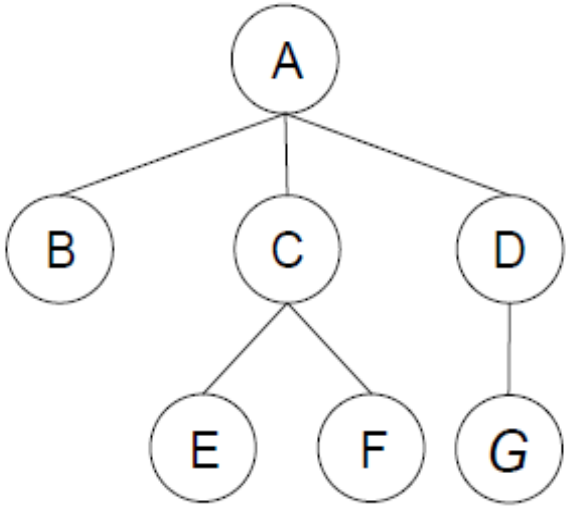
ED
Árvore
Árvore Binária

Profa. Célia Taniwaki

Árvore

- Uma árvore é uma estrutura que mantém uma relação de hierarquia ou composição entre os dados



Definição de Árvore

- Árvore T – conjunto finito de elementos, (**nós ou vértices), tais que:**

  Se **T = Ø, a árvore é dita vazia;**

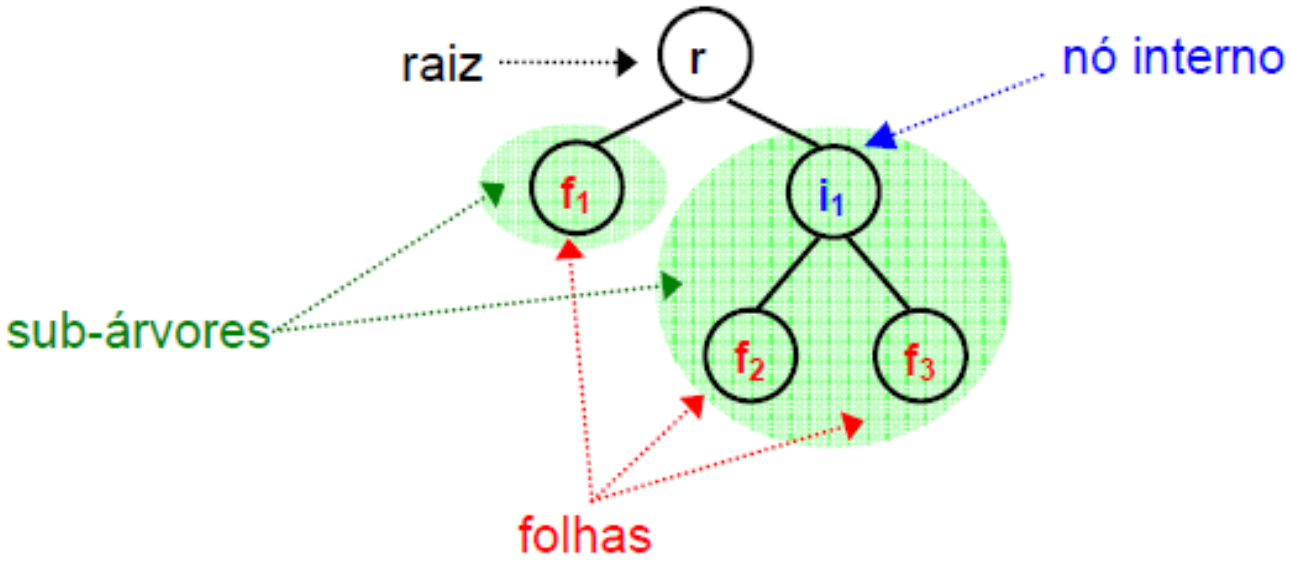
  Caso contrário:

  - **T contém um nó especial, denominado raiz;**
  - os demais nós ou constituem um único conjunto vazio, ou são divididos em **m≥*1 conjuntos disjuntos não vazios (T1,T2,…,Tn), que são, por sua vez, cada*** qual uma árvore;

