

---

# Estrutura de Dados

## Árvore - Conceitos

— Profª. Ana Cristina dos Santos —

email: [ana.csantos@sp.senac.br](mailto:ana.csantos@sp.senac.br)

---

# Conteúdo

- Árvore e suas terminologias
- Árvore Binária e suas terminologias
- Percursos em árvores
- Árvore Binária de Busca
- Operações básicas: busca, inserção e remoção

# Árvores

- Agora que estudamos **estruturas de dados lineares** como pilhas e filas e temos alguma experiência com recursão, veremos uma estrutura de dados comum chamada de árvore.
- As árvores são usadas em muitas áreas da ciência da computação, incluindo sistemas operacionais, gráficos, sistemas de bancos de dados e redes de computadores.

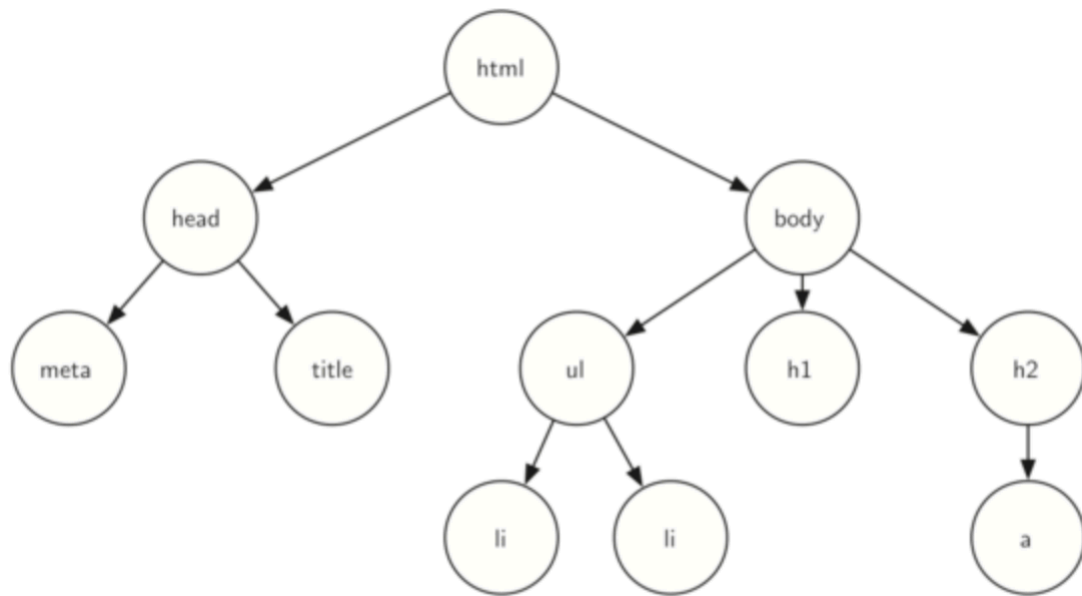
# Árvores

- Estruturas de dados de árvores têm muitas coisas em comum com seus primos botânicos. Uma estrutura de dados em árvore tem uma raiz, ramificações e folhas.
- A diferença entre uma árvore na natureza e uma árvore na ciência da computação é que a estrutura de dados de uma árvore tem sua raiz no topo e suas folhas na parte inferior.

# Árvores: exemplo

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xml:lang="en" lang="en">
<head>
  <meta http-equiv="Content-Type"
        content="text/html; charset=utf-8" />
  <title>simple</title>
</head>
<body>
<h1>A simple web page</h1>
<ul>
  <li>List item one</li>
  <li>List item two</li>
</ul>
<h2><a href="http://www.cs.luther.edu">Luther CS </a></h2>
</body>
</html>
```

# Árvores: exemplo



# Árvores

- Uma **Árvore** é uma estrutura complexa, que modela inúmeros problemas práticos.
- Uma árvore (tree) pode ser implementada como um tipo abstrato de dados - TAD que armazena elementos de **maneira hierárquica**.

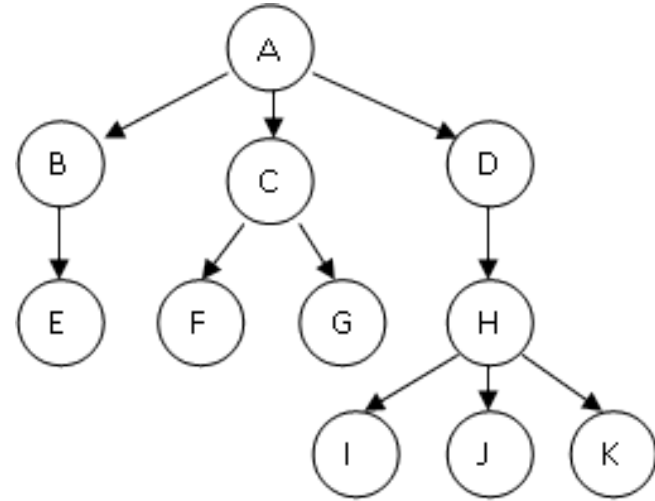
# Árvores

- A **árvore** é o exemplo mais básico de estruturas de **dados hierárquicas**
  - Nesse tipo de estrutura a relação entre os dados é formada pela subordinação de nós
  - O formalismo no conceito das árvores define que uma árvore  $T$  (de Tree) é um conjunto finito de nós, onde:
    - Existe um nó denominado “**raiz da árvore**”;
    - Os demais nós da árvore são divididos em  $n \geq 0$  subconjuntos que formam as chamadas subárvores.



# Árvores

- No exemplo ao lado, os círculos representam os nós da árvore, sendo que o **nó A** representa a **raiz da árvore**:



# Árvores

- Terminologia de Árvores
  - Na figura anterior cada **nó da árvore** representa a **raiz de uma subárvore**.
