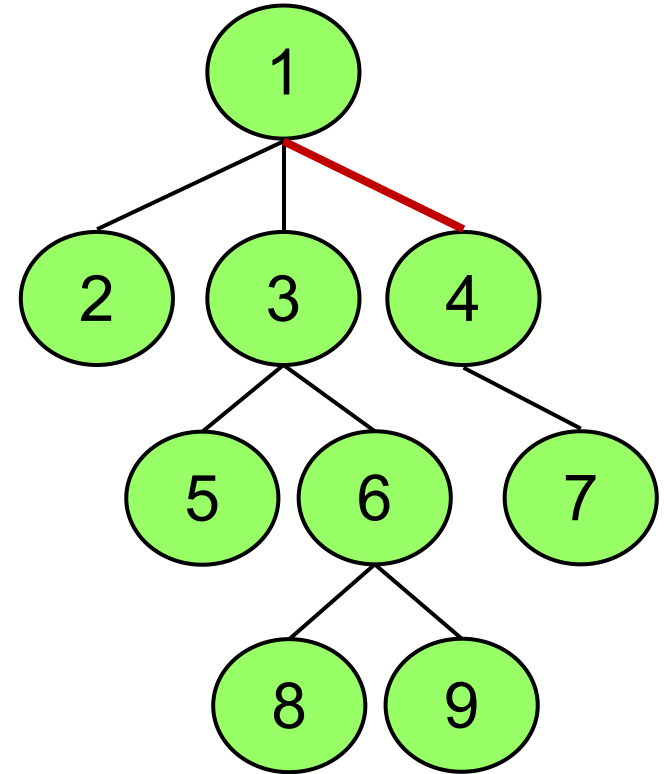
// Montagem da árvore

insere(F,I); insere(F,H);

insere(C,F); insere(C,E);

insere(D,G);

insere(A,D);



Exemplos de utilização das funções

Criação da árvore genérica:

// Criação dos nós raízes das subárvores

A = cria_arvore(1); B = cria_arvore(2);

C = cria_arvore(3); D = cria_arvore(4);

E = cria_arvore(5); F = cria_arvore(6);

G = cria_arvore(7); H = cria_arvore(8);

I = cria_arvore(9);

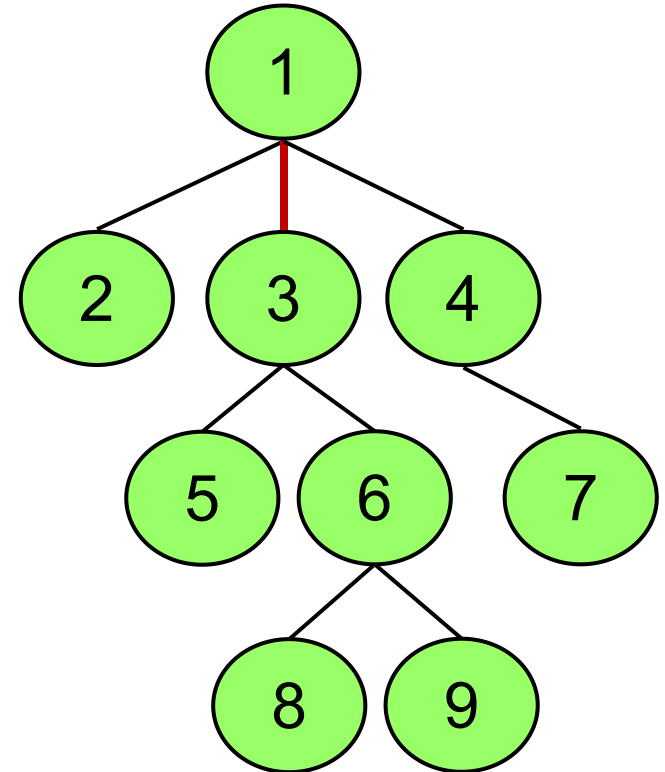
// Montagem da árvore

insere(F,I); insere(F,H);

insere(C,F); insere(C,E);

insere(D,G);

insere(A,D); insere(A,C);



Exemplos de utilização das funções

Criação da árvore genérica:

// Criação dos nós raízes das subárvores

A = cria_arvore(1); B = cria_arvore(2);

C = cria_arvore(3); D = cria_arvore(4);

E = cria_arvore(5); F = cria_arvore(6);

G = cria_arvore(7); H = cria_arvore(8);

I = cria_arvore(9);