    T1,T2,…,Tn são chamadas sub-árvores de T;

    Um nó sem sub-árvores é **nó-folha, ou folha.**

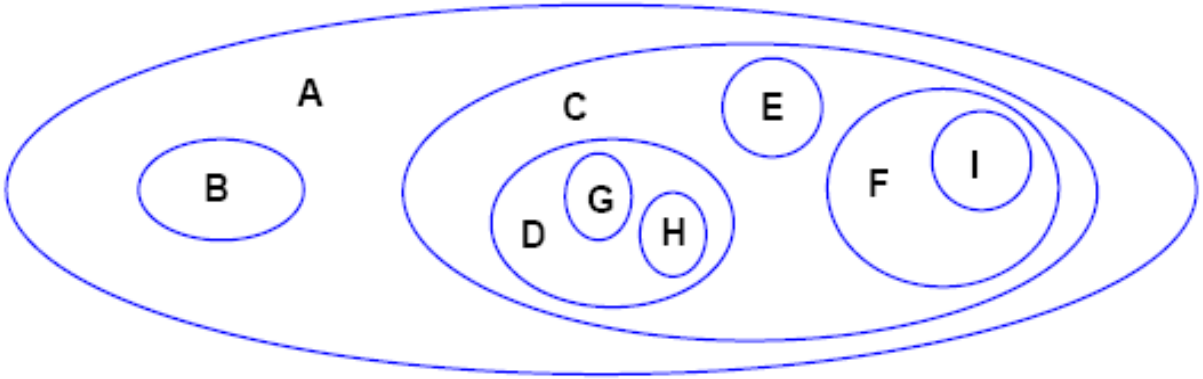
Árvore



Árvore
Outras Representações

- Outras formas de se representar uma árvore:
  - Parênteses aninhados:

    (A (B) (C (D (G) (H)) (E) (F (I))))
  - Diagrama de Venn:



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).**

PROFUNDIDADE:

nível máximo de qualquer nó folha na árvore, ou número de nós do maior percurso do nó raiz até qualquer nó folha.

GRAU:

o grau de um nó T de uma árvore = 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;

Conceitos

ALTURA:

A altura de um nó v é o número de nós no maior caminho de v até um de seus descendentes.

Os nós folha sempre têm altura igual a 0;

ALTURA DA ÁRVORE:

a altura de uma árvore T é dada pela altura máxima de seus nós.

Denota-se:

a altura de uma árvore com raiz dada pelo nó T por h(T),

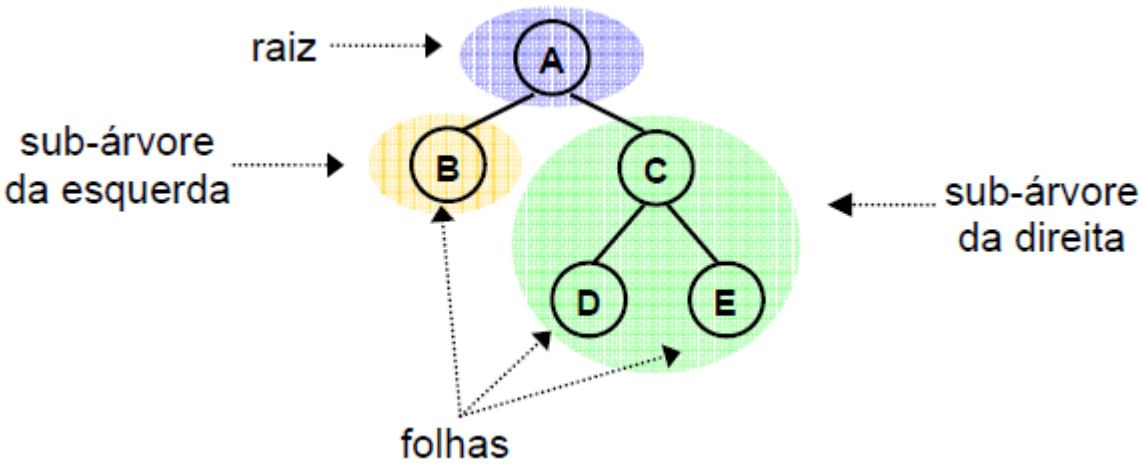
a altura de uma sub-árvore com raiz T1 por h(T1).