  - Porém, além da terminologia “raiz” a árvore é composta de:
    - **Nó pai:** nó ao qual um nó está ligado (diretamente);
    - **Nó filho:** cada um dos nós subordinados ao nó pai;
    - **Nó ancestral:** todos os nós acima de um dado nó (em direção a raiz);
    - **Nó descendente:** todos os nós abaixo de um dado nó;

# Árvores

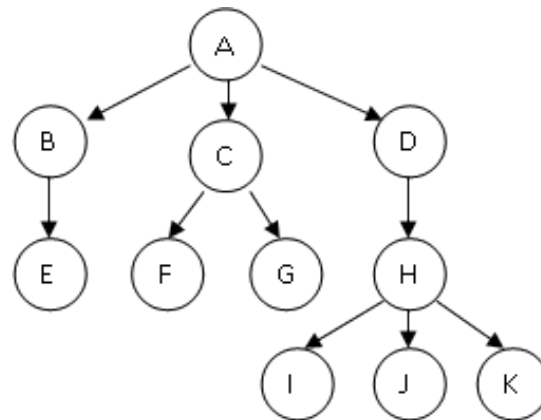
- Terminologia de Árvores
  - Na figura anterior cada nó da árvore representa a raiz de uma subárvore.
  - Porém, além da terminologia “raiz” a árvore é composta de:
    - **Nós irmãos:** todos os nós que tem o mesmo pai;
    - **Grau:** número de subárvores (ou de nós filhos) de um nó;
    - **Folha (Nó Terminal):** nó sem filho ou com grau zero;
    - **Nível:** número de nós entre um nó e a raiz;

# Árvores

- Terminologia de Árvores
  - Na figura anterior cada nó da árvore é representa a raiz de uma subárvore. Porém, além da terminologia “raiz” a árvore é composta de:
    - **Altura da árvore (Profundidade):** nível mais alto;
    - **Grau da árvore:** o mesmo valor do nó com a maior quantidade de filhos (maior grau da árvore);
    - **Floresta:** conjunto de árvores disjuntas.
      - Eliminando a raiz → floresta
      - Criando uma nova raiz → árvore

# Árvores

- Terminologia de Árvores
  - De acordo com os dados ilustrados na figura anterior, podemos montar uma tabela com os dados da terminologia de árvores, veja:



- **O nível da árvore e o grau da árvore é 3.**

Nó	Grau	Nível	Filhos	Irmãos	Observações
A	3	0	B, C, D		Raiz
B	1	1	E	C, D	
C	2	1	F, G	B, D	
D	1	1	H	B, C	
E	0	2			Folha
F	0	2		G	Folha
G	0	2		F	Folha
H	3	2	I, J, K		
I	0	3		J, K	Folha
J	0	3		I, K	Folha
K	0	3		I, J	Folha

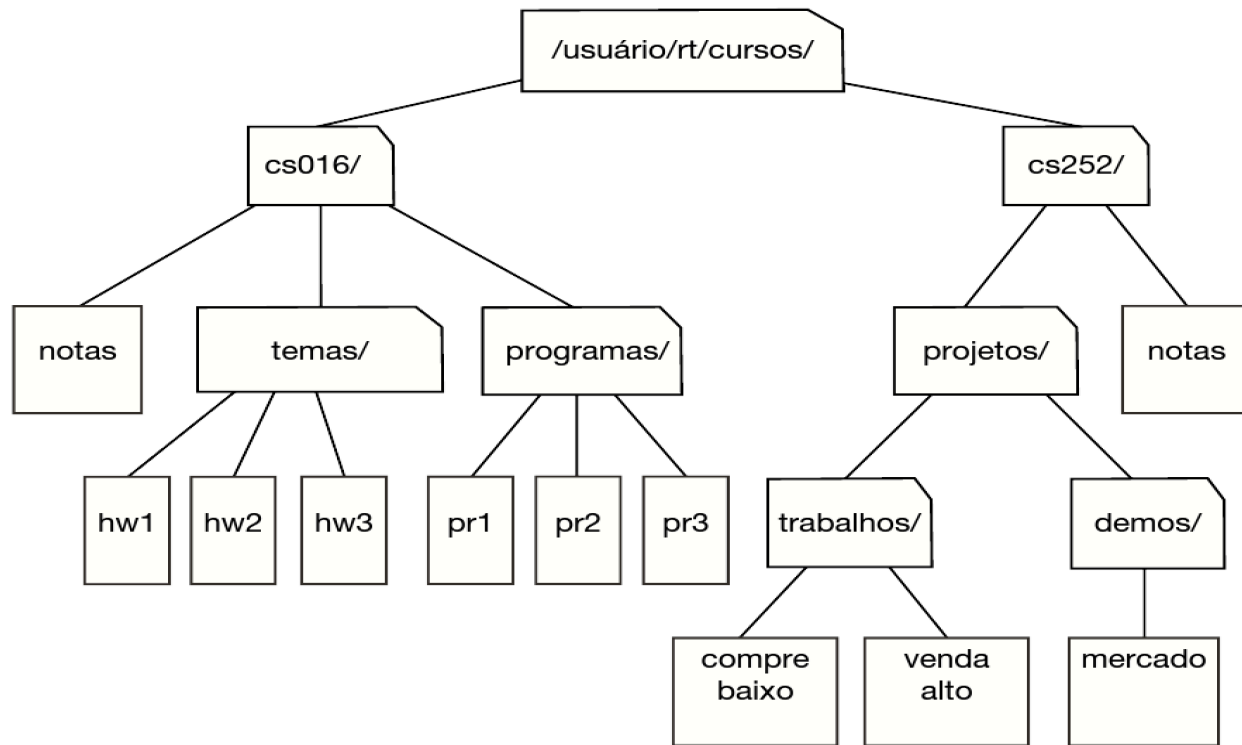
# Árvores

- Terminologia de Árvores

- Algumas observações:

- A **raiz da árvore** tem nível 0 e o nível dos demais nós é igual ao número de segmentos que o liga à raiz;
    - A **altura** (ou profundidade) **da árvore** é definida pelo nível mais alto da árvore;
    - Ao **eliminarmos** a raiz de uma árvore, o que restar dessa operação forma uma floresta;
    - Todos os **nós da árvore**, com exceção do nó raiz, têm como ancestral a raiz da árvore;

# Árvores: exemplo



# Árvores: exemplo

- Na árvore mostrada:
  - Identifique a raiz, as folhas e os nós internos.
