Arvore binária

Profa. Fabrícia Damando Santos
fabriciadamando@gmail.com

Introdução

- Os Tipos Abstratos de Dados estudados foram Listas Simplesmente e Duplamente Encadeadas, Fila e Pilha, sendo que o que muda nesses Tipos as operações, pois a Estrutura de Dados base para tais tipos são as listas lineares, sejam elas estáticas ou dinâmicas. Embora tais listas apresentem vantagens quanto ao uso, à manipulação e à alocação, ainda possuem problemas:
- · **Lista encadeada**
  - o Eficiente para inserção e remoção dinâmica de elementos, mas ineficiente para busca;
- · **Lista seqüencial (ordenada)**
  - o Eficiente para busca, mas ineficiente para inserção e remoção de elementos

- Em busca de contornar essas desvantagens, foi proposto o conceito de **Árvores, que apresenta solução eficiente para inserção, remoção e busca**

- As árvores são estruturas de dados adequadas para a representação de hierarquias.

- A forma mais natural para definirmos uma estrutura de árvore é usando recursividade.

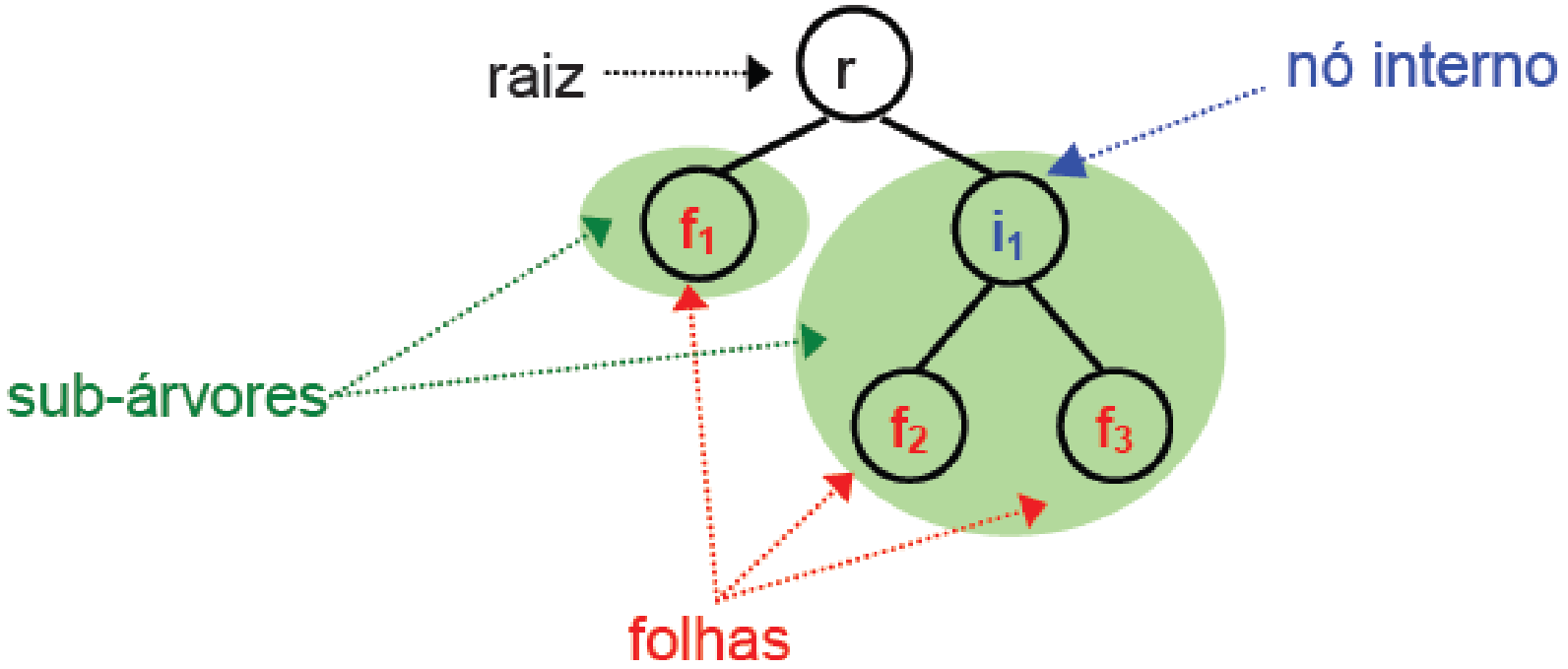
Definição

- Uma árvore é composta por um conjunto de nós. Existe um nó **r, denominado raiz, que contém zero ou mais sub-árvores, cujas raízes são ligadas diretamente a r.**

- Esses nós **raízes das sub-árvores são ditos filhos do nó pai, r.**
- **Nós com filhos são comumente chamados de nós internos e nós que não têm filhos são chamados de folhas, ou nós externos.**

Definição

- Uma árvore T é um conjunto finito de nós tais que:
  - T=0, e a árvore é vazia
  - Existe um nó especial r que é a raiz de T
  - Os restantes constituem um conjunto vazio ou são divididos em m ≥ 1 conjuntos não vazios
  - Estes conjuntos não vazios são as subárvores
  - Cada subárvore é uma árvores


raiz
r
nó interno
f1
i1
sub-árvores
f2
f3
folhas

Conceitos

- **NÍVEL: o nível de um nó T é definido como:**
  - O nível de um nó raiz é **0;**
  - O nível de um nó não raiz é dado por (nível de seu nó **PAI + 1).**