Mais Conceitos

- Além disso, mediante a representação hierárquica, pode-se definir que:
  - Um nó X é um ANCESTRAL do nó Y (e Y é **DESCENDENTE** de X) se X for o PAI de Y ou então se X for o PAI de algum ANCESTRAL de Y;
- Dois nós são IRMÃOS se forem filhos do mesmo pai.
- Por fim, uma FLORESTA é o conjunto de Árvores disjuntas.

Árvores Binárias

- 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 esquerda;**
  - Uma **árvore binária chamada de sub-árvore da direita.**



Propriedades

- Uma árvore binária possui algumas propriedades:
  - O número máximo de nós no nível ***i de uma árvore binária é*** 2^i***.***
  - O número máximo de nós em uma árvore binária de altura ***k é*** $2^{k+1}-1$***.***

Caminhamento

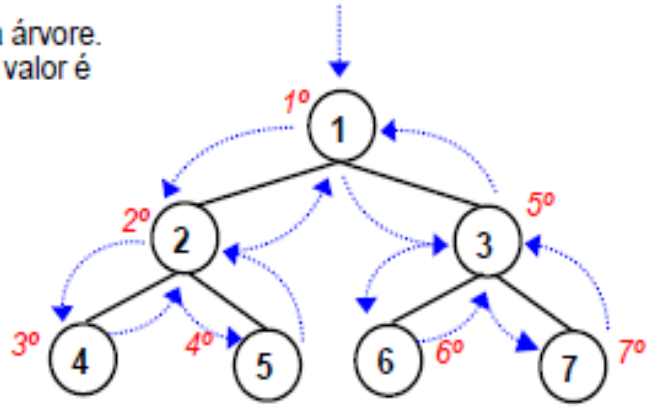
- Diversas formas de percorrer ou caminhar em uma árvore listando seus nós, as principais:
  - Pré-ordem (Pré-fixa)
  - Em-ordem (Infixa)
  - Pós-ordem (Pós-fixa)

- Para todas elas:
  - Se T é uma árvore nula, então a lista é nula.
  - Se T é uma árvore de um único nó então a lista contém apenas este nó.
  - O que muda é a ordem de apresentação da raiz.

Pré-ordem ou Pré-fixa ou Profundidade

- Pré-ordem:
  - Mostra o valor do nó
  - Visita o nó (filho) esquerdo
  - Visita o nó (filho) direito

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

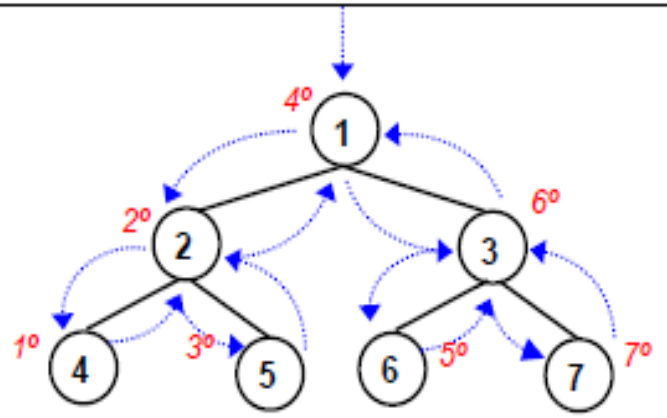


Resultado do Percurso: 1,2,4,5,3,6,7

Em-ordem ou Infixa ou Ordem Simétrica

- Em-ordem:
  - Visita o nó (filho) esquerdo
  - Mostra o valor do nó
  - Visita o nó (filho) direito

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).