  - Mostre quais são os filhos do nó programas/
  - Identifique quem é o pai de mercado/
  - notas/ e programas/ são irmãos ?
  - Quais são os níveis dos nós cs252/ e mercado/ ?
  - Qual é a altura desta árvore ?



# Árvores

- Representações de Árvores

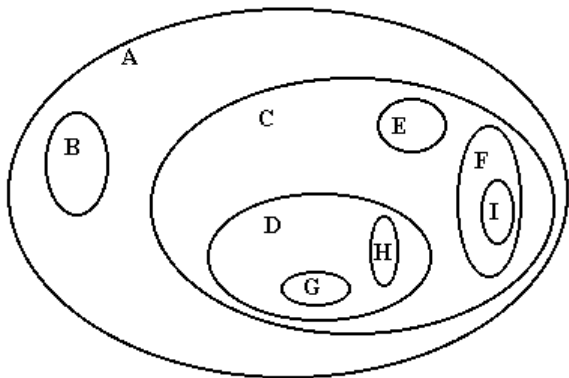
- As três formas clássicas para representar graficamente uma árvore são ilustradas nesse tópico:
  - Parênteses Aninhados
  - Diagrama de Inclusão
  - Hierárquica
- Veja alguns exemplos nos próximos *slides*

# Árvores

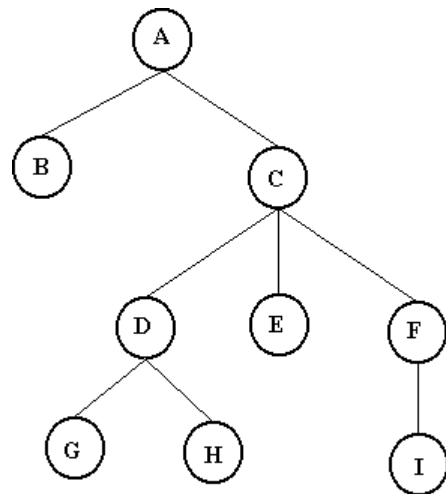
## Parênteses Aninhados

( A ( B ) ( C ( D ( G ) ( H ) ) ( E ) ( F ( I ) ) ) )

## Diagrama de Inclusão



## Hierárquica

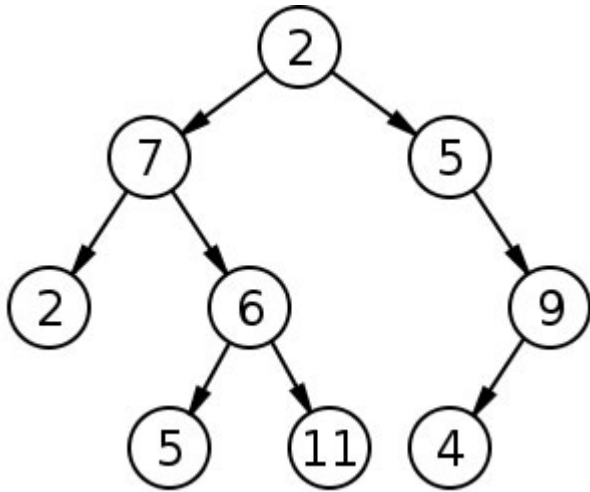


# Árvores Binárias

- Uma árvore T é classificada como **Árvore Binária** se cada um dos seus nós for de grau igual a: 0, 1 ou 2
  - Como cada nó da árvore pode ter no máximo dois nós filhos, eles são denominados como:
    - nó filho esquerdo; e
    - nó filho direito

# Árvores Binárias

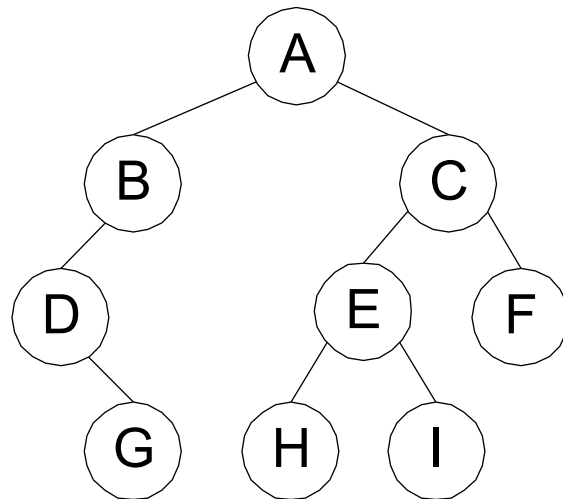
- Um dos modos mais clássicos para representar uma árvore binária é ilustrado na figura abaixo



- A árvore binária possui 9 nós e:
  - O nó 2 é a raiz da árvore;
  - O nó 7 é a raiz da subárvore da esquerda do nó 2;
  - O nó 5 é a raiz da subárvore da direita do nó 2.
  - Os nós 2, 5, 11, 4 são as folhas da árvore binária.
- Observe que é possível armazenar valores idênticos em nós distintos da árvore

# Árvores Binárias : Definição

- A Árvore Binária acima:
- O nó **D** é **descendente** de **A**.
- O nível de **F** é 2 e sua altura é 2,
- O nível de **A** é 0 e sua altura é 0.
- A altura da árvore é 3.

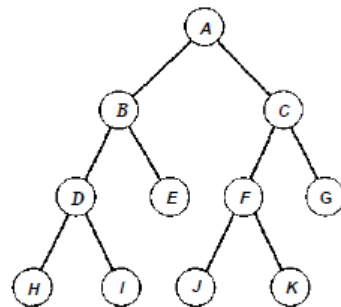
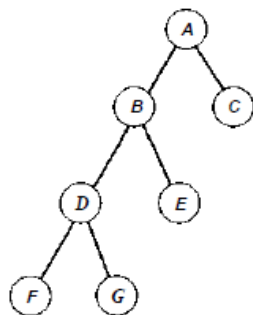
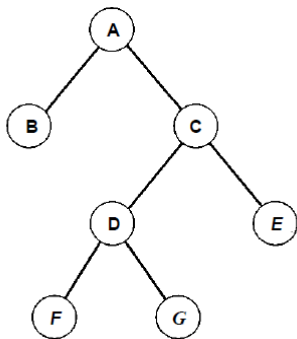


# Variações de Árvores Binárias

- Tipos de Árvores Binárias
  - Apesar de ser um tópico simples, cada autor na computação dá uma nomenclatura diferente para as árvores binárias
  - Os três tipos de árvores binárias que vamos estudar são:
    - **Árvore Estritamente Binária**
    - **Árvore Binária Quase Completa**
    - **Árvore Binária Completa**

# Variações de Árvores Binárias

- Tipos de Árvores Binárias
  - **Árvore Estritamente Binária** é aquela onde todos os nós (numa árvore binária) que não são folhas tiver subárvores esquerda e direita (não vazias). Ou ainda, cada nó possui 0 ou 2 filhos.