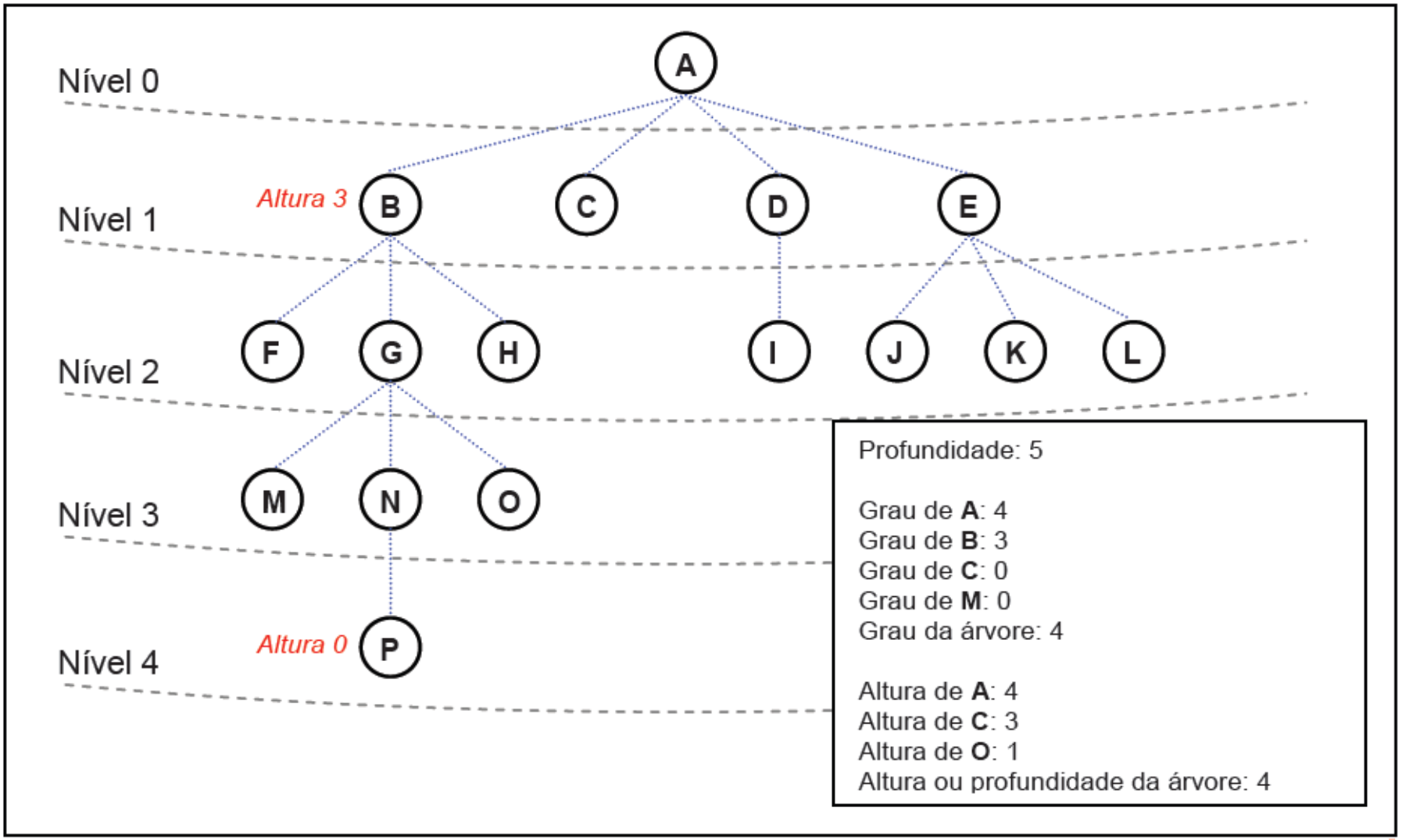
- **GRAU: o grau de um nó T de uma árvore é igual ao número de filhos do nó T;**

- **GRAU DA ÁRVORE: o grau de uma árvore T é o grau máximo entre os graus de todos os seus nós;**

- **ALTURA : A altura de um nó v em uma árvore binária é a distância entre v e o seu descendente mais afastado. Mas precisamente, a altura de v é o número de passos do mais longo caminho que leva de v até uma folha. Os nós folha sempre têm altura igual a 0;**

- **ALTURA ou PROFUNDIDADE DE UMA ÁRVORE: A altura de uma árvore T é dada pela altura da raiz da árvore.**

  - Denota-se a altura de uma árvore com raiz dada pelo nó **T por h(T), e a altura de uma sub-árvore com raiz T^1 por h(T^1).**

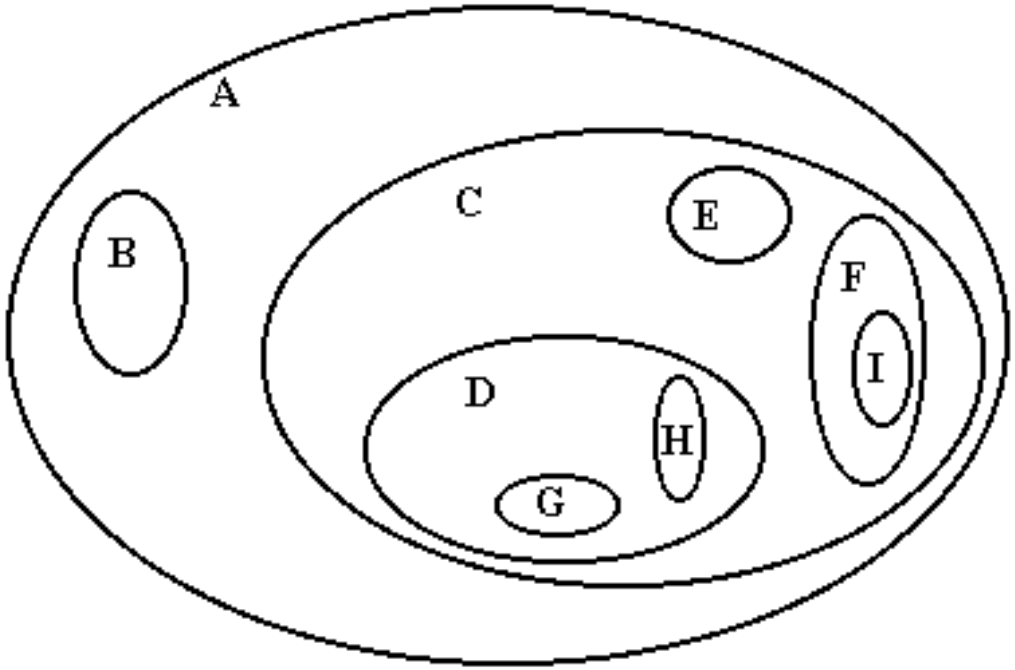

Nível 0
Nível 1
Nível 2
Nível 3
Nível 4
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Altura 3
Altura 0
Profundidade: 5
Grau de **A**: 4
Grau de **B**: 3
Grau de **C**: 0
Grau de **M**: 0
Grau da árvore: 4
Altura de **A**: 4
Altura de **C**: 3
Altura de **O**: 1
Altura ou profundidade da árvore: 4

Tipos de árvores

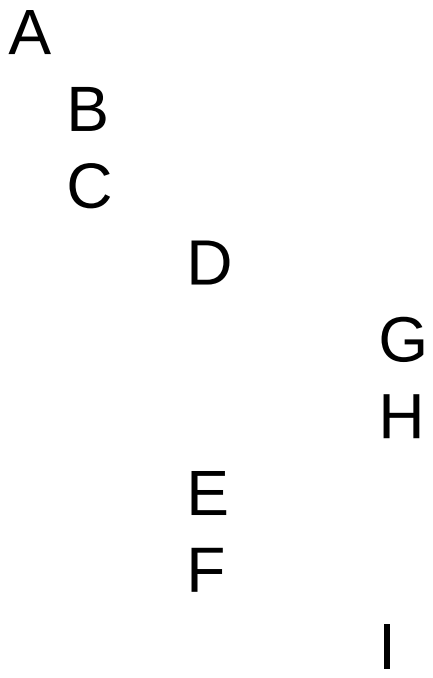
- Arvores binárias, onde cada nó tem, no máximo, dois filhos.
- Árvores genéricas, onde o número de filhos é indefinido.

Representação

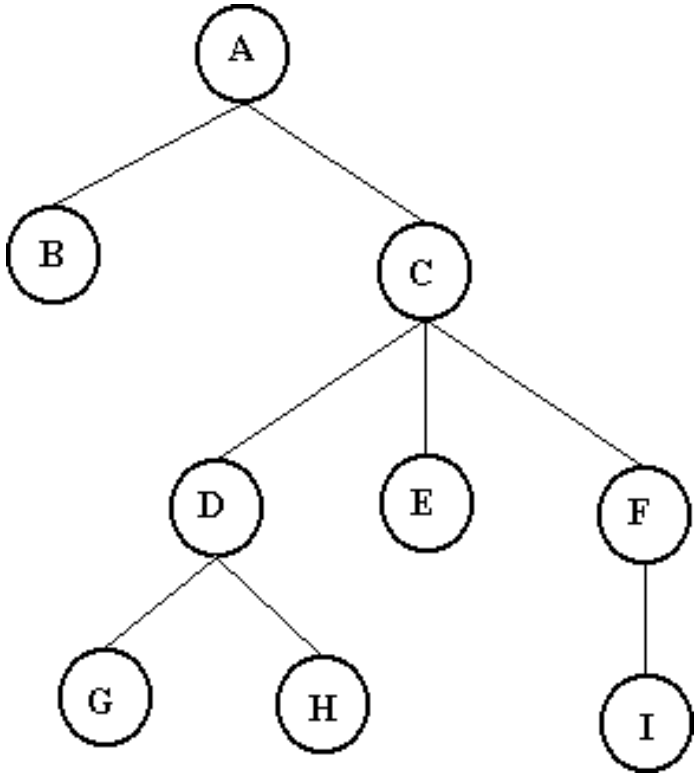
- Representação pelo diagrama de inclusão – Diagrama de Venn