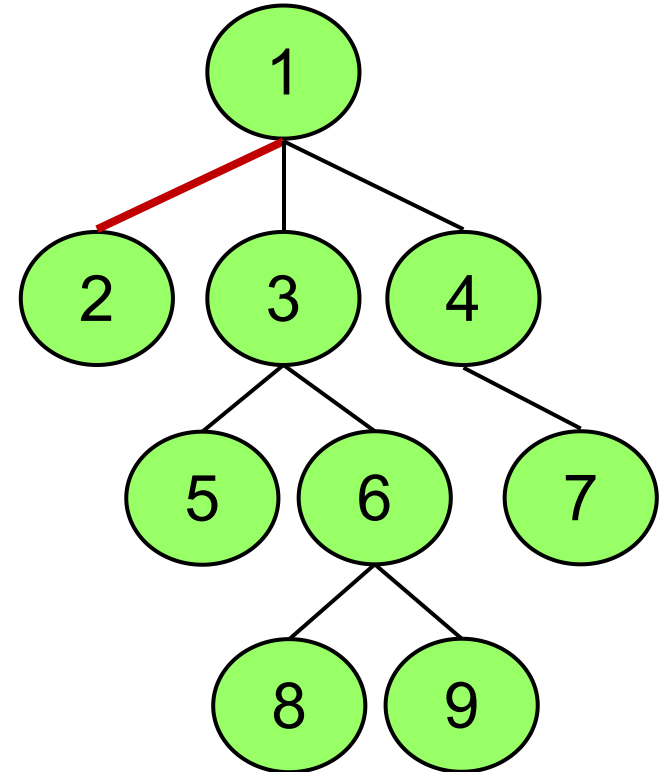
// Montagem da árvore

insere(F,I); insere(F,H);

insere(C,F); insere(C,E);

insere(D,G);

insere(A,D); insere(A,C); insere (A,B);



Exemplos de utilização das funções

Criação da árvore genérica:

// Criação dos nós raízes das subárvores

A = cria_arvore(1); B = cria_arvore(2);

C = cria_arvore(3); D = cria_arvore(4);

E = cria_arvore(5); F = cria_arvore(6);

G = cria_arvore(7); H = cria_arvore(8);

I = cria_arvore(9);

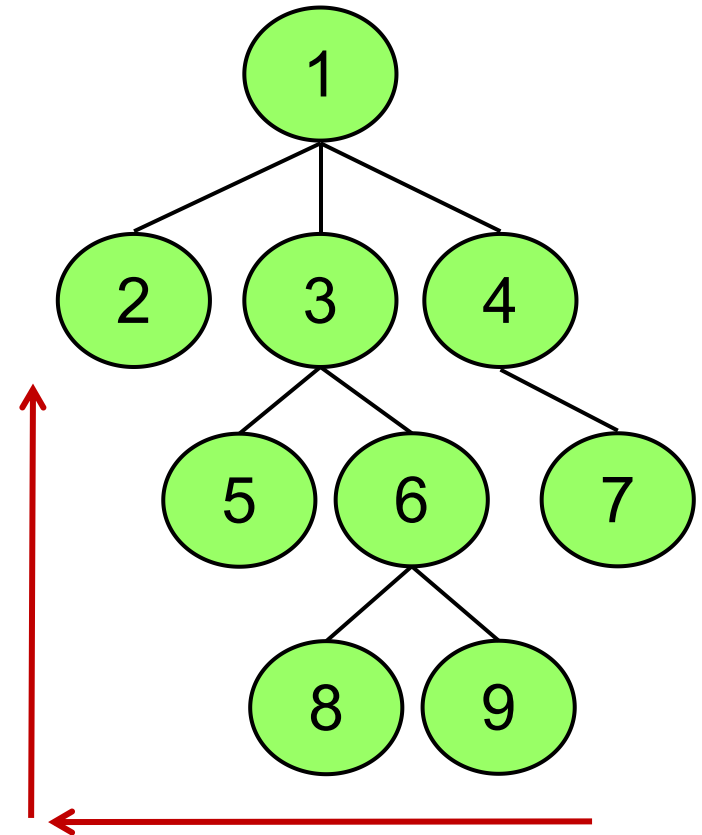
// Montagem da árvore

insere(F,I); insere(F,H);

insere(C,F); insere(C,E);

insere(D,G);

insere(A,D); insere(A,C); insere (A,B);



Ordem: Dir → Esq

Base (folhas) → topo (raiz)

Árvore genérica: implementação

Demais operações são **idênticas àquelas implementadas no **TAD árvore binária****

Árvore genérica: implementação

exibe_arvore (Arv A)

SE árvore vazia **ENTÃO** // A = **NULL**

 escreva("árvore vazia");

FIM_SE

escreva ("< "); // Abertura de contexto na notação textual

Exibe o campo **info** de A;

Exibe subárvore **filha à esquerda**;

escreva(">"); // Fechamento de contexto na notação textual

Exibe subárvore **irmã à direita**;

FIM

Árvore genérica: implementação

exibe_arvore (Arv A)

SE árvore **NÃO** vazia **ENTÃO** // $A \neq \text{NULL}$

escreva ("**<**"); // Abertura de contexto na notação textual

Exibe o campo **info** de A;

Exibe subárvore **filha à esquerda**;

escreva ("**>**"); // Fechamento de contexto na notação textual

Exibe subárvore **irmã à direita**;

FIM_SE

FIM

**Fechamento de escopo ocorre
antes de exibir irmão à direita**

Árvore genérica: implementação

int **busca** (Arv A, int elem)

...

SE árvore vazia **ENTÃO** // A = **NULL**

retorna 0;

FIM_SE

SE campo **info** de A = elem **ENTÃO**

retorna 1;

FIM_SE

Busca no **filho à esquerda**;

SE encontrou **ENTÃO**

retorna 1;

FIM_SE

Busca no **irmão a direita**;

SE encontrou **ENTÃO**

retorna 1;

FIM_SE

retorna 0;

FIM

...

Árvore genérica: implementação

libera_arvore (*Arv* * *A*)

SE árvore NÃO vazia **ENTÃO** // **A* ≠ **NULL**

Libera subárvore filha à esquerda;

Libera subárvore irmã à direita;

*Libera memória alocada para o **nó raiz**; // free(**A*);*

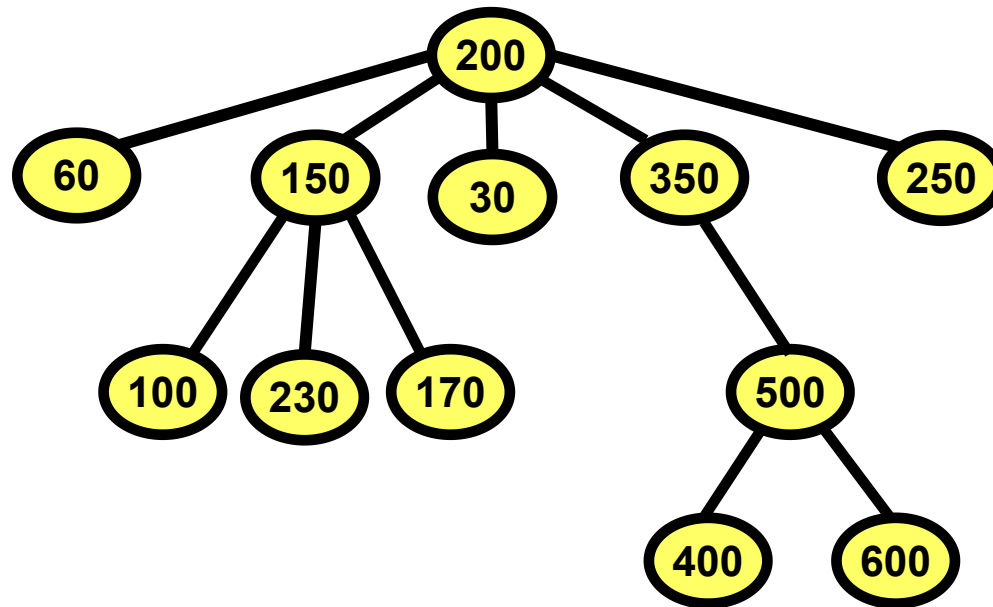
FIM_SE

*Faz conteúdo de *A* = **NULL**; // **A* = **NULL**;*

FIM

Exercícios

1. Apresente a árvore através das representações de contiguidade física, baseada nos pais, g e filho à esquerda / irmão à direita.



2. Implemente as operações básicas de uma árvore genérica de números inteiros, utilizando a representação filho à esquerda / irmão à direita. Acrescente uma função que determine a altura da árvore.

Bibliografia

Slides adaptados do material da Profa. Dra. Gina Maira Barbosa de Oliveira e da Profa. Dra. Denise Guliato.

EDELWEISS, N; GALANTE, R. Estruturas de dados (Série Livros Didáticos Informática UFRGS, v. 18), Bookman, 2008.

CORMEN, T.H. et al. Algoritmos: Teoria e Prática, Campus, 2002

ZIVIANI, N. Projeto de algoritmos: com implementações em Pascal e C (2ª ed.), Thomson, 2004

CELES, W.; CERQUEIRA, R. & RANGEL, J. L. Introdução a Estruturas de Dados: com técnicas de programação em C, Elsevier, 2004