# Variações de Árvores Binárias

- Tipos de Árvores Binárias
  - Uma árvore binária de profundidade  $d$  é tipada como **Árvore Binária Quase Completa** se:
    - Cada folha da árvore estiver no nível  $d$  ou no nível  $d-1$ ;
    - Para cada nó  $n \downarrow d$  na árvore com um descendente direto no nível  $d$ , todos os descendentes esquerdos de  $n \downarrow d$  que forem folhas estiverem também no nível  $d$ .



# Variações de Árvores Binárias

- Tipos de Árvores Binárias

- Como descobrir se uma árvore binária é uma **Árvore Binária Quase Completa**?

- Podemos numerar os nós de árvores binárias completas seguindo as seguintes regras:

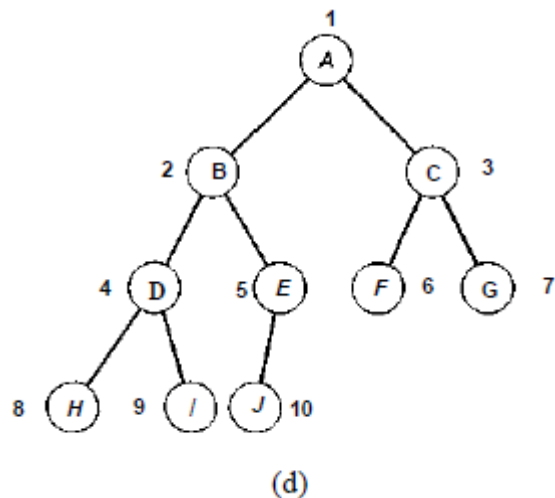
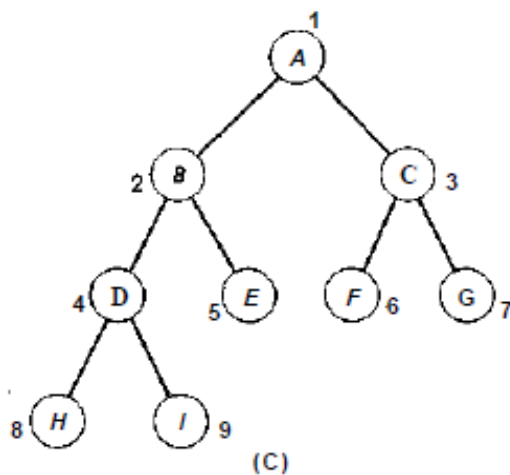
- Atribui-se o número 1 à raiz da árvore;
      - Os nós filhos da esquerda recebem um número equivalente ao dobro do seu nó pai ( $2n$ );
      - Os nós filhos da direita recebem um número equivalente ao dobro mais um do seu nó pai ( $2n + 1$ ).
      - **Ao final, se não estiver faltando nenhum número a árvore é quase completa.**

# Variações de Árvores Binárias

- Tipos de Árvores Binárias
  - **Árvore Binária Quase Completa**
    - Apresenta a seguinte propriedade: se  $r$  é um nó tal que alguma subárvore de  $r$  é vazia, então  $r$  se localiza ou está no último (maior) ou nó penúltimo nível da árvore.

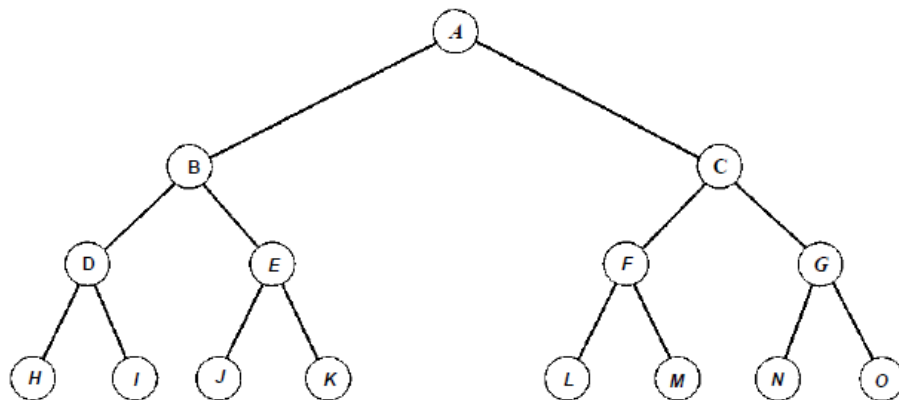
# Variações de Árvores Binárias

- Tipos de Árvores Binárias
  - Exemplos de **Árvores Binárias Quase Completas**



# Variações de Árvores Binárias

- Tipos de Árvores Binárias
  - **Árvore Binária Completa** é aquela que também é **estritamente binária** e todos os nós folhas estão no nível  $d$  da árvore



# Variações de Árvores Binárias

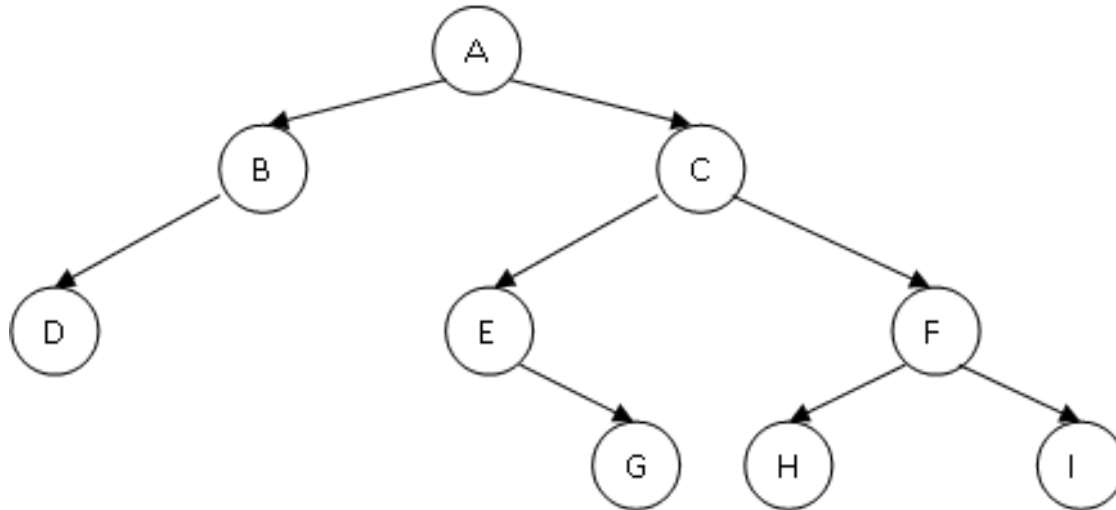
- Exercícios:
  - Desenhe uma árvore estritamente binária com 7 nós
  - Desenhe uma árvore binária completa com 9 nós

# Árvores Binárias

- Percursos em Árvores Binárias
  - O **percurso** é um processo que visita todos os nós de uma árvore binária uma única vez
  - Esse processo gera uma sequência linear de nós
    - Algo muito parecido como colocar os nós de uma árvore em uma única linha
  - Existem três maneiras recursivas de se visitar os nós da árvore binária: **pré-ordem, in-ordem ou ordem simétrica e pós-ordem**

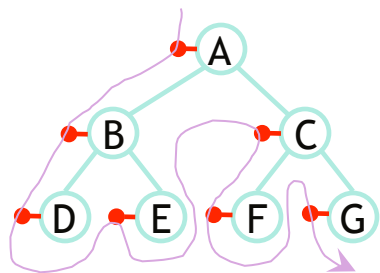
# Árvores Binárias

- Percursos em Árvores Binárias
  - Para exemplificar os 3 tipos de percursos, vamos utilizar a árvore binária abaixo:



# Árvores Binárias

- Percursos em Árvores Binárias
  - **Pré-Ordem:** é um tipo de percurso que segue as seguintes regras:
    - **V**isita o nó raiz;
    - Percorre a subárvore esquerda (pré-ordem);
    - Percorre a subárvore direita (pré-ordem).



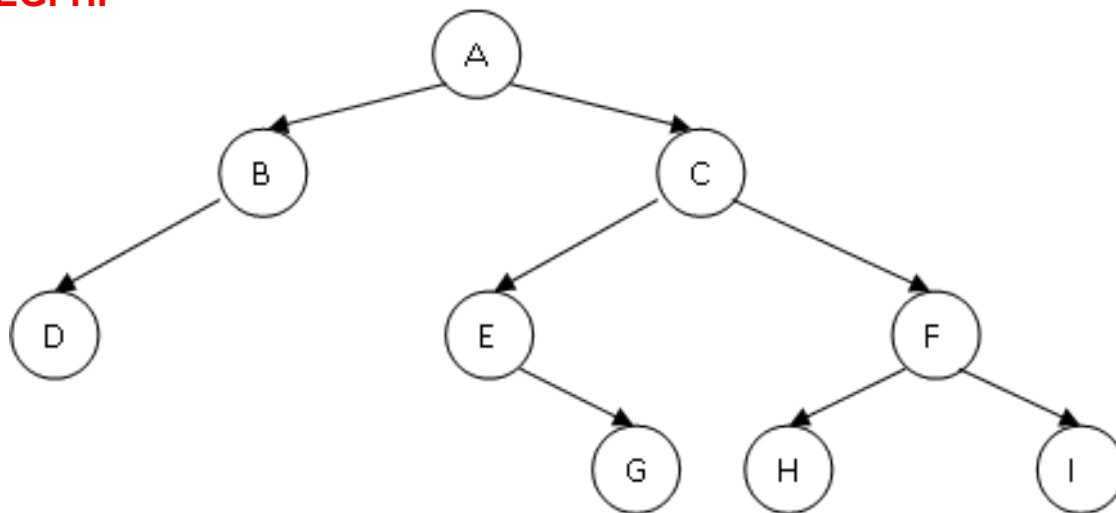
A B D E C F G





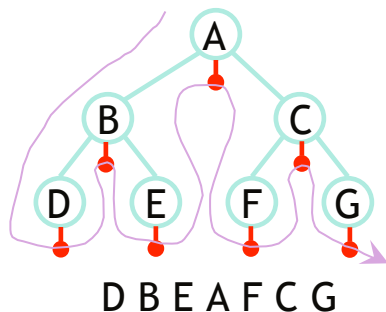
# Árvores Binárias

- Percursos em Árvores Binárias
  - O resultado do percurso dessa árvore na forma **pré-ordem** gera a seguinte saída:  
**ABDCEGFHI**



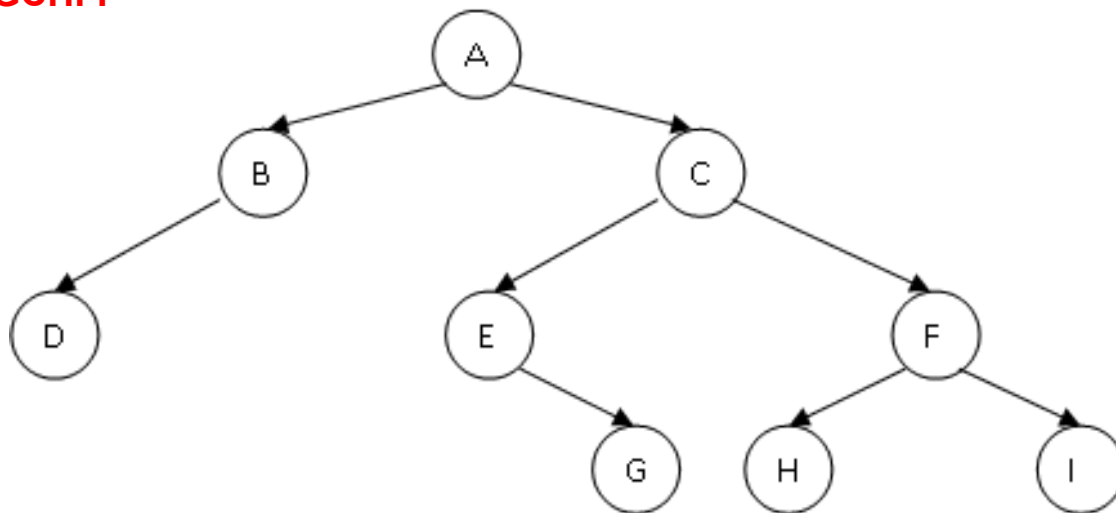
# Árvores Binárias

- Percursos em Árvores Binárias
  - **In-Ordem ou Ordem Simétrica:** é um tipo de percurso que segue as seguintes regras:
    - Percorre a subárvore esquerda (in-ordem);
    - **Visita o nó raiz;**
    - Percorre a subárvore direita (in-ordem).



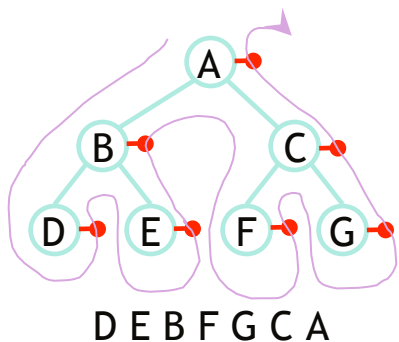
# Árvores Binárias

- Percursos em Árvores Binárias
  - O resultado do percurso dessa árvore na forma **in-ordem** gera a seguinte saída:  
**DBAEGCHF**



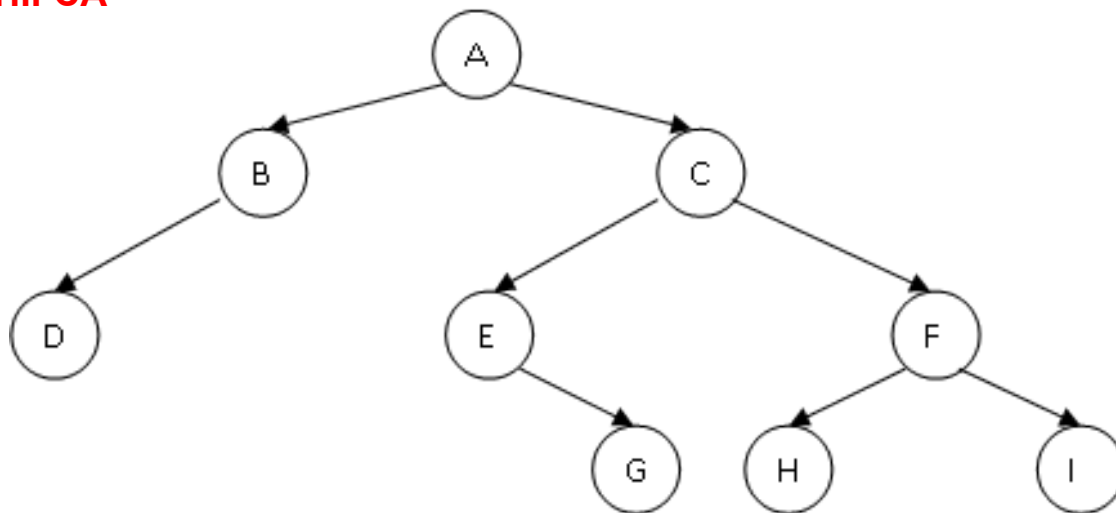
# Árvores Binárias

- Percursos em Árvores Binárias
  - **Pós-Ordem:** é um tipo de percurso que segue as seguintes regras:
    - Percorre a subárvore esquerda (pós-ordem);
    - Percorre a subárvore direita (pós-ordem);
    - **Visita** o nó raiz.



# Árvores Binárias

- Percursos em Árvores Binárias
  - O resultado do percurso dessa árvore na forma **pós-ordem** gera a seguinte saída:  
**DBGEHIFCA**



# Árvores Binárias

- Percursos em Árvores Binárias
  - Os percursos também são conhecidos como:
    - **Pré-Ordem:** Pré-fixado, varredura r-e-d e preorder traversal
    - **In-Ordem:** central, em ordem, varredura e-r-d e inorder traversal
    - **Pós-Ordem:** Pós-fixado, varredura e-d-r e postorder traversal

# Árvores Binárias

- Representação de Expressões Matemáticas
  - As árvores binárias são utilizadas para representar expressões matemática sob forma hierárquica, chamadas de **árvore de expressão**.
  - Os **operandos e operadores** são distribuídos sob os nós de uma **árvore estritamente binária**, onde:
    - A raiz da árvore contém um operador que deve ser aplicado aos resultados das subárvores esquerda e direita;
    - Os nós não folha representam os operadores;
    - Os nós folhas representam os operandos.

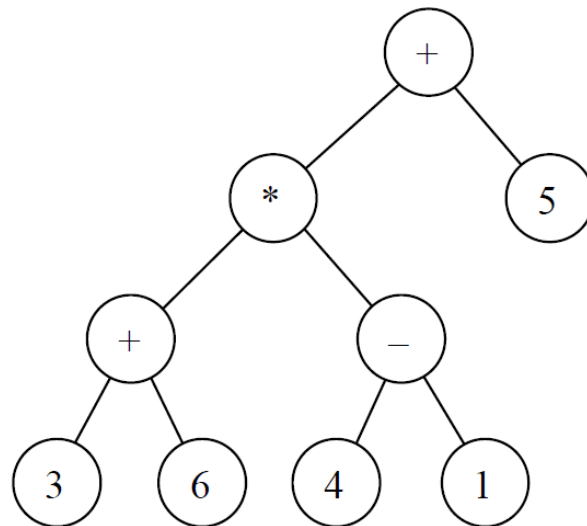
# Árvores Binárias

- Ao percorrer a árvore ao lado com o processo pré-ordem gera a seguinte expressão:

**+ \* + 3 6 - 4 1 5**

- Já no percurso pós-ordem, a saída gerada é a seguinte:

**3 6 + 4 1 - \* 5 +**

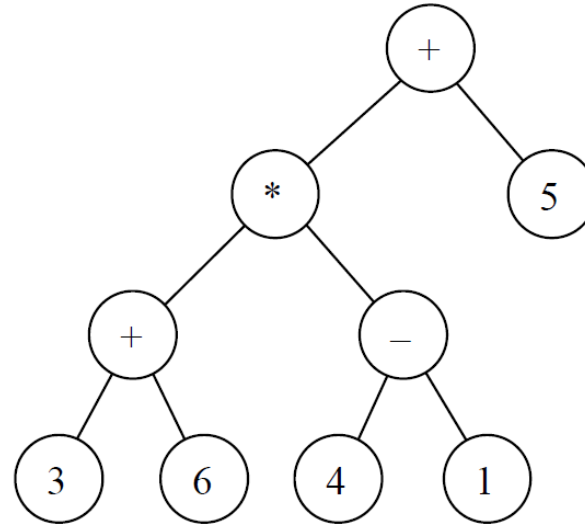




# Árvores Binárias

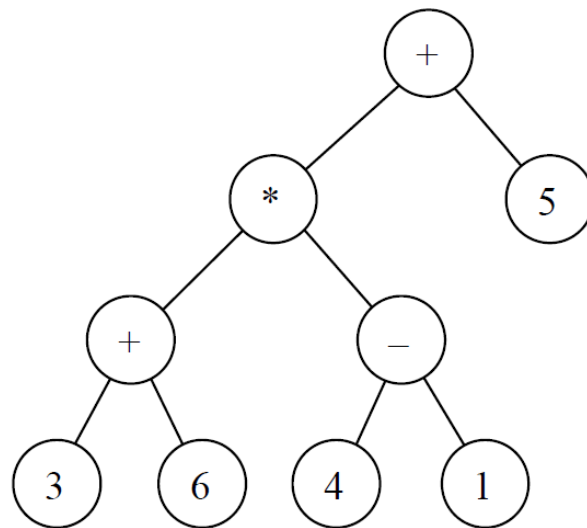
- Observem que os dois percursos anteriores não geraram expressões matemáticas válidas
- O percurso **in-ordem** é o responsável por gerar saídas válidas

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



# Árvores Binárias

- Notação Polonesa ou Posfixa:
  - Notação prefixa
  - Notação infixa
  - Notação posfixa



# Árvores Binárias de Busca

- **Árvore Binária de Busca** (= search trees) ou **Árvore Binária de Pesquisa**:
  - Todas as chaves da **sub-árvore esquerda são menores que a chave da raiz**.
  - Todas as chaves da **sub-árvore direita são maiores que a chave raiz**.
  - As sub-árvores direita e esquerda são também Árvores Binárias de Busca.
- O objetivo desta árvore é estruturar os dados de forma flexível, permitindo **pesquisa binária**.

# Árvore Binária de Busca

- **Árvore Binária de Busca**

- Para **montar** uma **árvore binária de busca**, devemos considerar as seguintes regras:
  - Todos os nós da árvore armazenam valores;
  - Todos os **descendentes esquerdos** em uma subárvore contêm valores **menores ou iguais** do que a raiz;
  - Todos os **descendentes direitos** em uma subárvore contêm valores **maiores** do que a raiz.

# Árvore Binária de Busca - ABB

- Exemplos:
  - Representar os dados a seguir em uma ABB:  
**5 8 2 12 10 14 9**
  - Representar os dados a seguir em uma ABB:  
**9 12 10 5 8 2 14**
  - Representar os dados a seguir em uma ABB:  
**12 9 14 5 10 8 2**
- Obs: Uma **ABB não é única** para um conjunto de dados, a árvore depende da ordem nas quais os elementos são inseridos na árvore.

# Árvore Binária de Busca

- É importante ressaltar que os símbolos armazenados nesse tipo de árvore podem ser **numéricos, alfabéticos ou alfanuméricos**
  - Nesse caso, **devemos considerar** a organização dos símbolos em ordem crescente
  - **Por exemplo:** Como organizar os “nomes” dos meses do ano?
    - Abril < Agosto < Dezembro < Fevereiro < Janeiro < Julho < Junho < Maio < Março < Novembro < Outubro < Setembro

# Árvore Binária de Busca

- Se for aplicado o **percurso em ordem simétrica** na árvore, será gerado uma saída com os símbolos em **ordem crescente**
- As operações mais básicas sob uma árvore binária de busca são as seguintes:
  - **Busca** por um símbolo na árvore;
  - **Inserir** um símbolo na árvore;
  - **Excluir** um símbolo da árvore.

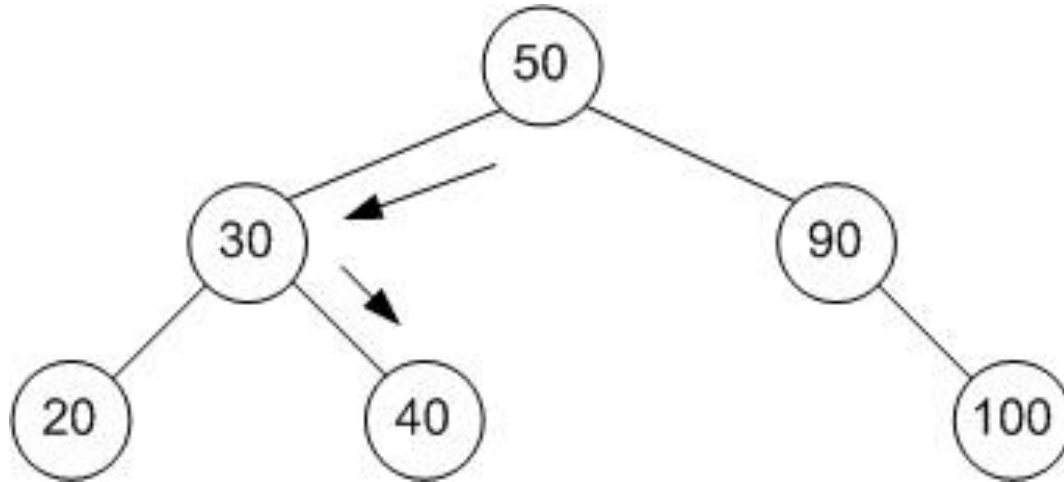
# Busca em Árvore Binária de Busca

- A **busca** por um dado em uma árvore binária de pesquisa é realizada da seguinte forma:
  - **Examina a raiz de uma subárvore**
  - **Se o valor for igual:** você achou o valor na árvore
  - **Caso contrário**
    - **Se o valor for menor do que a raiz:** continue a busca recursivamente na subárvore à esquerda
    - **Caso contrário:** continue a busca recursivamente na subárvore à direita



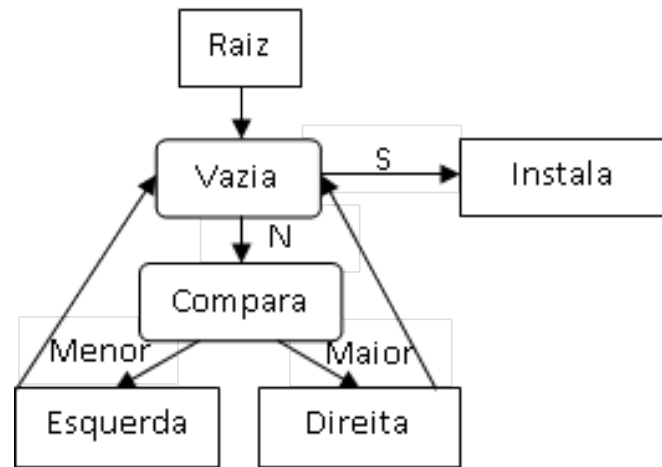
# Busca em Árvore Binária de Busca

- **Por exemplo:** Procurar o valor 40 na árvore



# Inserção em Árvore Binária de Busca

- Antes de **inserir** um símbolo, é necessário realizar o processo de busca visto no *slide* anterior
  - Caso não encontre o valor na árvore (atingiu um nó folha) insere o símbolo como filho do nó folha
  - Veja o diagrama ao lado:

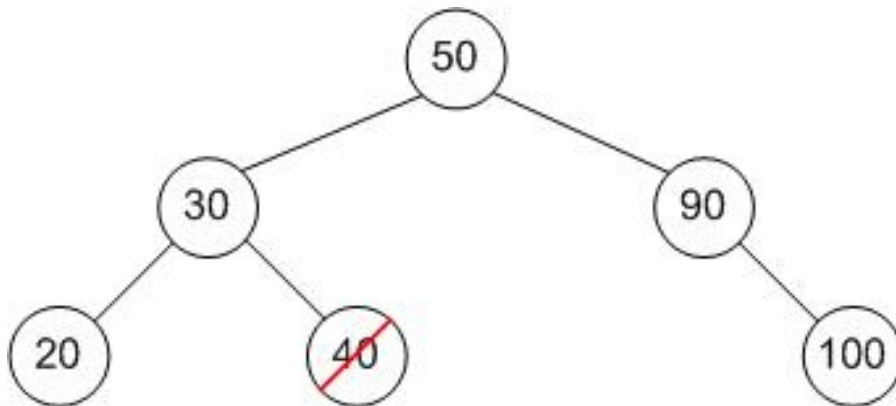


# Remoção em Árvore Binária de Busca

- A operação **mais difícil** em uma árvore binária de pesquisa é o **processo de exclusão**.
- O grande problema nesse caso é a **necessidade de reestruturar** a árvore binária em alguns casos extremos
  - Vamos considerar os **três casos** possíveis de exclusão:
    - Excluir o nó folha
    - Excluir um nó de grau 1
    - Excluir um nó de grau 2

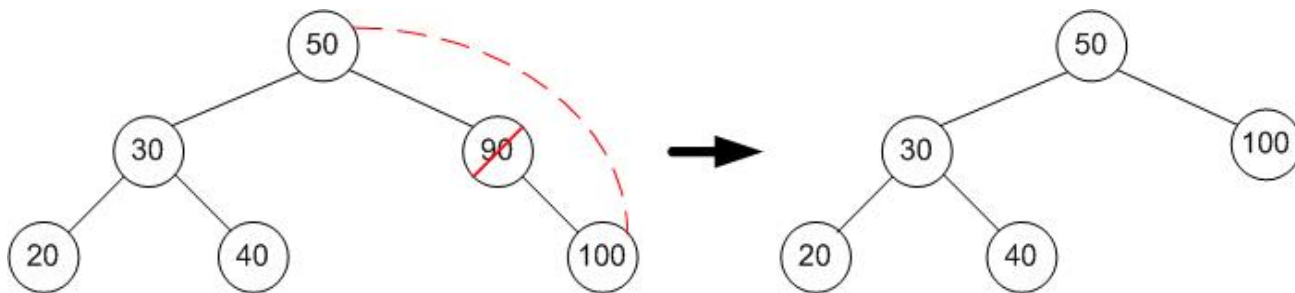
# Remoção em Árvore Binária de Busca

- **Excluir o nó folha**
  - Basta remover a folha



# Árvore Binária de Busca

- **Excluir um nó de grau 1**
  - O nó filho substitui o nó pai em sua posição.



# Árvore Binária de Busca

- **Excluir um nó de grau 2**
  - Podemos excluir esse tipo de nó de duas maneiras:
    - Se você desejar substituir o símbolo por um valor maior, substitua-o pelo **descendente direito mais à esquerda**
    - Se você desejar **substituir** o símbolo por um **valor menor**, substitua-o pelo **descendente esquerdo mais à direita**

# Remoção em Árvore Binária de Busca

- **Excluir um nó de grau 2**

■ Por exemplo:

