GRUPO EDUCACIONAL Opet

ALGORITMOS

PROFª: SIMONE DOMINICO

ESTRUTURA DE DADOS

GRUPO EDUCACIONAL Opet

PILHA

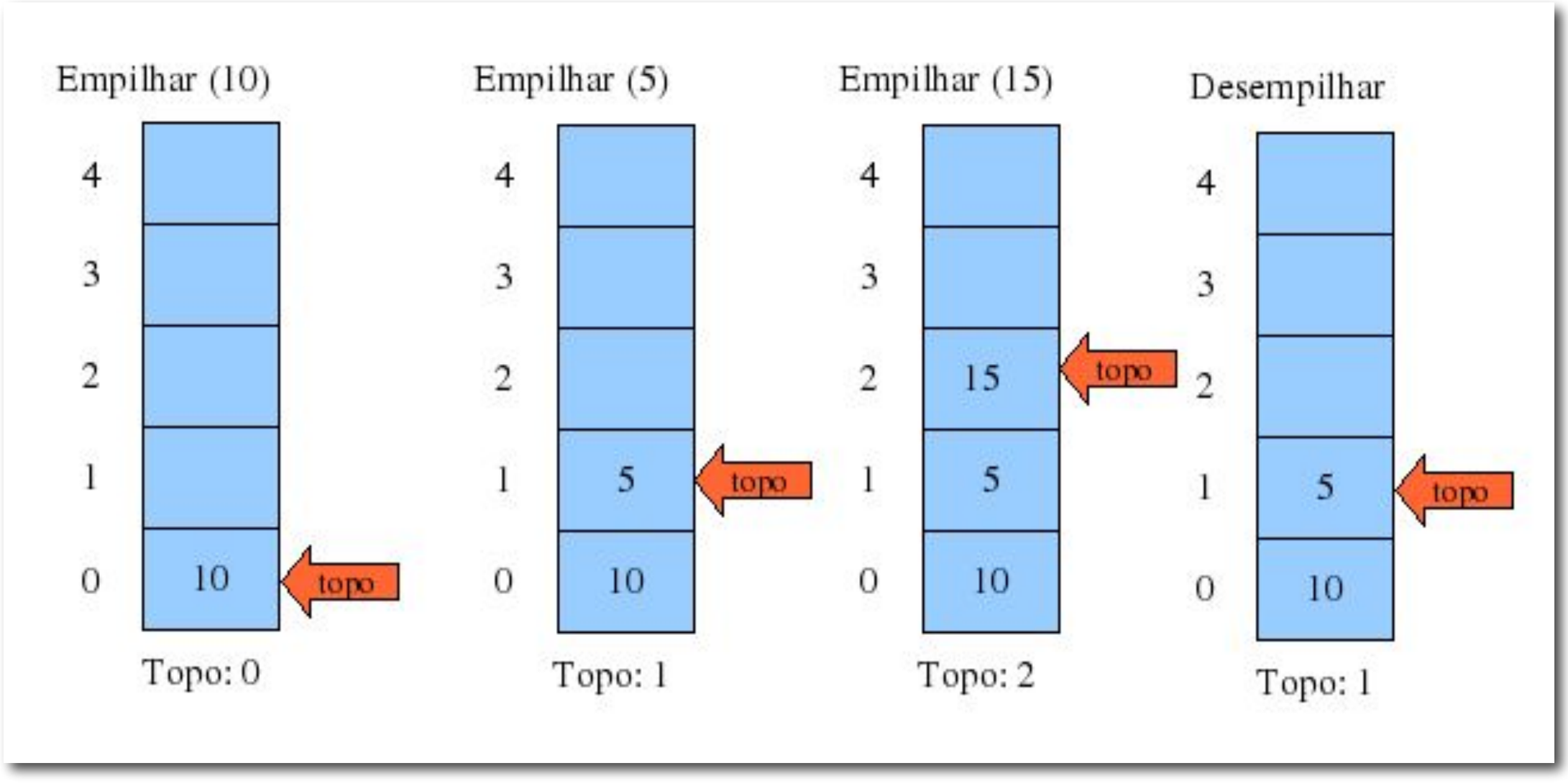
PILHA

- Uma pilha é uma estrutura de dados que admite remoção de elementos e inserção de novos objetos.
- Mais especificamente, uma pilha (= stack) é uma estrutura sujeita à seguinte regra de operação:
  - sempre que houver uma remoção, o elemento removido é o que está na estrutura há menos tempo.

- Em outras palavras, o primeiro objeto a ser inserido na pilha é o último a ser removido. Essa política é conhecida pela sigla LIFO (= Last-In-First-Out).

Pilha

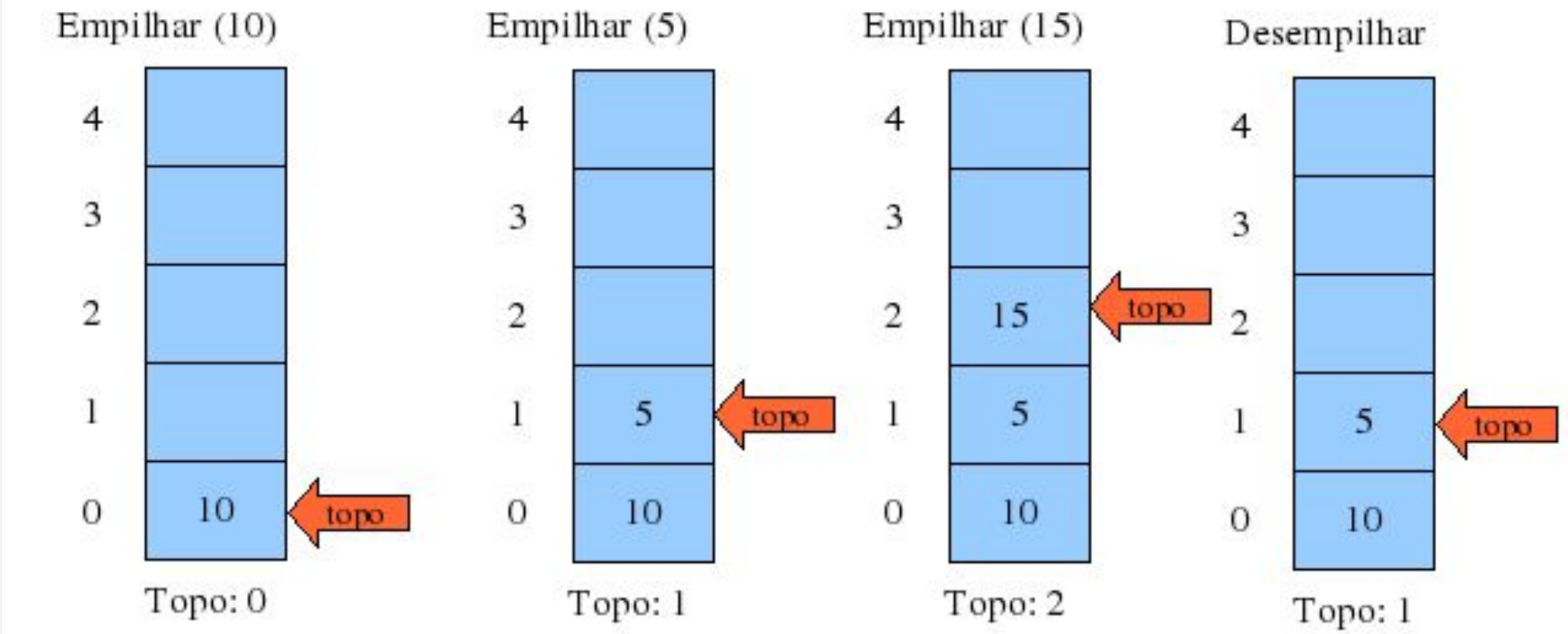
PILHA


Empilhar (10)
4
3
2
1
0
10
topo
Topo: 0
Empilhar (5)
4
3
2
1
0
5
topo
10
Topo: 1
Empilhar (15)
4
3
2
1
0
15
topo
5
10
Topo: 2
Desempilhar
4
3
2
1
0
5
topo
10
Topo: 1

PILHA – IMPLEMENTAÇÃO COM VETOR

- Suponha que nossa pilha está armazenada em um vetor pilha[N].
- Digamos que a parte do vetor ocupada pela pilha é: pilha[t]:
  - O índice t+1 indica a primeira posição vaga da pilha e t é o topo da pilha.
  - A pilha está vazia se t vale 0 e cheia se t vale N.
- Para remover, ou tirar, um elemento da pilha:
  - essa operação é conhecida como desempilhar (= to pop) faça
  - x = pilha[--t];
- Para inserir, ou colocar, um objeto y na pilha — a operação é conhecida como empilhar (= to push) — faça
- pilha[t++] = y;

PILHA – IMPLEMENTAÇÃO COM VETOR



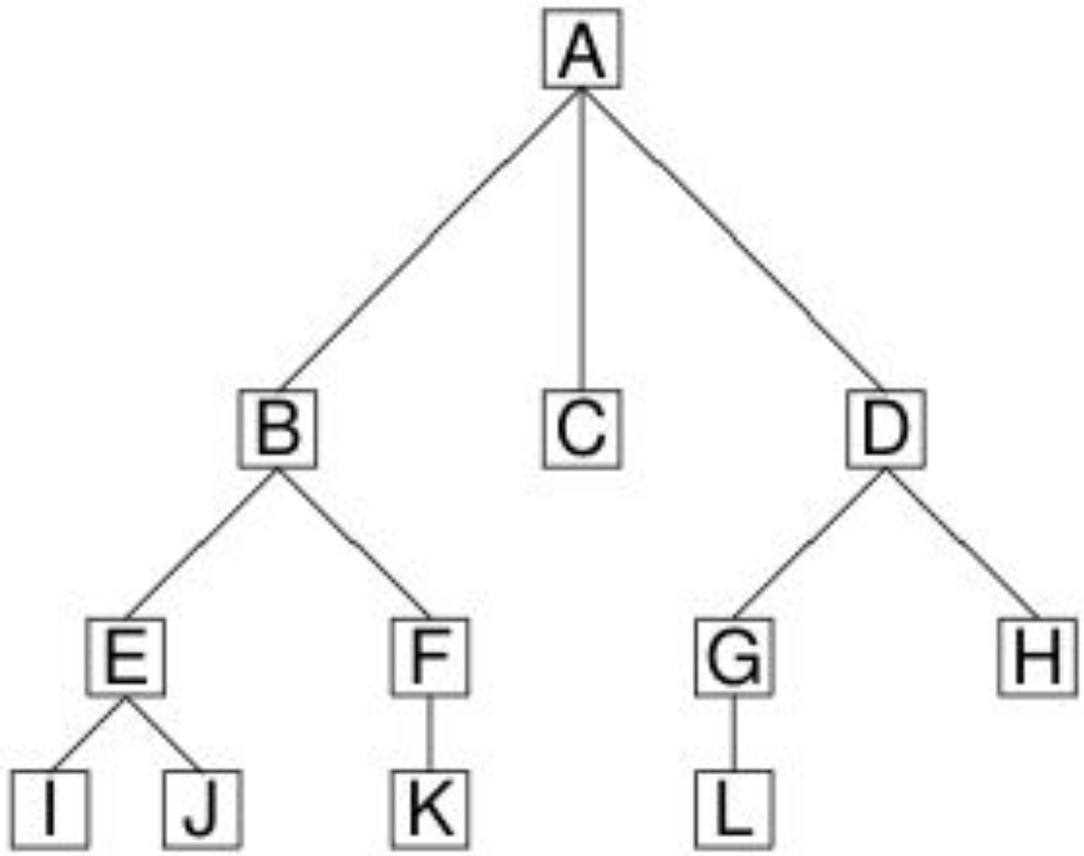
GRUPO EDUCACIONAL

Opet

Árvore Binária

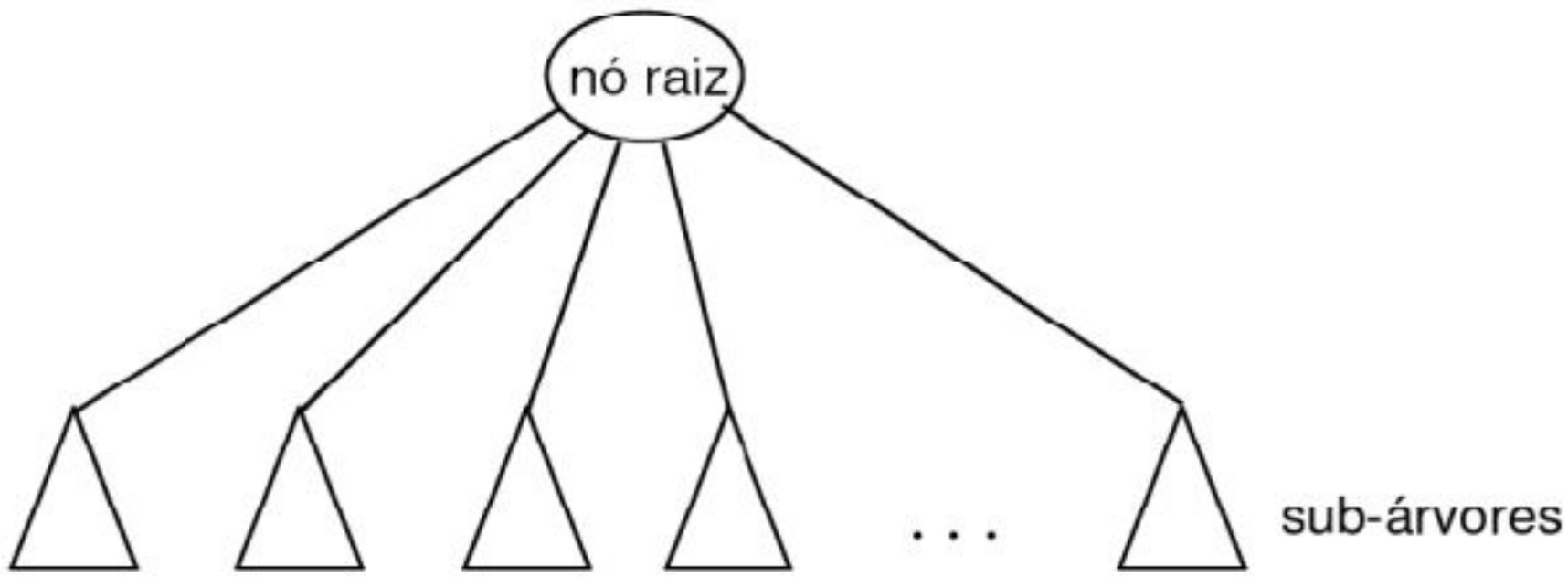
Árvore – Definição

- Árvore Uma árvore do tipo T é constituída de uma estrutura vazia, ou um elemento, ou um nó do tipo T chamado raiz com um número finito de árvores do tipo T associadas, chamadas as sub-árvores da raiz.



Definição Recursiva de Árvore

- Um conjunto de nós tal que:
  - existe um nó r, denominado raiz, com zero ou mais subárvores, cujas raízes estão ligadas a r
  - os nós raízes destas sub-árvores são os filhos de r
  - os nós internos da árvore são os nós com filhos
  - as folhas ou nós externos da árvore são os nós sem filhos



Definição estrutura Árvore

SubÁrvore

- Seja a árvore acima T = {A, B, ...}
- A árvore T possui duas subárvores:
  - Tb e Tc onde Tb = { B } e Tc = {C, D, ...}
- A subárvore Tc possui 3 subárvores:
  - – Td, Tf e Te onde Td = {D, G, H}, Tf = {F, I}, Te = {E}
- As subárvores Tb, Te, Tg, Th, Ti possuem apenas o nó raiz e nenhuma subárvore.

Definição estrutura Árvore

Conceitos Básicos

- Nós filhos, pais, tios, irmãos e avô
- Grau de saída (número de filhos de um nó)
- Nó folha (grau de saída nulo) e nó interior (grau de saída diferente de nulo)
- Grau de uma árvore (máximo grau de saída)
- Floresta (conjunto de zero ou mais árvores)

Definição estrutura Árvore

Conceitos Básicos

- Caminho
  - Uma sequência de nós distintos v1, v2, ..., vk, tal que existe sempre entre nós consecutivos (isto é, entre v1 e v2, entre v2 e v3, ... , v(k-1) e vk) a relação "é filho de“ ou "é pai de" é denominada um caminho na árvore.
- Comprimento do caminho
  - Um caminho de vk vértices é obtido pela sequência de k-1 pares. O valor k -1 é o comprimento do caminho.
- Nível ou profundidade de um nó
  - número de nós do caminho da raiz até o nó.