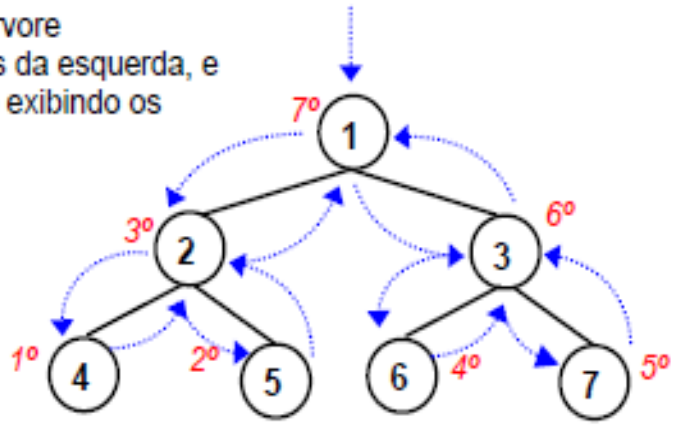


Resultado do Percurso: 4,2,5,1,6,3,7

Pós-ordem ou Pós-fixa

- Pós-ordem:
  - Visita o nó (filho) esquerdo
  - Visita o nó (filho) direito
  - Mostra o valor do nó

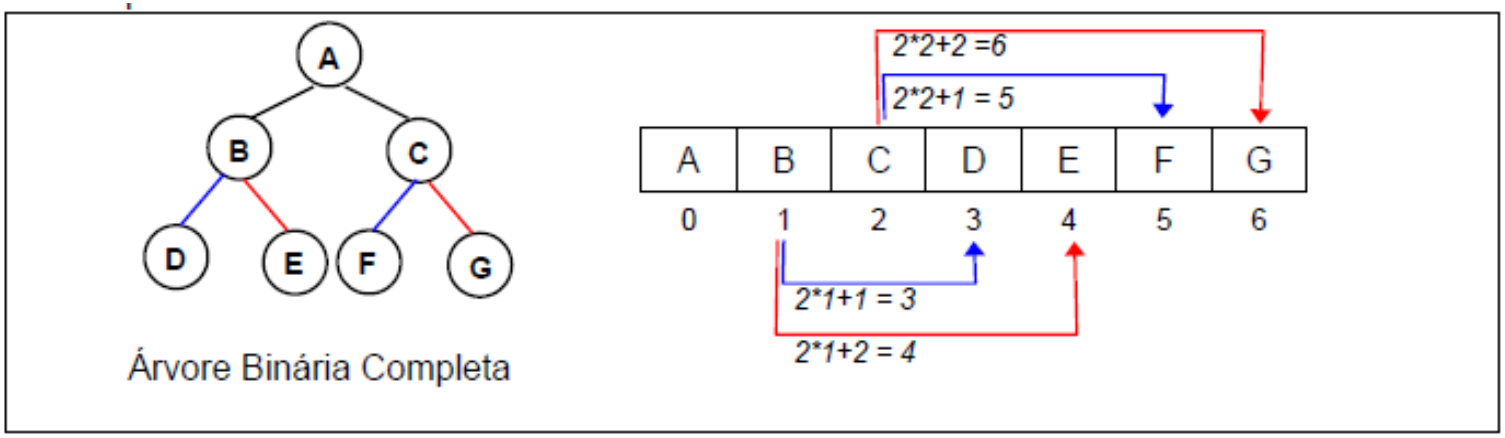
Inicia o percurso pela raiz da árvore
Caminha inicialmente pelos nós da esquerda, e em seguida pelos da direita, só exibindo os valores quando todos os nós descendentes já tiverem sido visitados (3ª passagem).