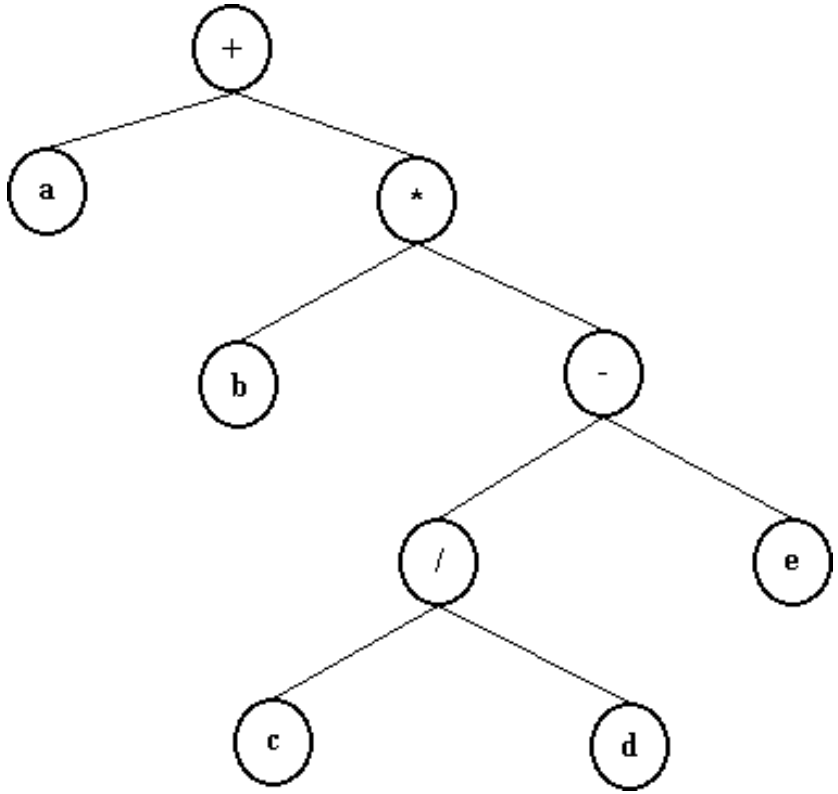
- Alinhamento dos nós



- Representação hierárquica

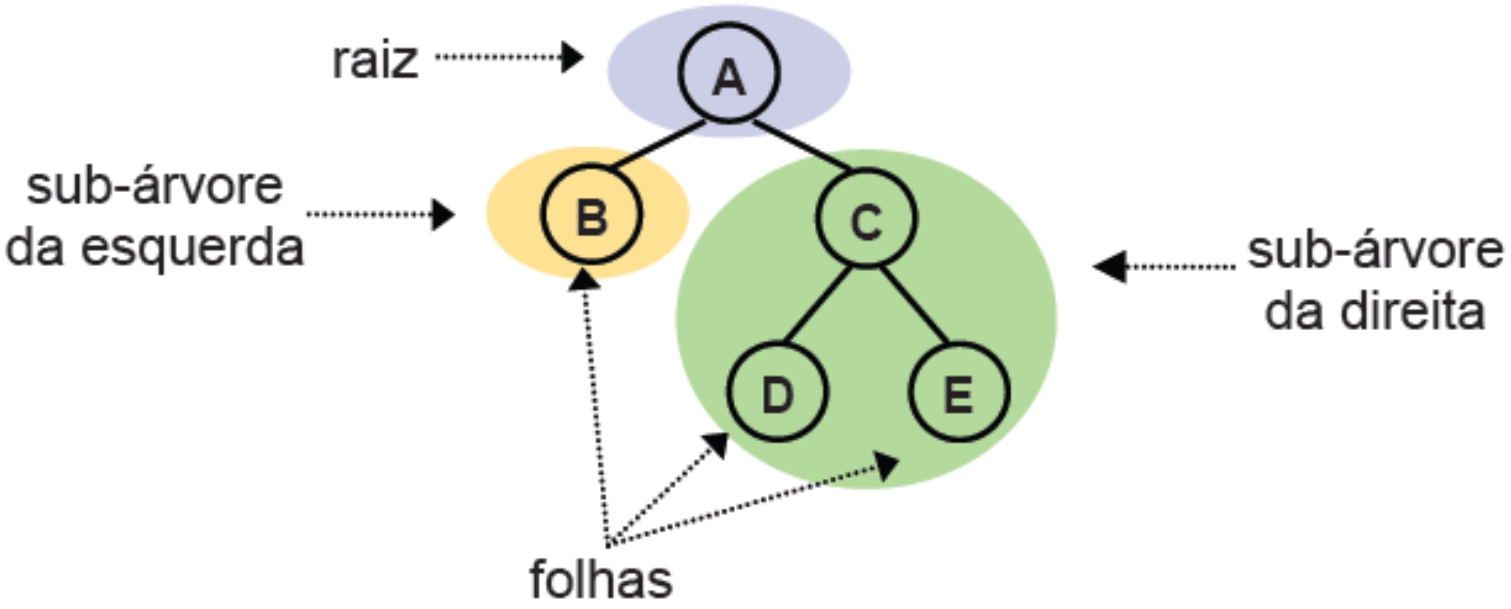


- Representação da expressão aritmética:
  (a + (b *((c / d) - e)))



Árvore binária

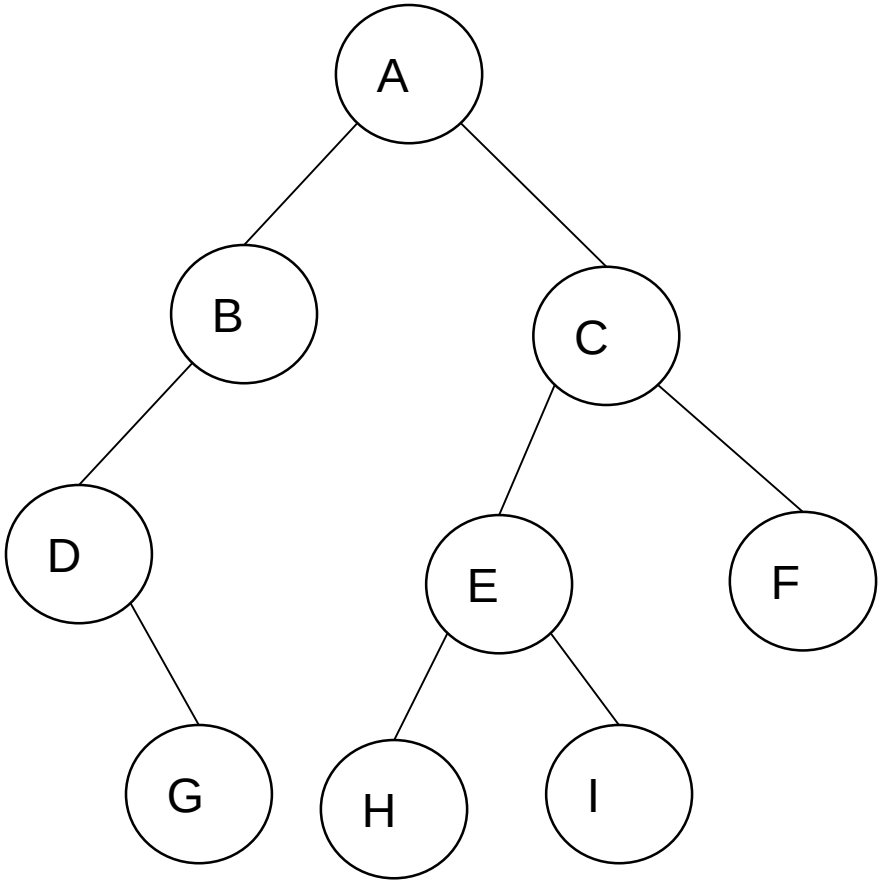
- Uma **árvore binária é um conjunto finito de elementos que ou é vazio ou é dividido em três subconjuntos disjuntos**
  - A **raiz da árvore;**
  - Uma **árvore binária chamada de sub-árvore da direita.**
  - Uma **árvore binária é um conjunto finito de elementos que ou é vazio ou é dividido em três subconjuntos disjuntos:**


raiz
A
sub-árvore
da esquerda
B
C
sub-árvore
da direita
D
E
folhas

Árvore Binária

- Uma árvore binária T é um conjunto finito de nós tais que:
  - T=0, e a árvore é vazia
  - Existe um nó especial r que é a raiz de T
  - Os restantes podem ser divididos em dois subconjuntos TE e TD
  - TE e TD são a subárvores esquerda e a direita
  - Cada subárvore é também uma árvore binária

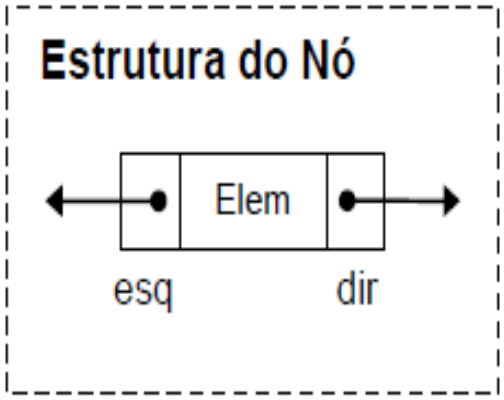
Arvore binária

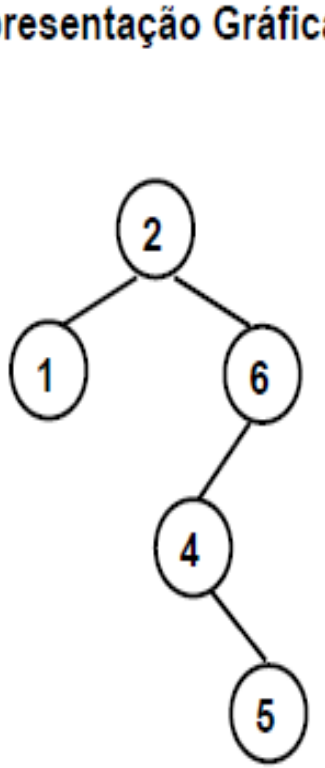


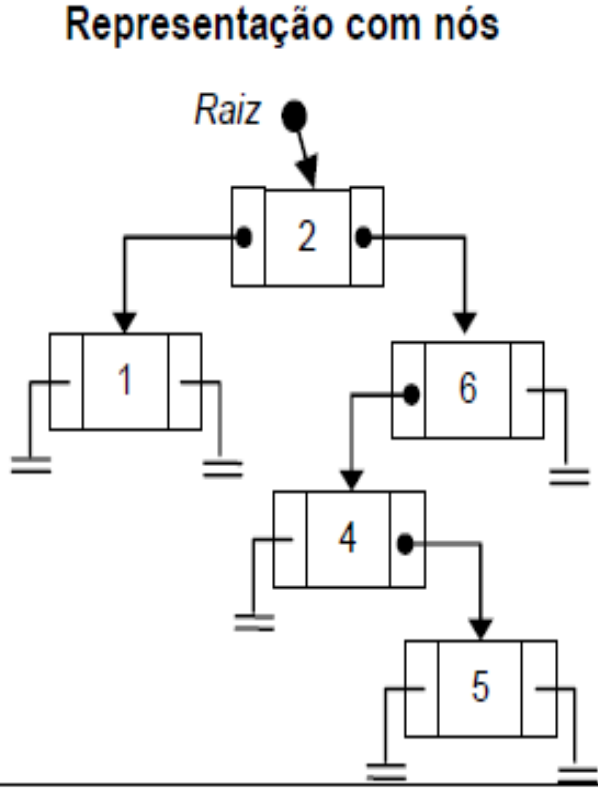
Armazenamento de uma Árvore Binária

- O armazenmento de um árvore binária surge naturalmente de sua definição
- O ponteiro ptraiz indica a raiz da árvore
- Cada nó deve possuir dois campos de ponteiros, esq e dir, que apontam para as subárvores esquerda e direita, respectivamente
- Se não houver uma subárvore, o ponteiro correspondente receberá NULL, representado nas figuras por λ

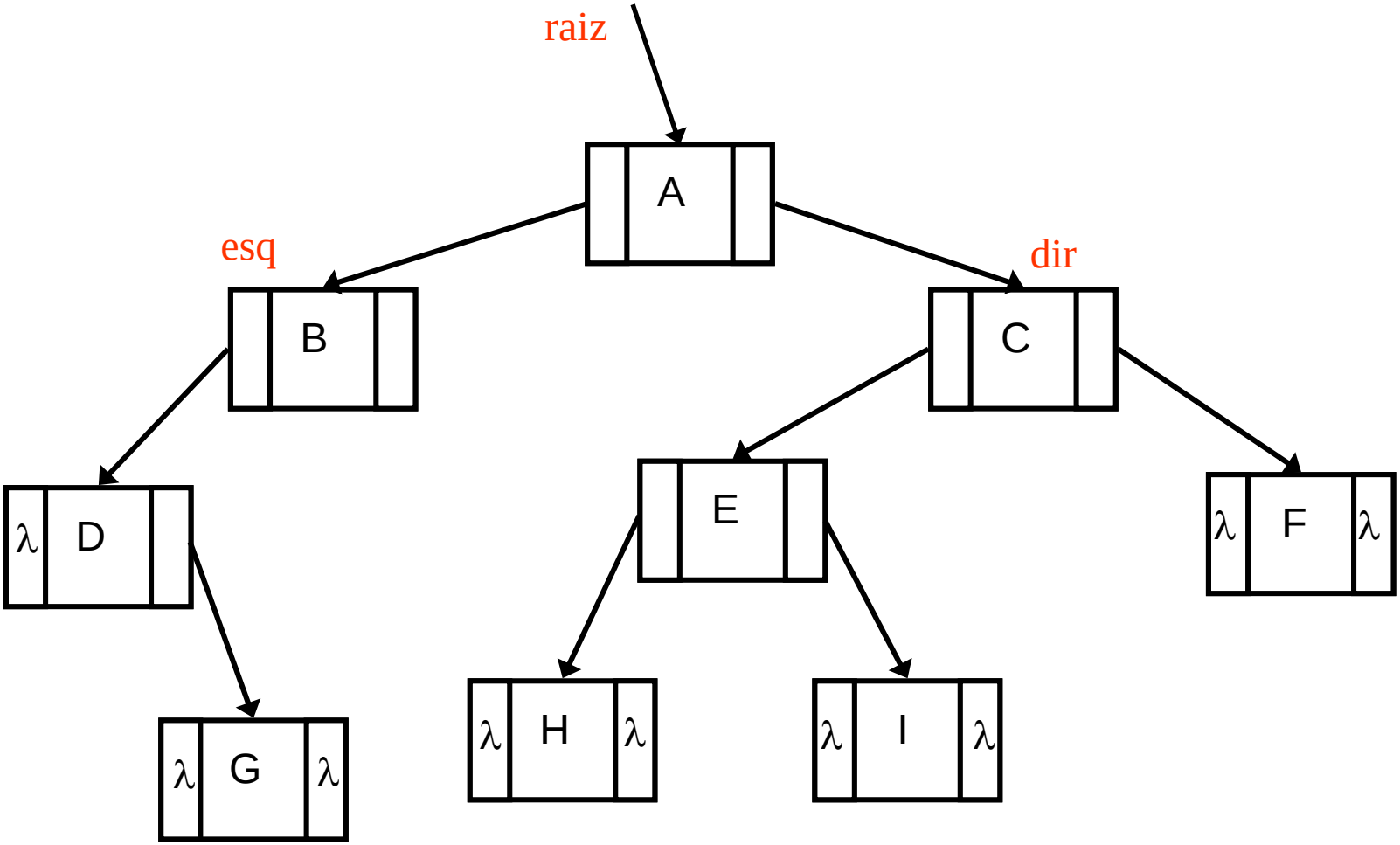
Alocação dinâmica







Armazenamento de uma Árvore Binária



Operações com árvores

- Definir uma árvore vazia
- Criar um nó raiz
- Verificar se árvore vazia ou não
- Criar um filho à direita de um dado nó
- Criar um filho à esquerda de um dado nó
- Verificar qual o nível de um dado nó
- Retornar o pai de um dado nó

Percurso em arvore binária

- Existem três formas básicas de percurso em uma árvore binária
  - Pré-ordem ou pré-fixado ou PROFUNDIDADE
  - Ordem simétrica ou central
  - Pós-ordem ou pós-fixado

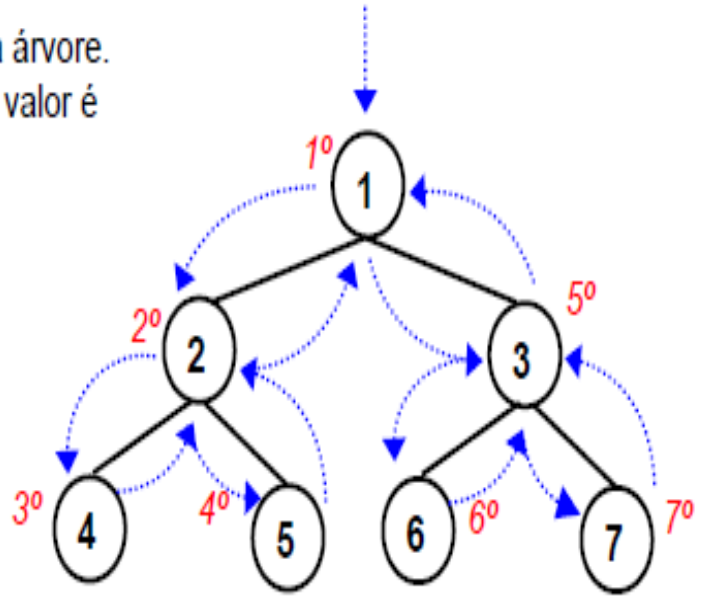
  - TE e TD são a subárvores esquerda e a direita
  - Cada subárvore é também uma árvore binária

Percurso em arvores binarias

- Pré-ordem ou pré-fixado / PROFUNDIDADE
  - Visitar a raiz
  - Percorrer sua subárvore esquerda em pré-ordem, visita nó esquerda
  - Percorrer sua subárvore direita em pré-ordem, visita nó direita

- No exemplo: A B D . G . . . C E H . . I . . F . .
- Ou simplesmente: A B D G C E H I F

Inicia o percurso pela raiz da árvore. Assim que o nó é visitado, o valor é mostrado (1ª passagem).

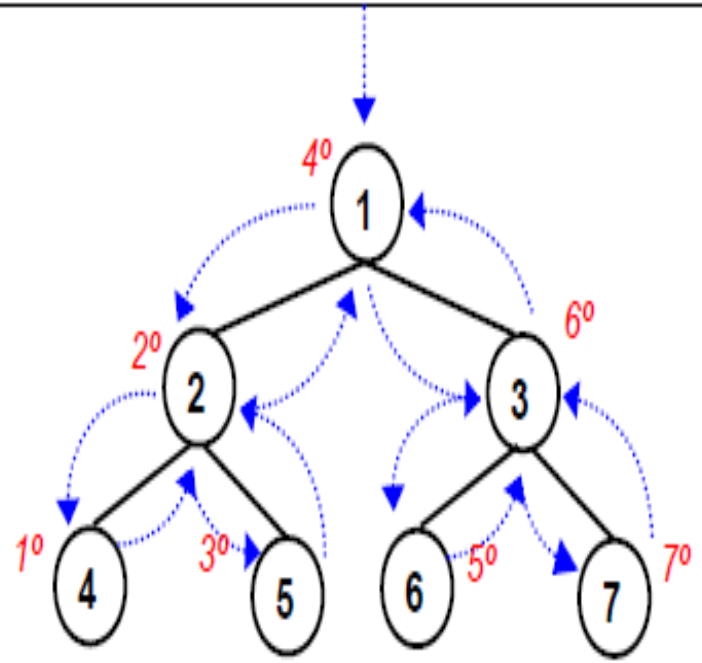


Resultado do Percurso: 1,2,4,5,3,6,7

- Pós-ordem ou pós-fixado ou ORDEM SIMÉTRICA
  - Visita o nó esquerdo;
  - 2. Mostra o valor do nó;
  - 3. Visita o nó direito;

- No exemplo: . . . G D . B . . H . . I E . . F C A
- Ou simplesmente: G D B H I E F C A

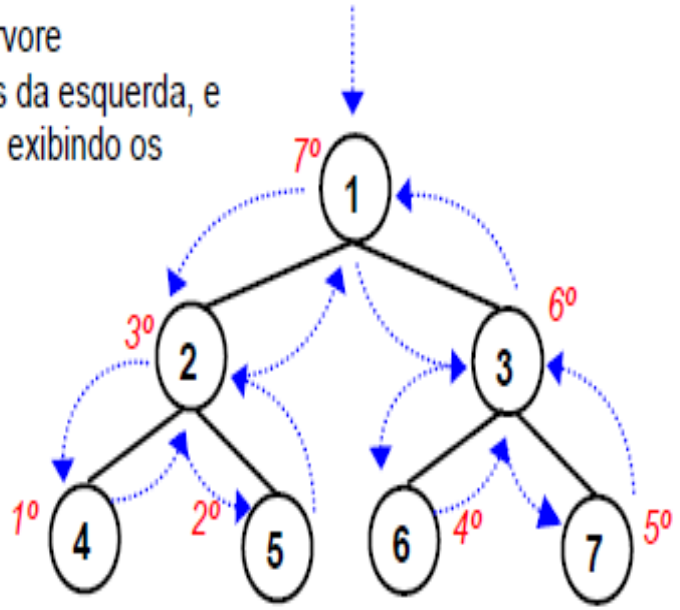
Inicia o percurso pela raiz da árvore.
Caminha inicialmente pelos nós da esquerda, só exibindo os valores quando todos à esquerda já tiverem sido visitados (2ª passagem).



Resultado do Percurso: 4,2,5,1,6,3,7

- Percurso em Pós-ordem
  - Visita o nó esquerdo;
  - 2. Mostra o valor do nó;
  - 3. Visita o nó direito;

Inicia o percurso pela raiz da árvore
Caminha inicialmente pelos nós da esquerda, e em seguida pelos da direita, só exibindo os valores quando todos os nós descendentes já tiverem sido visitados (3ª passagem).



Resultado do Percurso: 4,5,2,6,7,3,1

Implementação em arvore binária

- Definição de nós em árvores binárias

```
// estrutura tipo no com três
//campos: esq, info e dir
typedef struct
 {
   void *esq;
   char info[25];
   void *dir;
 } no;

// arvore binária
no *ptarvore;
```

```
typedef struct No pno;
struct No {
// conteúdo a ser armazenado
   no nó
int elem;
// auto-referências para os nós
   da esquerda e direita da
   árvore
pno *esq;
pno *dir;
}
```

Pré-ordem

```
int construcao(no **ptarv)
{
   char info[25];
   printf("\nDigite a Informacao: ");  // inicializar no
   scanf("%s",info);

   if(strcmp(info,"."))
   {
      // criacao do no raiz - arvore vazia
      (*ptarv)=(no *)malloc(sizeof(no));
      strcpy((*ptarv)->info,info);
      printf("\nESQ");
      construcao(&(*ptarv)->esq);
      printf("\nDIR");
      construcao(&(*ptarv)->dir);
   }else
      (*ptarv)=NULL;
   return 0;
}
```

Percurso em pre-ordem

```
int pre_ordem(no **ptarv)
{
   if(*ptarv!=NULL)
   {
       printf("\nInfo: %s",(*ptarv)->info);
       printf("\n");
       pre_ordem(&(*ptarv)->esq);
       pre_ordem(&(*ptarv)->dir);
   }
   return 0;
}
```

Percurso em ordem simétrica

```
int simetrica(no **ptarv)
{
   if(*ptarv!=NULL)
   {
       simetrica(&(*ptarv)->esq);
       printf("\nInfo: %s",(*ptarv)->info);
       printf("\n");
       simetrica(&(*ptarv)->dir);
   }
   return 0;
}
```

Percurso em pós-ordem

```
int pos_ordem(no **ptarv)
{
   if(*ptarv!=NULL)
   {
       pos_ordem(&(*ptarv)->esq);
       pos_ordem(&(*ptarv)->dir);
       printf("\nInfo: %s",(*ptarv)->info);
       printf("\n");
   }
   return 0;
}
```

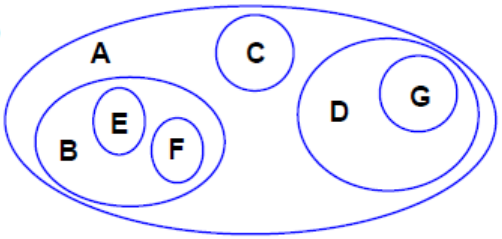
Destruição em pré-ordem

```
int destruicao(no **ptarv)
{
   if(*ptarv!=NULL)
   {
       destruicao(&(*ptarv)->esq);
       destruicao(&(*ptarv)->dir);
       free(*ptarv);
       *ptarv=NULL;
   }
   return 0;
}
```

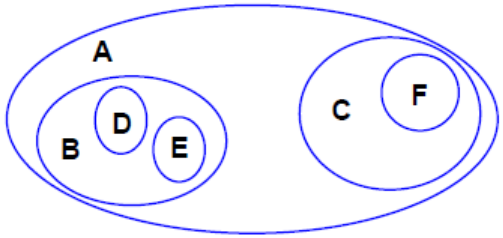
Exercícios

- Dado os diagramas abaixo, construa a representação em forma de árvore

a)



b)



- Para cada árvore, responda:
  - i. Grau dos nós;
  - ii. Grau da árvore;
  - iii. Folhas da árvore;
  - iv. Raiz da árvore;
  - v. Nós em cada nível;
  - vi. Altura da árvore;
  - b. É uma árvore binária?
  - Qual a altura máxima de uma Árvore Binária com ***n nós?***
  - Qual a altura mínima de uma Árvore Binária com ***n nós?***

Exercícios

- Implementar:
  - Percurso em pré-ordem ou Profundidade
  - Percurso em Ordem ou Ordem Simétrica
  - Percurso em Pós-Ordem