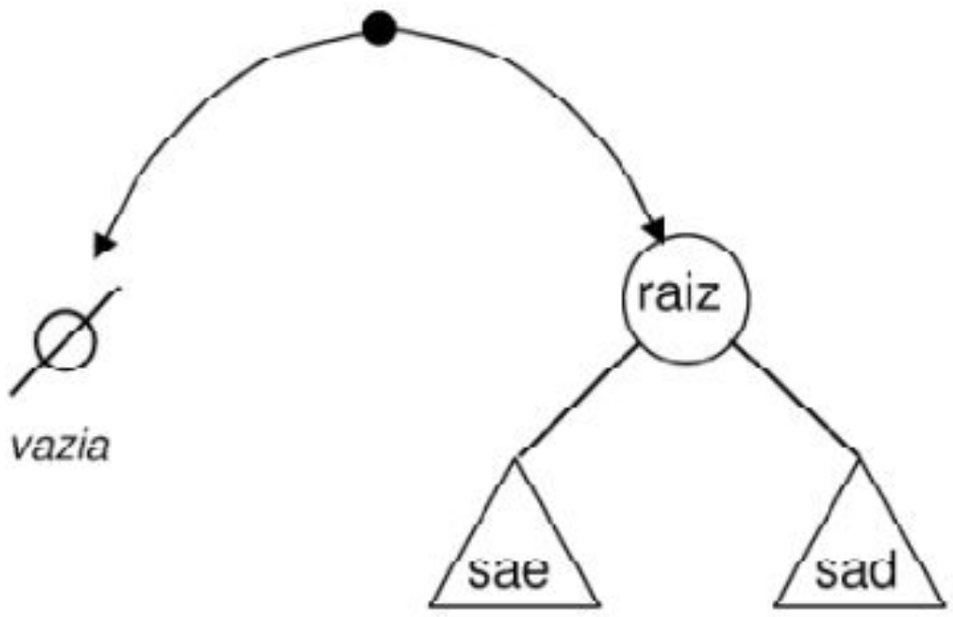
Exemplo

- Árvore binária representando expressões aritméticas binárias
  - Nós folhas representam os operandos
  - Nós internos representam os operadores

(3+6)*(4-1)+5

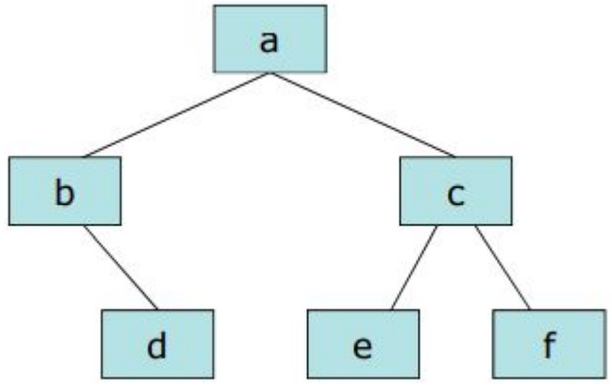
Árvore binária

- Uma árvore em que cada nó tem zero, um ou dois filhos
- Uma árvore binária é:
  - uma árvore vazia; ou – um nó raiz com duas sub-árvores:
- a subárvore da direita (sad) • a subárvore da esquerda (sae)



Árvore binária

- Propriedade das árvores
  - Existe apenas um caminho da raiz para qualquer nó
- Altura de uma árvore
  - comprimento do caminho mais longo da raiz até uma das folhas
  - a altura de uma árvore com um único nó raiz é zero
  - a altura de uma árvore vazia é - 1
- Esforço computacional necessário para alcançar qualquer nó da árvore é proporcional à altura da árvore
- Exemplo: – h = 2



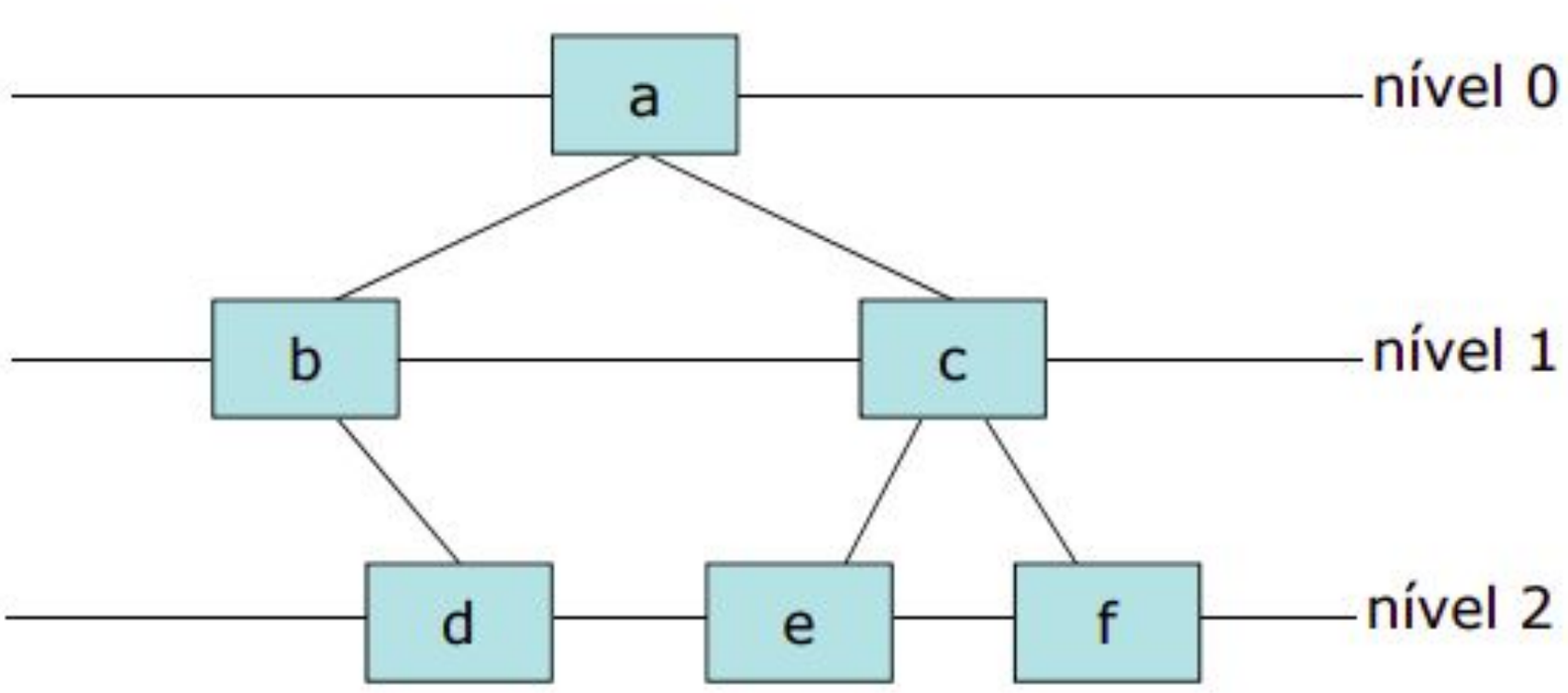
Árvore binária em C

- Representação: ponteiro para o nó raiz
- Representação de um nó na árvore:
  - Estrutura em C contendo
    - A informação propriamente dita (exemplo: um caractere, ou inteiro)
    - Dois ponteiros para as sub-árvores, à esquerda e à direita

```
struct arv {
  char info;
  struct arv* esq;
  struct arv* dir;
};
```

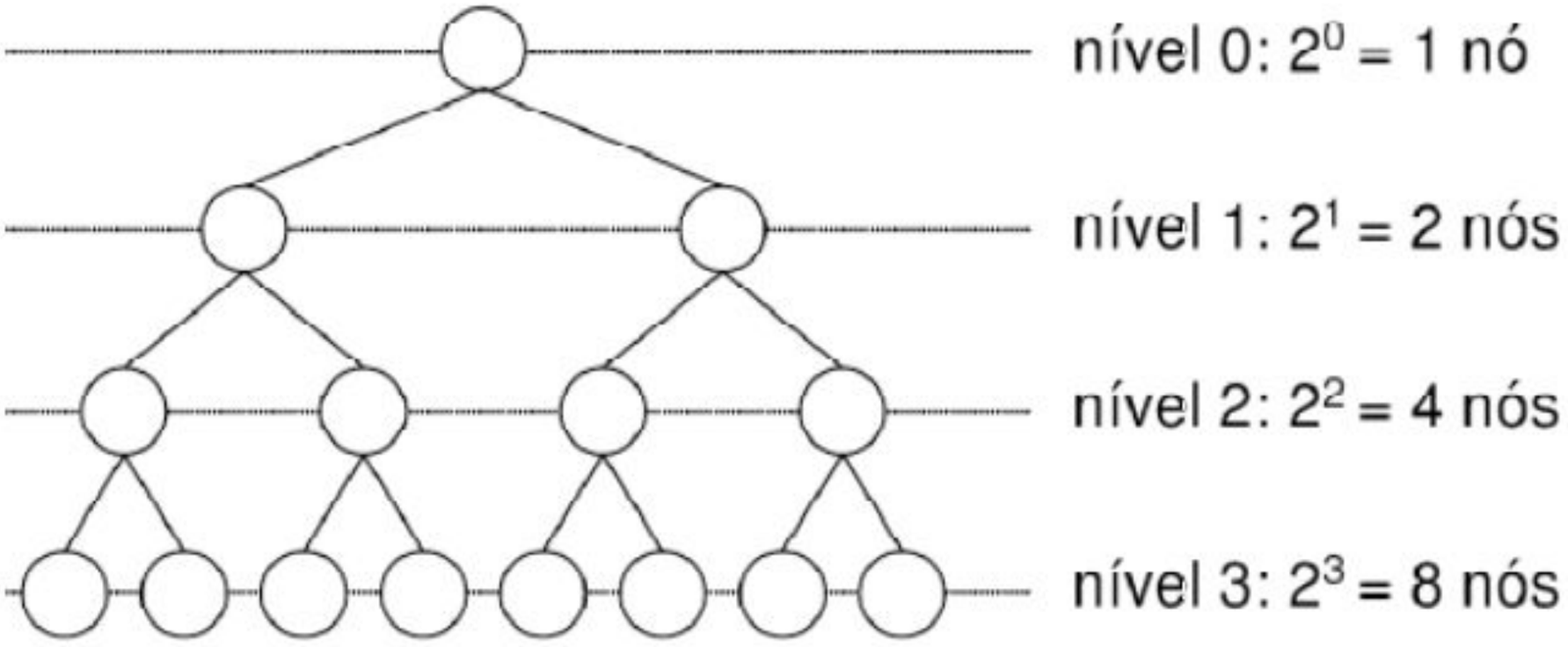
Árvore binária em C

- Nível de um nó
  - a raiz está no nível 0, seus filhos diretos no nível 1, ...
  - o último nível da árvore é a altura da árvore



Árvore binária em C

- Árvore Cheia
  - todos os seus nós internos têm duas sub-árvores associadas
  - número n de nós de uma árvore cheia de altura h – n = 2 h+1 -1



Árvore binária em C

- **typedef struct arv Arv;**

//Cria uma árvore vazia Arv*

- **arv_criavazia (void);**

//cria uma árvore com a informação do nó raiz c, e //com subárvore esquerda e e subárvore direita d

- **Arv* arv_cria (char c, Arv* e, Arv* d);**

//libera o espaço de memória ocupado pela árvore a

- **Arv* arv_libera (Arv* a);**

//retorna true se a árvore estiver vazia e false //caso contrário

- **int arv_vazia (Arv* a);**

//indica a ocorrência (1) ou não (0) do caracter c

- **int arv_pertence (Arv* a, char c);**

//imprime as informações dos nós da árvore

- **void arv_imprime (Arv* a);**