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

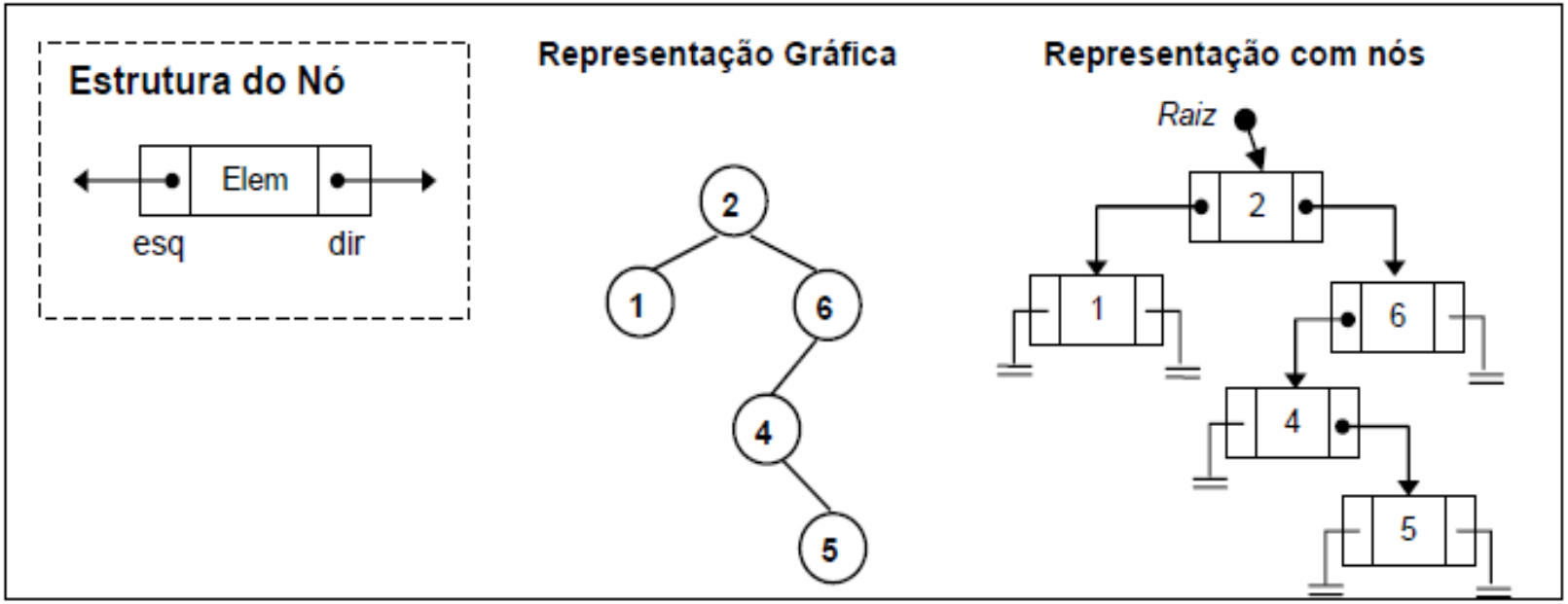
Implementação: Lista Sequencial

- Através de lista sequencial (vetor):
  - Armazenar os nós, por níveis, em um vetor
  - Se um nó ocupa a posição i de um vetor, seus filhos estão nas posições 2i+1 e 2i+2
  - Vantagem: espaço necessário apenas para o conteúdo (ligações estão implícitas)
  - Desvantagem: se a árvore não é completa, ocupa muito espaço desnecessário



Implementação: Estrutura dinâmica

- Cada nó contém:
  - o seu valor,
  - uma referência para o filho esquerdo
  - uma referência para o filho direito.



Exercícios

1. Implementar a classe para representar um nó:

   Classe Node

   Atributos:

   – elemento (valor do nó)
   – esq (do tipo Node, referencia o filho esquerdo)
   – dir (do tipo Node, referencia o filho direito)

   Construtor: Node (elemento)

   – Atribui o valor do elemento
   – Atribui null para esq e dir

   Propriedades (ou setters e getters)

Exercícios

2. Implementar a classe para representar uma árvore binária:

Classe ArvoreBin

Atributos:

– raiz (de tipo Node)

Construtor: ArvoreBin ( )

– Atribui null a raiz

Método GetRaiz - devolve raiz

Método CriaRaiz (valor)

Cria um objeto Node com valor e atribui esse objeto a raiz

Método InsereDir (Node pai, valor)

Verifica se a árvore não é vazia e se pai não tem filho direito

Cria objeto Node com valor e insere como filho direito de pai

Retorna referência do novo nó.

Exercícios

Método InsereEsq (Node pai, valor)

Verifica se a árvore não é vazia e se pai não tem filho esq.

Cria objeto No com valor e insere como filho esquerdo de pai

Retorna referência do novo nó.

3. Implementar na classe Program, no método Main, o teste da Classe ArvoreBin. Crie a árvore a seguir: