Estrutura de Dados I

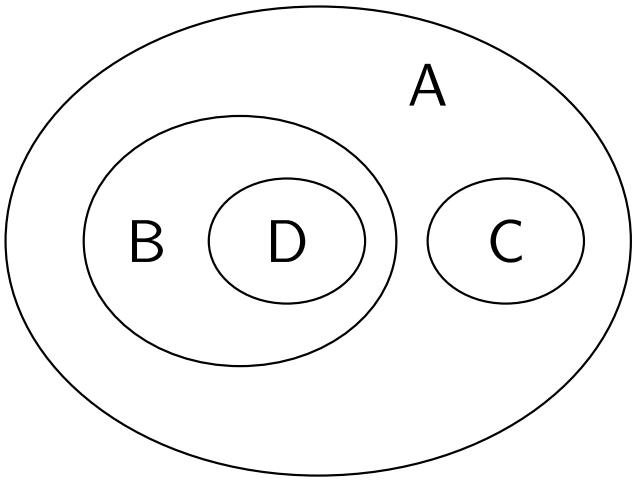
Luciana Lee

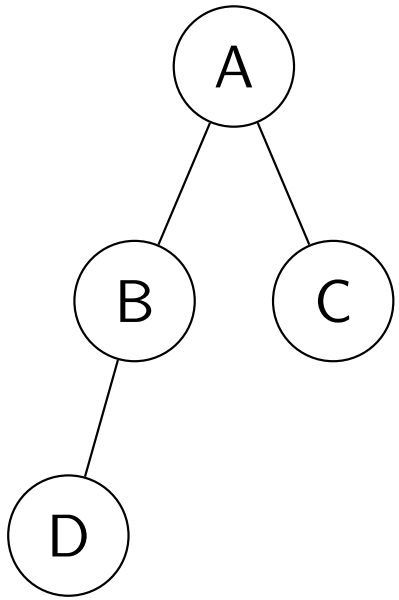
Árvore

- Uma das estruturas mais importantes da área de computação, em uma variedade de aplicações
- modela uma hierarquia entre elementos
  - Árvore genealógica
  - Diagrama hierárquico de uma organização
  - Hierarquia de pastas
- O conceito de árvores está diretamente ligado à recursão

Árvores - Formas de Representação

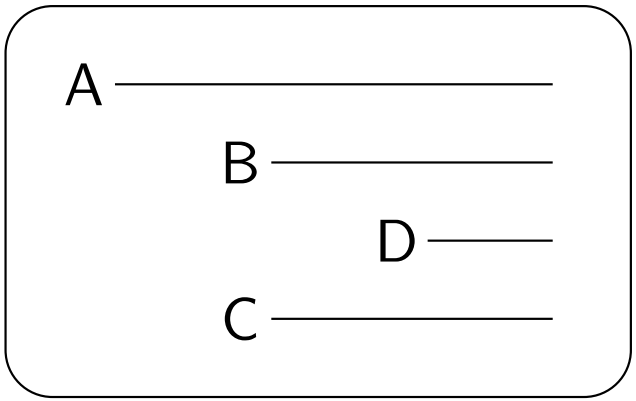
- Diagrama de Inclusão

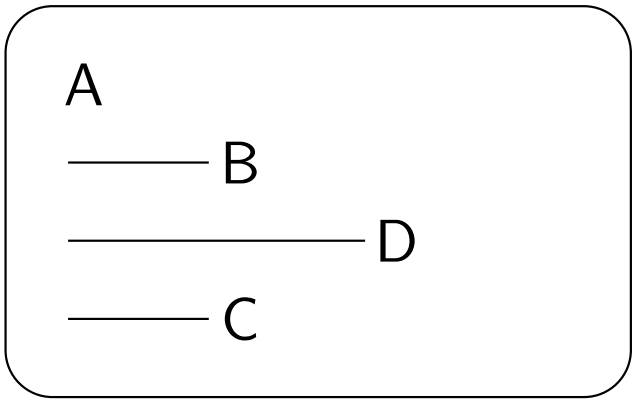


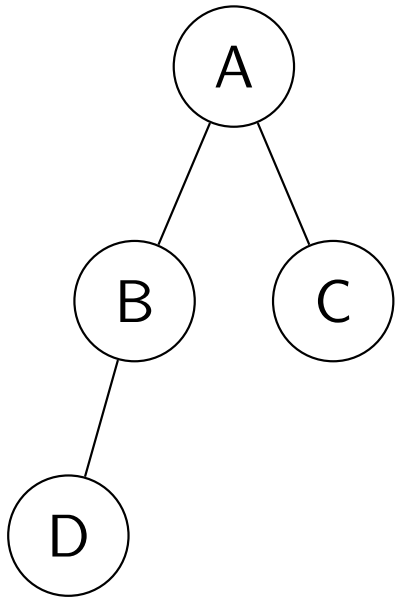


Árvores - Formas de Representação

- Diagrama de Barras







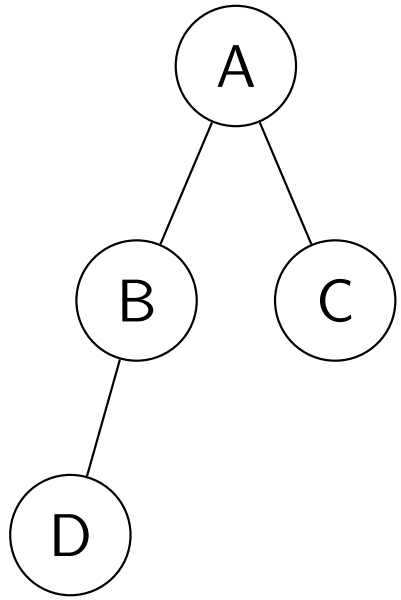
Árvores - Formas de Representação

- Níveis

1 A; 1.1 B; 1.1.1 D; 1.2 C;

- Aninhamento

(A ((B (D)) (C)))



Árvores - Definição

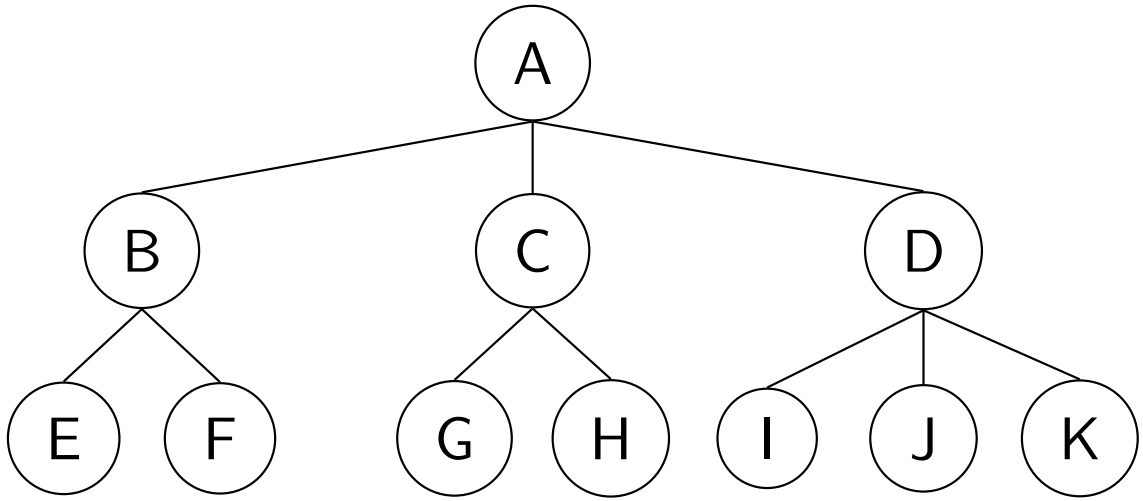
- Conjunto finito T de zero ou mais nós (nodos ou vértices), tal que:

Árvores - Definição

- Conjunto finito T de zero ou mais nós (nodos ou vértices), tal que:
  - Se o número de nós é igual a zero: a árvore é **vazia**

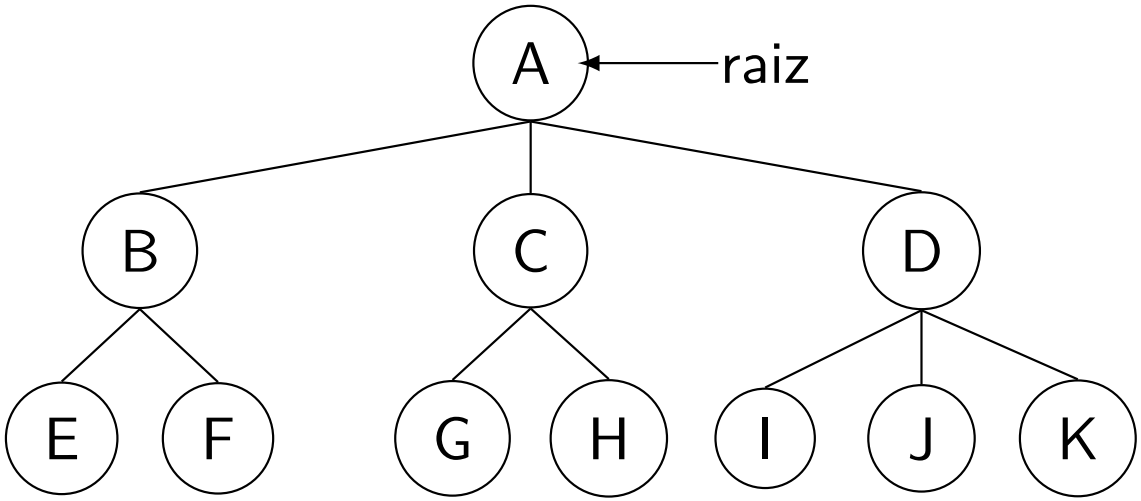
Árvores - Definição

- Conjunto finito T de zero ou mais nós (nodos ou vértices), tal que:
  - Se o número de nós é igual a zero: a árvore é **vazia**
  - Se o número de nós é maior do que zero:



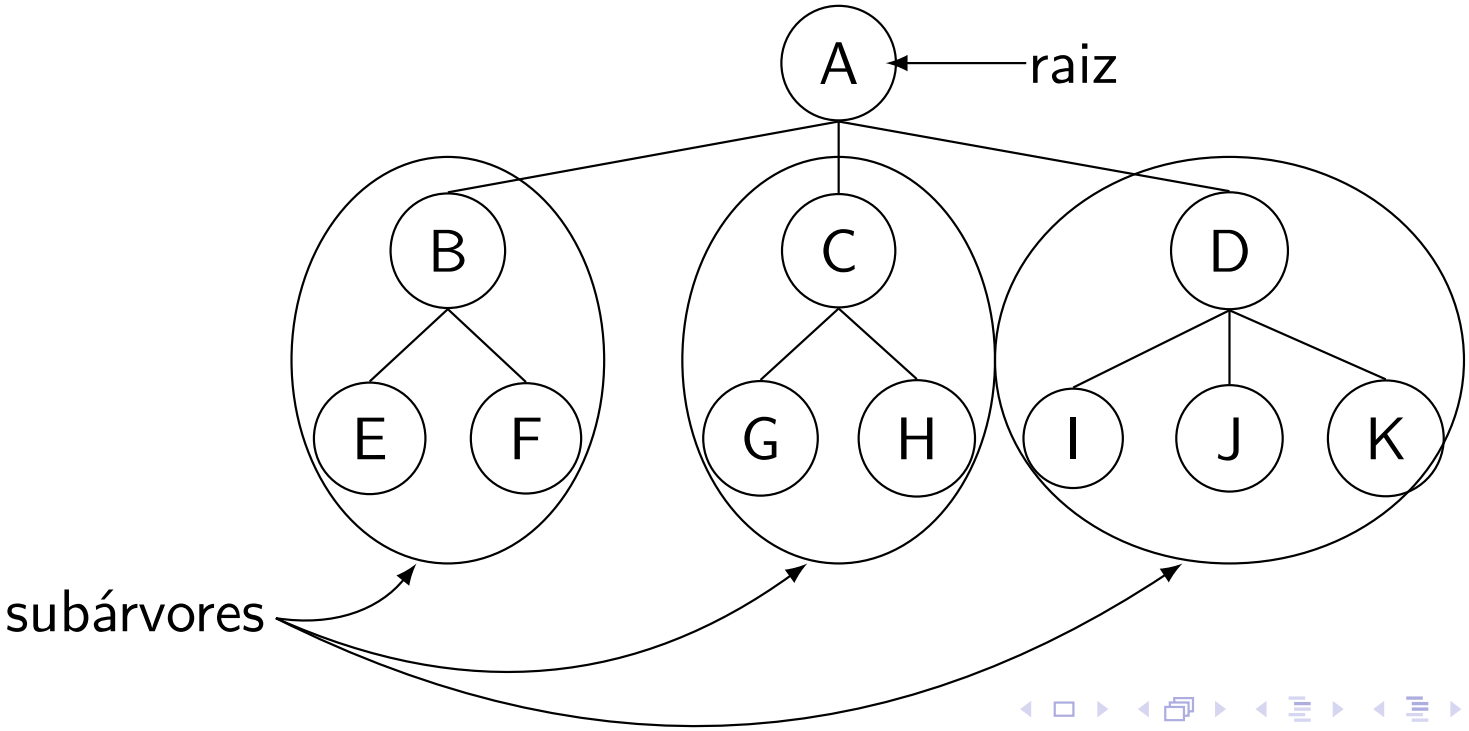
Árvores - Definição

- Conjunto finito T de zero ou mais nós (nodos ou vértices), tal que:
  - Se o número de nós é igual a zero: a árvore é **vazia**
  - Se o número de nós é maior do que zero:
    - ★ existe um nó denominado raiz da árvore, denotado por $r(T)$



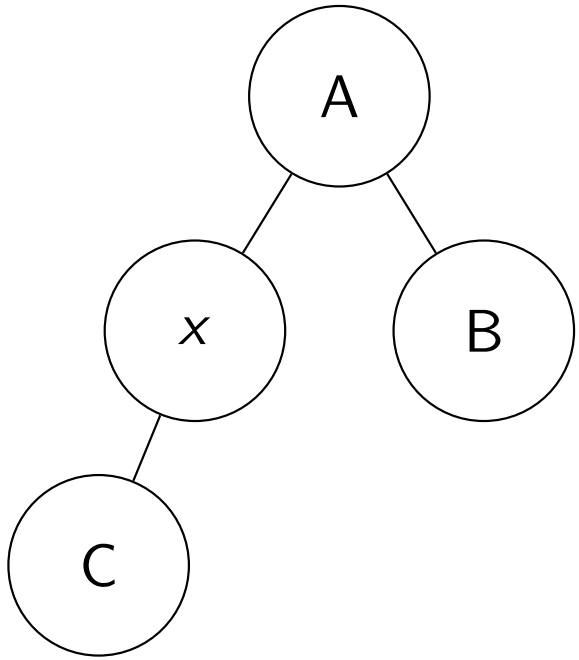
Árvores - Definição

- Conjunto finito T de zero ou mais nós (nodos ou vértices), tal que:
  - Se o número de nós é igual a zero: a árvore é **vazia**
  - Se o número de nós é maior do que zero:
    - ★ existe um nó denominado raiz da árvore, denotado por $r(T)$
    - ★ os demais nós formam $m \geq 0$ conjuntos disjuntos $S_1, S_2, \ldots, S_m$, onde cada um destes é uma árvore (S_i são denominadas **subárvores**)



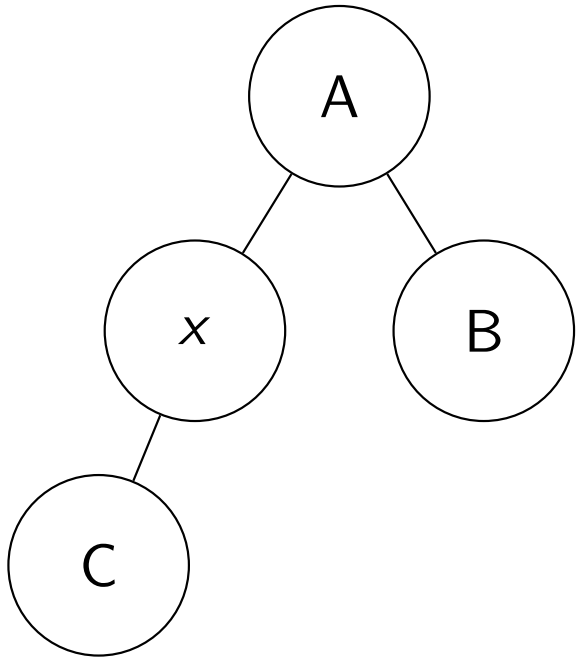
Árvores - Terminologia

- Utilizando o nó x como referencial:



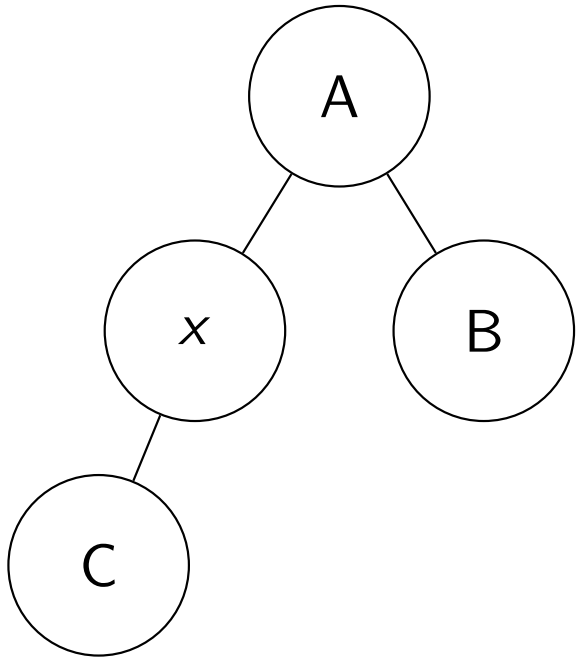
Árvores - Terminologia

- Utilizando o nó x como referencial:
  - x é filho de A;



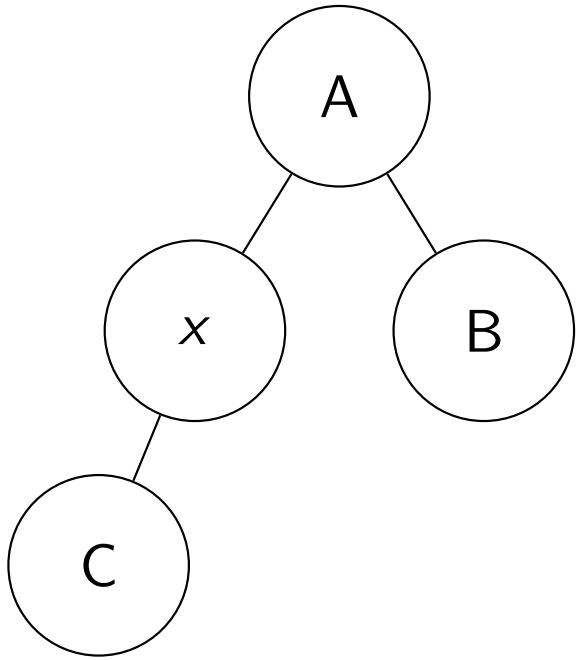
Árvores - Terminologia

- Utilizando o nó x como referencial:
  - x é filho de A;
  - x é pai de C;



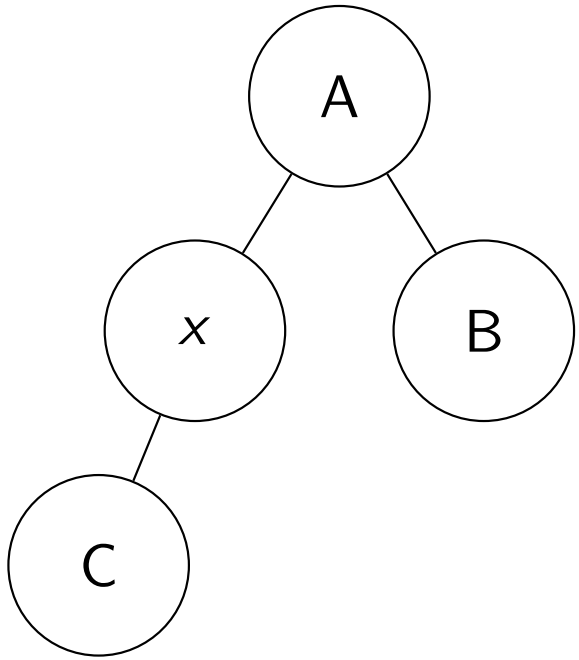
Árvores - Terminologia

- Utilizando o nó x como referencial:
  - x é filho de A;
  - x é pai de C;
  - x é irmão de B;



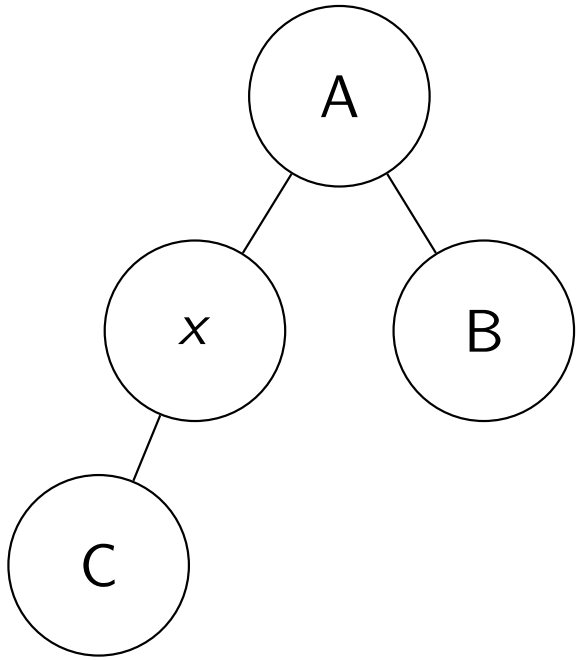
Árvores - Terminologia

- Utilizando o nó x como referencial:
  - x é filho de A;
  - x é pai de C;
  - x é irmão de B;
  - x é descendente de A;



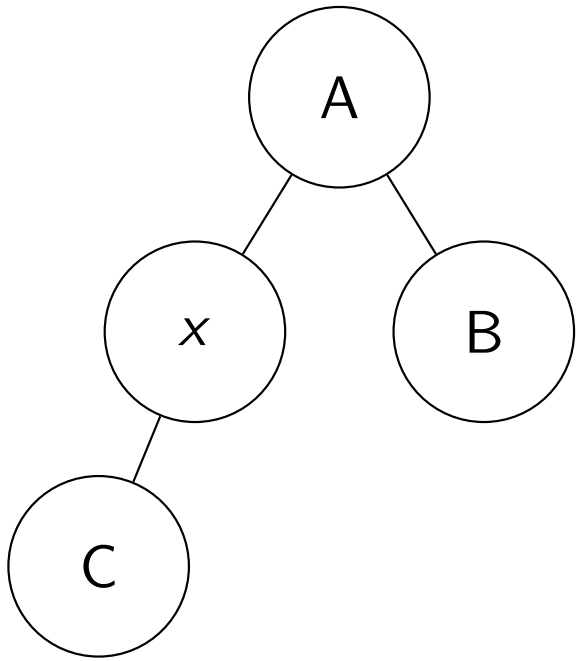
Árvores - Terminologia

- Utilizando o nó x como referencial:
  - x é filho de A;
  - x é pai de C;
  - x é irmão de B;
  - x é descendente de A;
  - A é ancestral de x;



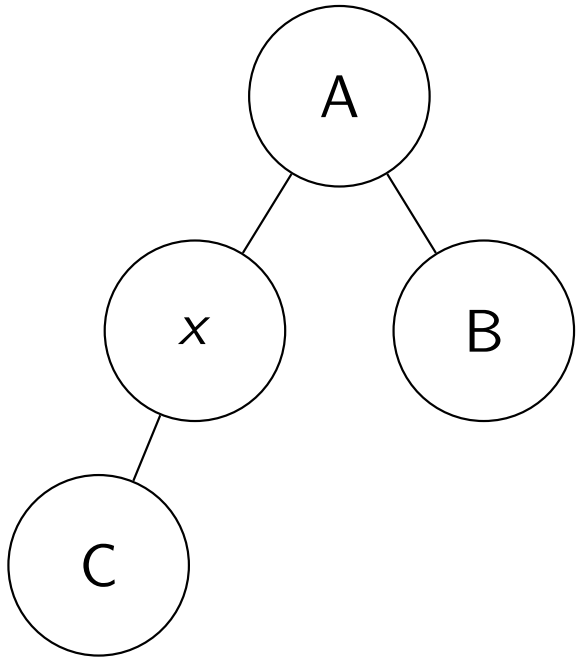
Árvores - Terminologia

- Utilizando o nó x como referencial:
  - x é filho de A;
  - x é pai de C;
  - x é irmão de B;
  - x é descendente de A;
  - A é ancestral de x;
  - Se x é diferente de A, então x é descendente próprio de A, e A é ancestral próprio de x;



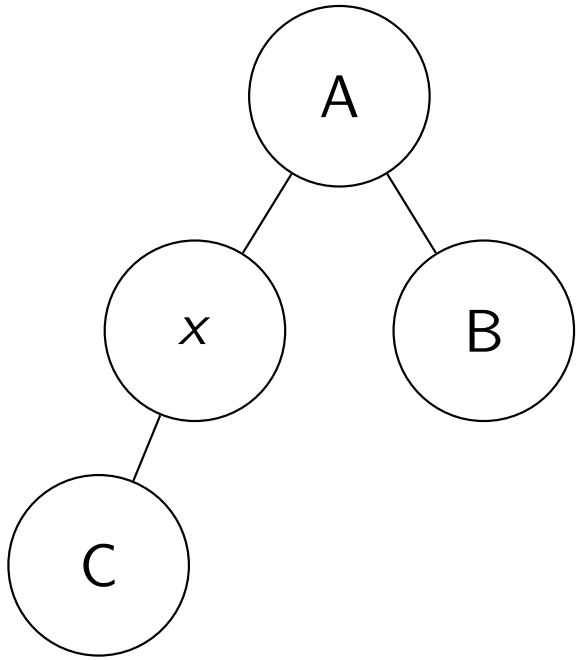
Árvores - Terminologia

- Utilizando o nó x como referencial:
  - x é filho de A;
  - x é pai de C;
  - x é irmão de B;
  - x é descendente de A;
  - A é ancestral de x;
  - Se x é diferente de A, então x é descendente próprio de A, e A é ancestral próprio de x;
  - Uma folha não possui descendentes próprios



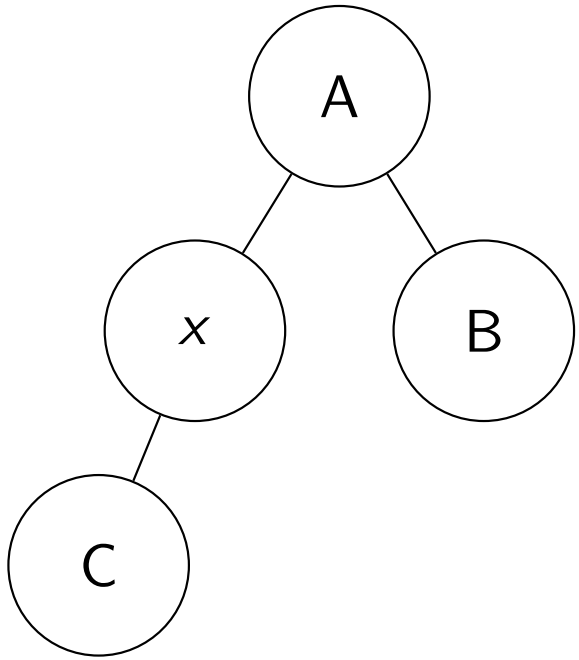
Árvores - Terminologia

- Utilizando o nó x como referencial:
  - x é filho de A;
  - x é pai de C;
  - x é irmão de B;
  - x é descendente de A;
  - A é ancestral de x;
  - Se x é diferente de A, então x é descendente próprio de A, e A é ancestral próprio de x;
  - Uma folha não possui descendentes próprios
- **Nó interno:** nó que possui um ou mais filhos.



Árvores - Terminologia

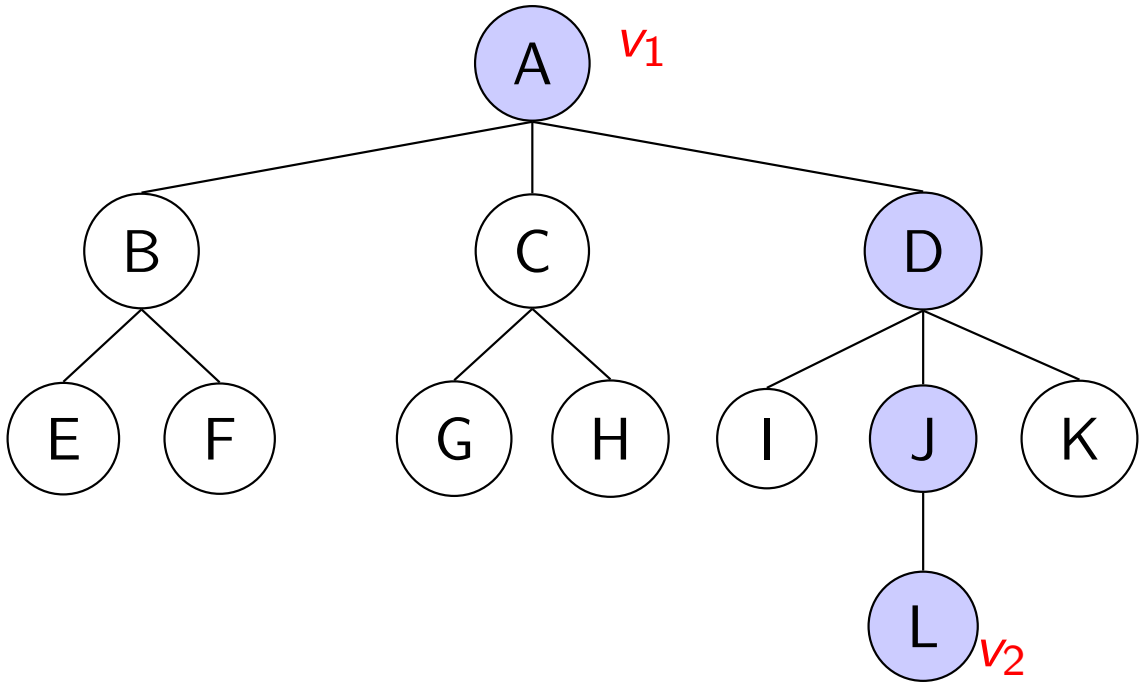
- Utilizando o nó x como referencial:
  - x é filho de A;
  - x é pai de C;
  - x é irmão de B;
  - x é descendente de A;
  - A é ancestral de x;
  - Se x é diferente de A, então x é descendente próprio de A, e A é ancestral próprio de x;
  - Uma folha não possui descendentes próprios



- **Nó interno:** nó que possui um ou mais filhos.
- **Nó folha:** nó que não possui filho

Caminho

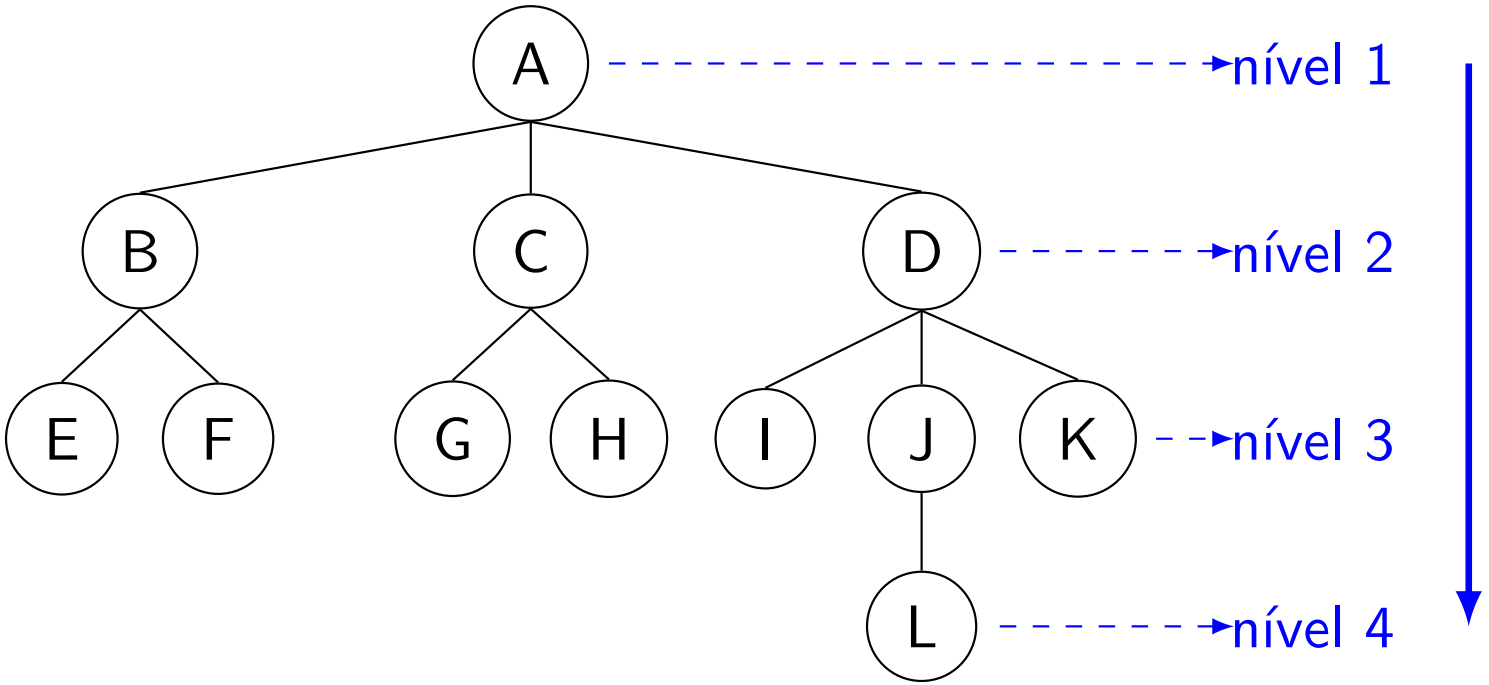
- Um **caminho** é uma sequência de nós **consecutivos distintos** entre dois nós da árvore.



- **Comprimento do caminho:** número de ligações entre os nós do caminho.

Nível

- **Nível:** número de ligações entre a raiz e o nó, acrescido de uma unidade



Altura

- **Altura de um nó:** número de ligações entre o nó e seu descendente folha de maior nível, acrescido de uma unidade.

Altura

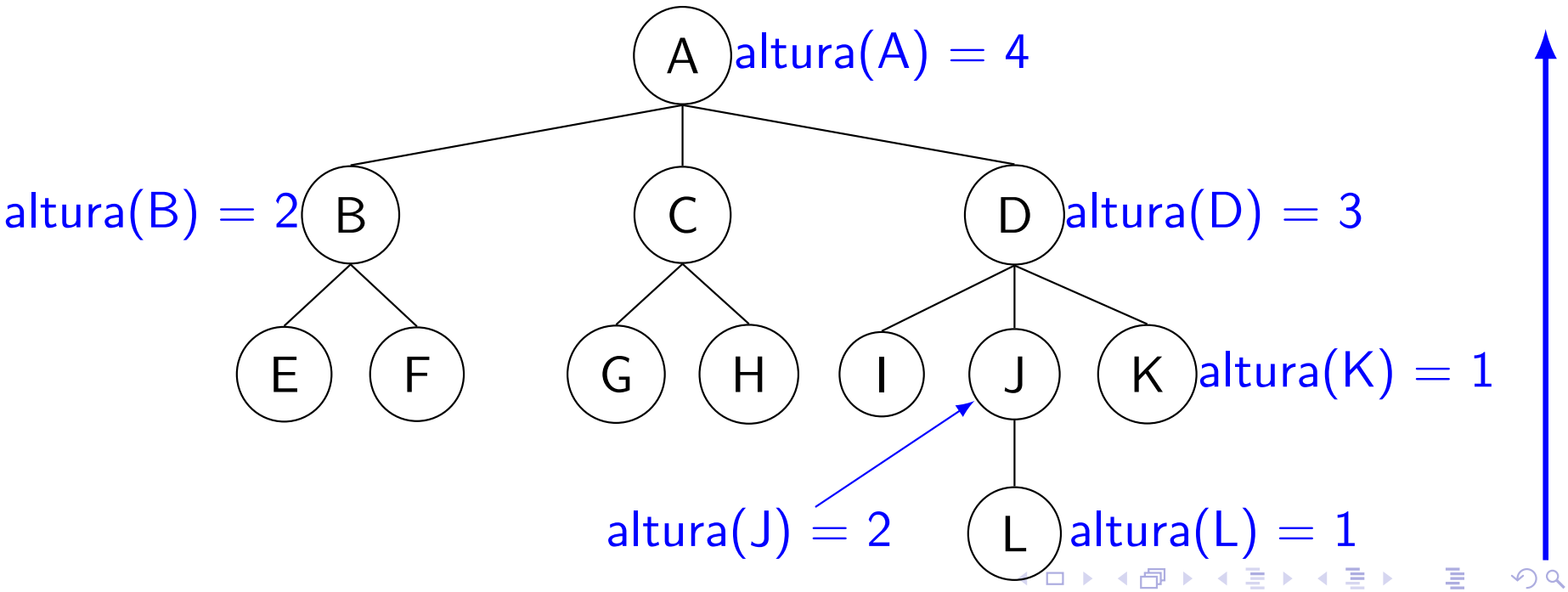
- **Altura de um nó:** número de ligações entre o nó e seu descendente folha de maior nível, acrescido de uma unidade.
- A altura de um nó folha é igual a 1.

Altura

- **Altura de um nó:** número de ligações entre o nó e seu descendente folha de maior nível, acrescido de uma unidade.
- A altura de um nó folha é igual a 1.
- A altura da árvore é a altura de seu nó raiz.

Altura

- **Altura de um nó:** número de ligações entre o nó e seu descendente folha de maior nível, acrescido de uma unidade.
- A altura de um nó folha é igual a 1.
- A altura da árvore é a altura de seu nó raiz.
- É equivalente afirmar que a altura de uma árvore é o maior nível dentre seus nós.



Árvores Binárias

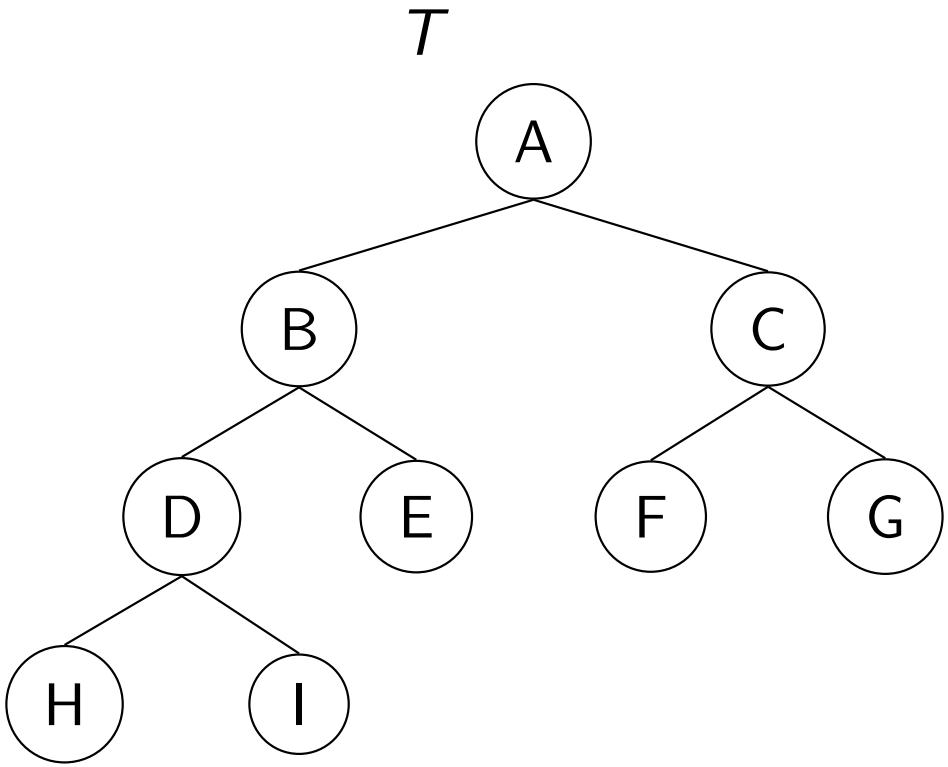
- As árvores binárias são uma das árvores mais utilizadas em computação.

Árvore Binária - Definição

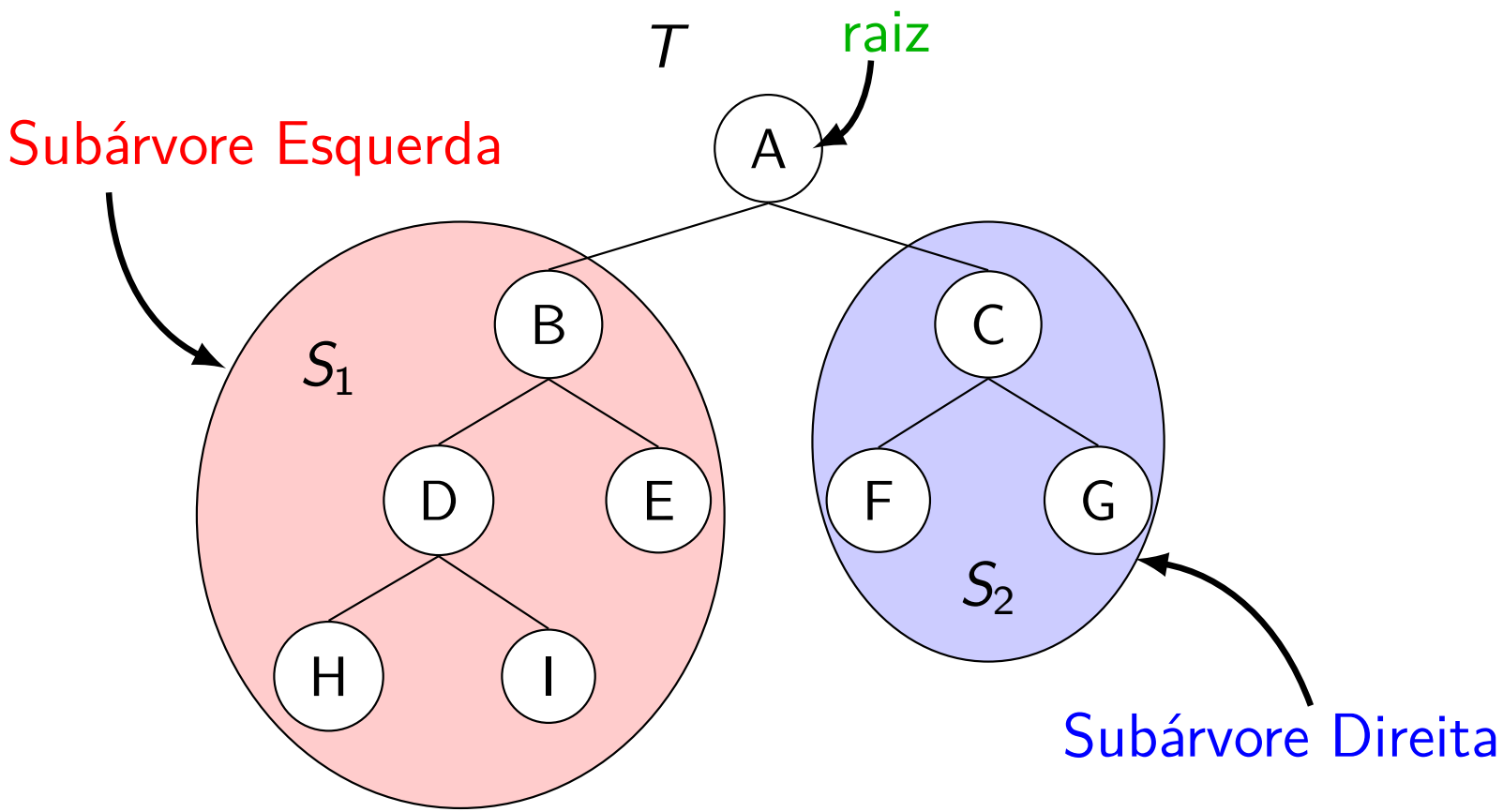
Uma árvore binária é um conjunto T de zero ou mais nós, tal que:

- Se o número de nós é maior ou igual a um:
  - existe um nó denominado raiz da árvore
  - os demais nós formam 2 conjuntos disjuntos S_1, S_2 (subárvore da esquerda e subárvore da direita), onde cada um destes é uma árvore binária.
- Se o número de nós é igual a zero, então a árvore é vazia.

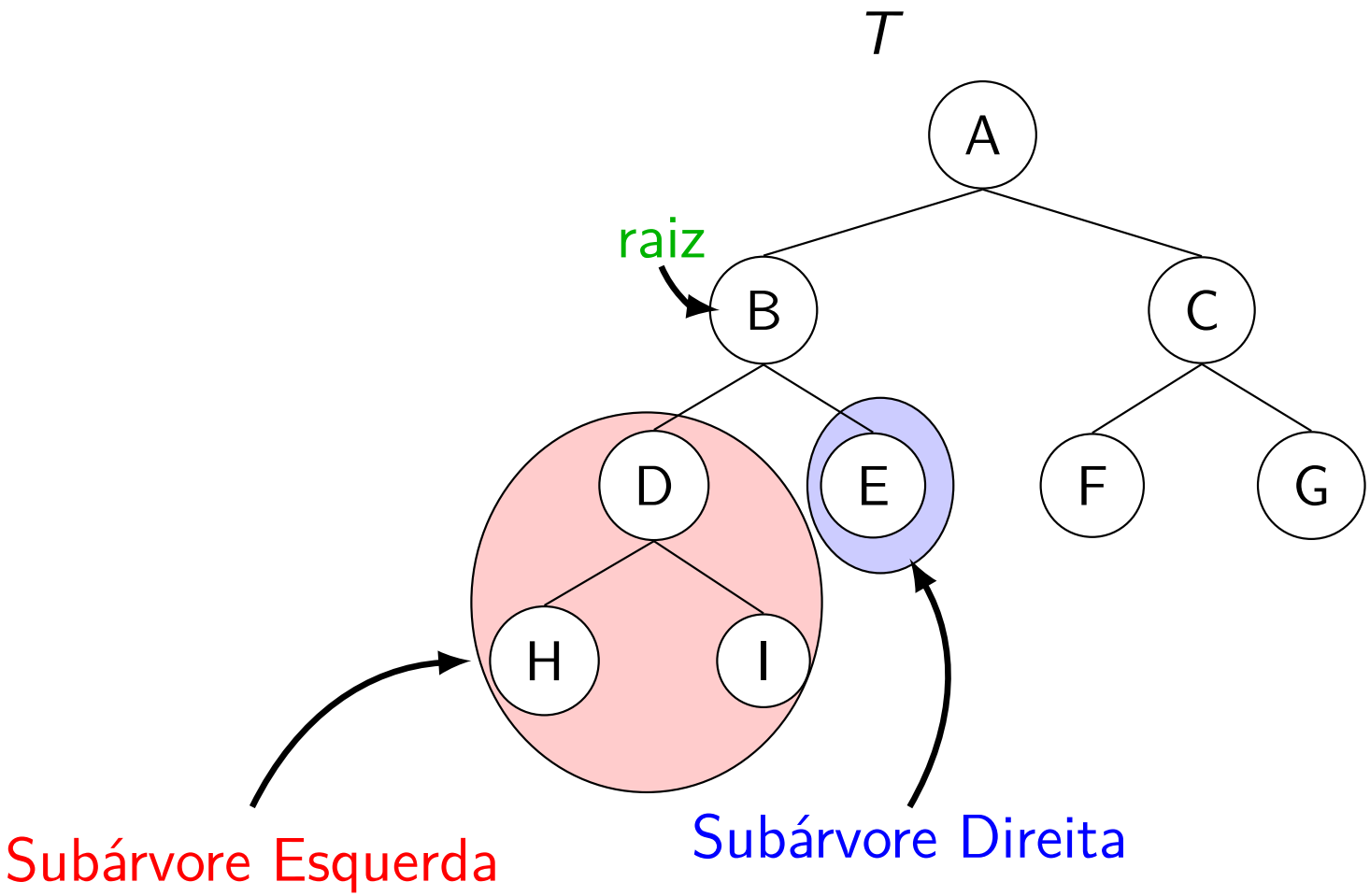
Árvores Binárias



Árvores Binárias

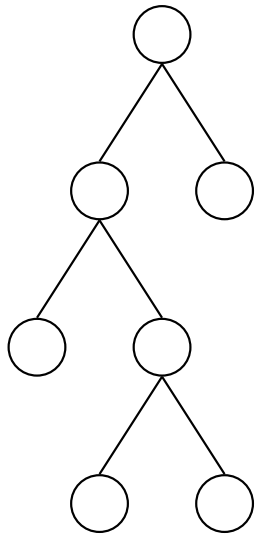


Árvores Binárias



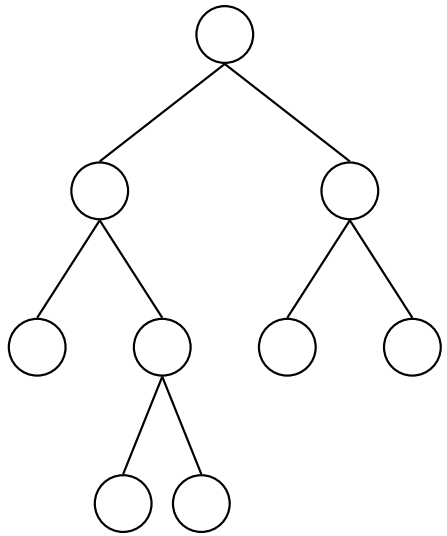
Tipos de Árvores Binárias

- **Árvore Estritamente Binária:** cada nó da árvore possui 0 ou 2 filhos.



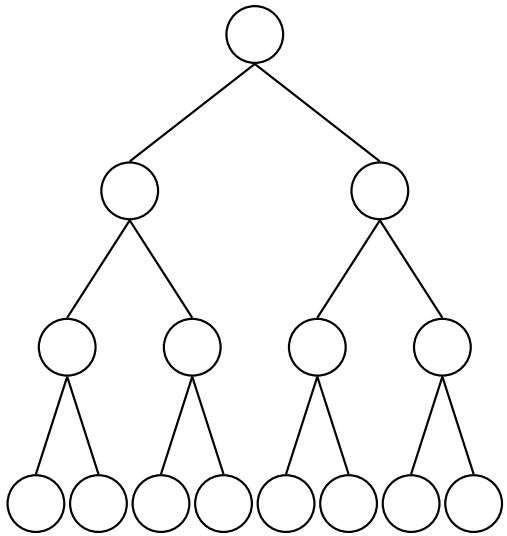
Tipos de Árvores Binárias

- **Árvore Binária Completa:** todos os nós folhas estão no último ou penúltimo nível.



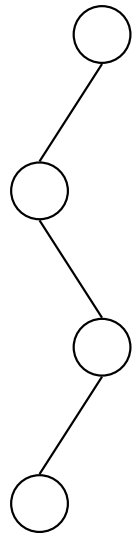
Tipos de Árvores Binárias

- **Árvore Binária Cheia:** todos os nós folha estão no último nível.



Tipos de Árvores Binárias

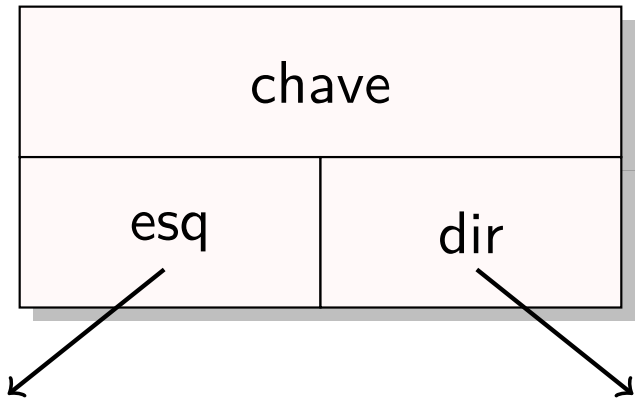
- **Árvore Zigue-zague:** nós internos com uma subárvore vazia.



Implementação de uma Árvore Binária

- Temos, então, a declaração de uma estrutura para um nó :

```
typedef struct node{
    int chave;
    struct node *esq, *dir;
}No;
```



Criação de um nó

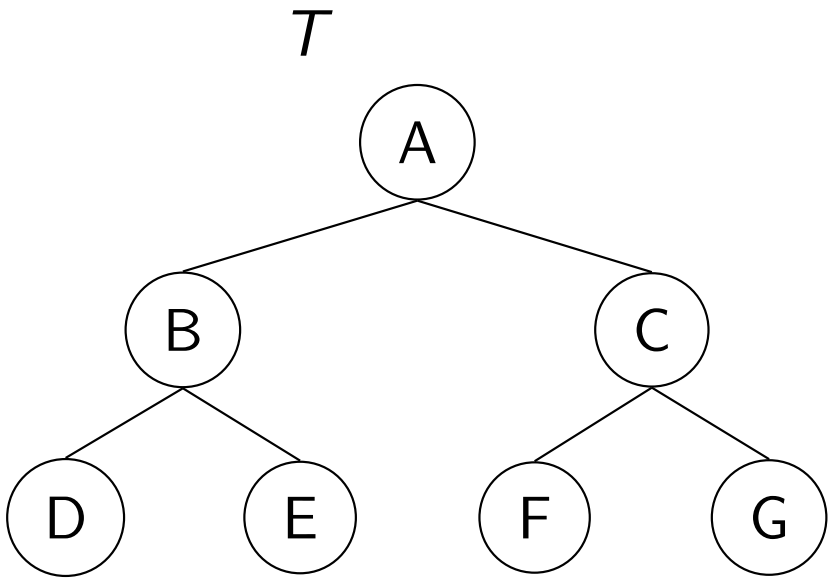
```
No *criaNo (int ch){
  No *novo = (No *)calloc(1, sizeof(No));
  if (!novo) {
    printf("ERRO: nao foi possivel alocar novo
        no.\n");
    exit(1);
  }
  novo->chave = ch;
  return novo;
}
```

Percursos em Árvores Binárias

- Quando realizamos um percurso em uma árvore, cada nó da árvore deve ser **"visitado"** apenas uma vez.
- O percurso deve sempre começar do nó raiz.
- "Visitar um nó" = acessar um nó para realização de alguma ação.
- Exemplo: imprimir as informações contidas no nó.

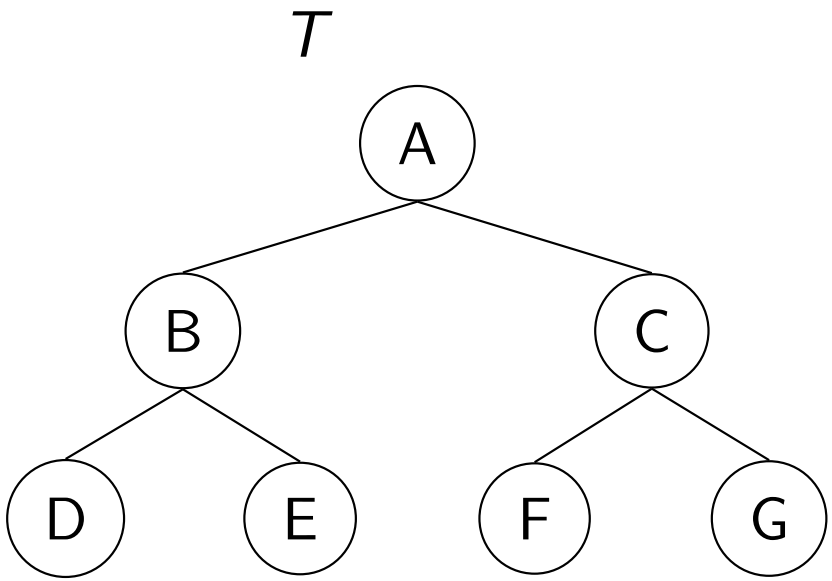
Percursos em Árvores Binárias

- Percurso em Pré-Ordem:
  - Visita a raiz;
  - Percorre a subárvore esquerda da raiz;
  - Percorre a subárvore direira da raiz.
- Percurso em Ordem Simétrica:
  - Percorre a subárvore esquerda da raiz;
  - Visita a raiz.
  - Percorre a subárvore direira da raiz;
- Percurso em Pós-Ordem:
  - Percorre a subárvore esquerda da raiz;
  - Percorre a subárvore direira da raiz;
  - Visita a raiz.



Percursos em Árvores Binárias

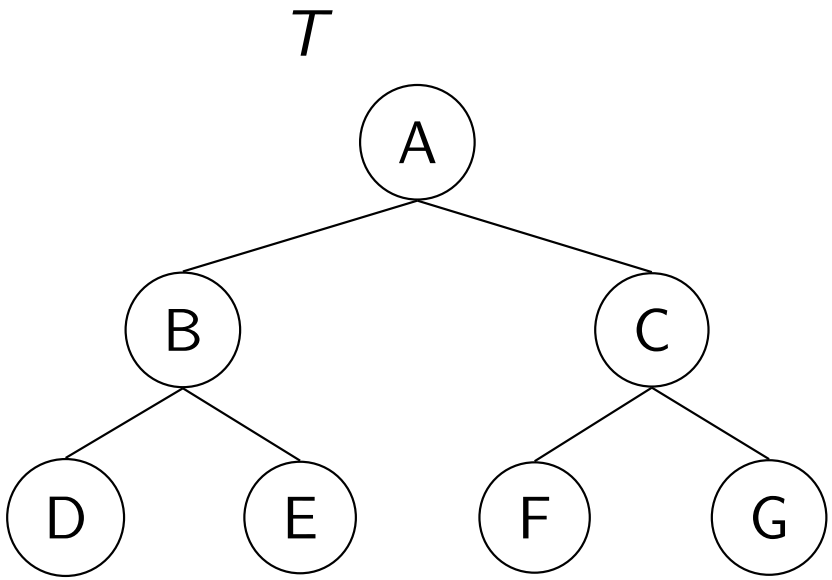
- Percurso em Pré-Ordem:
  - Visita a raiz;
  - Percorre a subárvore esquerda da raiz;
  - Percorre a subárvore direira da raiz.
- Percurso em Ordem Simétrica:
  - Percorre a subárvore esquerda da raiz;
  - Visita a raiz.
  - Percorre a subárvore direira da raiz;
- Percurso em Pós-Ordem:
  - Percorre a subárvore esquerda da raiz;
  - Percorre a subárvore direira da raiz;
  - Visita a raiz.



A - B - D - E - C - F - G

Percursos em Árvores Binárias

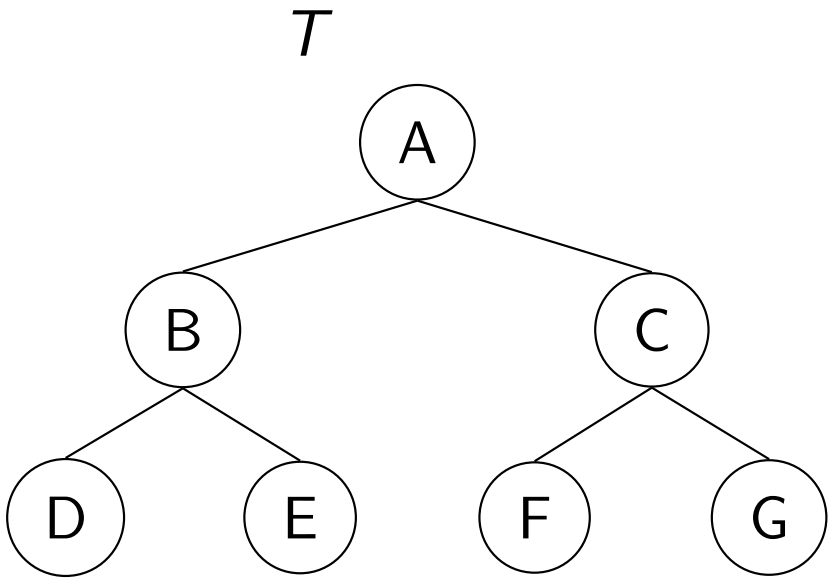
- Percurso em Pré-Ordem:
  - Visita a raiz;
  - Percorre a subárvore esquerda da raiz;
  - Percorre a subárvore direira da raiz.
- Percurso em Ordem Simétrica:
  - Percorre a subárvore esquerda da raiz;
  - Visita a raiz.
  - Percorre a subárvore direira da raiz;
- Percurso em Pós-Ordem:
  - Percorre a subárvore esquerda da raiz;
  - Percorre a subárvore direira da raiz;
  - Visita a raiz.



D - B - E - A - F - C - G

Percursos em Árvores Binárias

- Percurso em Pré-Ordem:
  - Visita a raiz;
  - Percorre a subárvore esquerda da raiz;
  - Percorre a subárvore direira da raiz.
- Percurso em Ordem Simétrica:
  - Percorre a subárvore esquerda da raiz;
  - Visita a raiz.
  - Percorre a subárvore direira da raiz;
- Percurso em Pós-Ordem:
  - Percorre a subárvore esquerda da raiz;
  - Percorre a subárvore direira da raiz;
  - Visita a raiz.



D - E - B - F - G - C - A

Percursos em Árvores Binárias

- Vamos ver um exemplo de implementação para os percursos em árvores binárias?

Percursos em Árvores Binárias

- O percurso em Pré-Ordem também é um **percurso em Profundidade**.
- Além dos percursos que vimos, ainda podemos realizar um percurso em que os nós da árvore são visitados por nível, da esquerda para a direita. Tal percurso se chama: **Percurso em Largura**.
- Para realizar o percurso em largura, precisamos de uma estrutura auxiliar: FILA.

Percursos em Árvores Binárias

- O percurso em largura executa os seguintes passos:
  1. Adicionar a raiz na FILA;
  2. Repetir até que a FILA fique vazia:
     a. Retirar o primeiro elemento da FILA (visita)
     b. Se o filho da esquerda do elemento for diferente de NULL, então adicioná-lo na FILA;
     c. Se o filho da direita do elemento for diferente de NULL, então adicioná-lo na FILA;
- Vamos ver a implementação?

Árvore Binária de Busca

- Diversas aplicações precisam buscar um determinado valor em um conjunto de dados
- Essa busca deve ser feita da forma mais eficiente possível
- Árvores binárias possibilitam buscas com eficiência

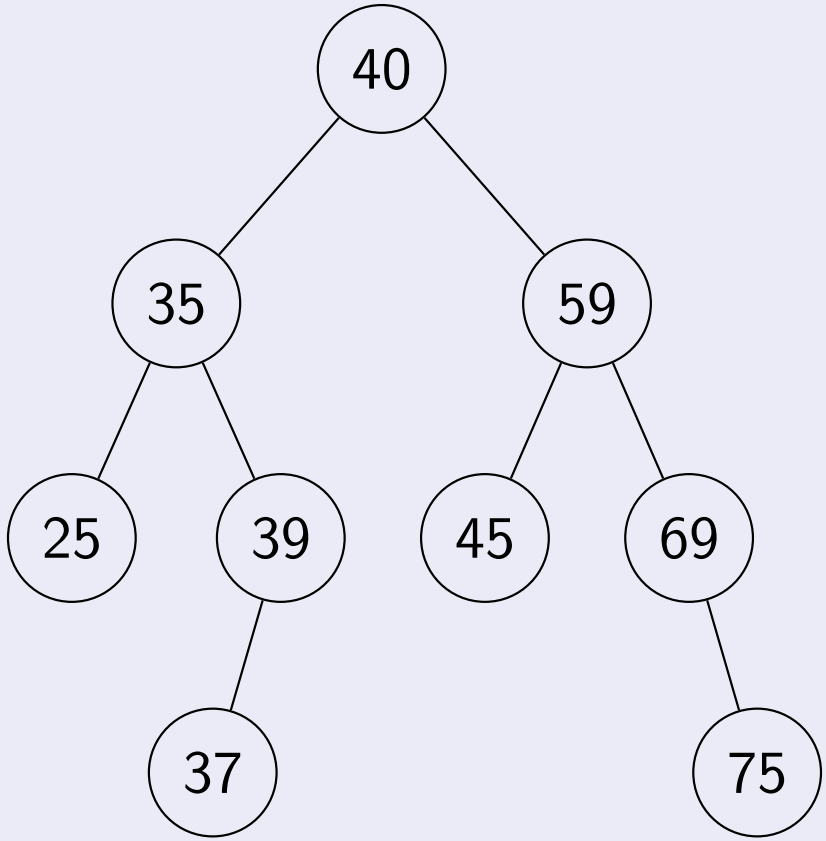
Árvore Binária de Busca

Uma árvore binária T é uma árvore binária de busca se:

- Chaves da subárvore esquerda de T são menores do que chave da raiz de T; e
- Chaves da subárvore da direita de T são maiores do que a chave da raiz de T; e
- Subárvores da esquerda e da direita de T são árvores binárias de busca.

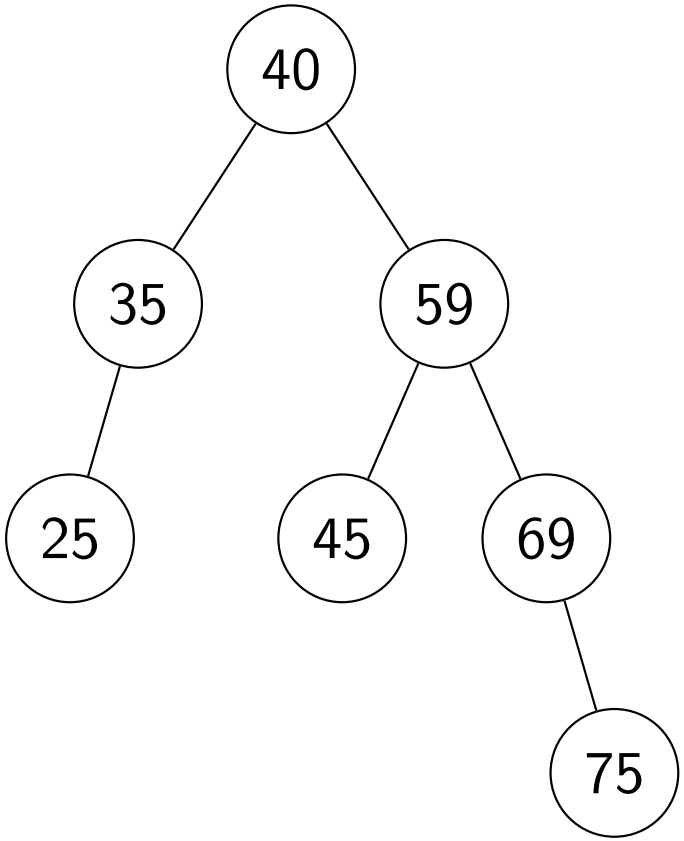
Árvore Binária de Busca

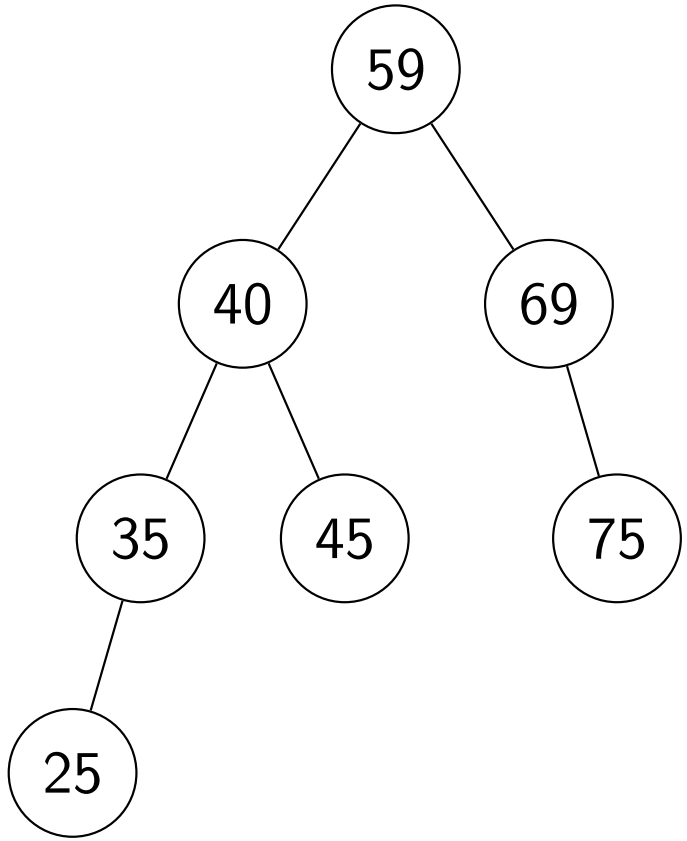
Exemplo



Árvore Binária de Busca

- Para um mesmo conjunto de chaves, existem várias árvores binárias de busca possíveis





Árvore Binária de Busca: Busca por uma chave

- Dada uma árvore de busca T e um número k, encontrar um nó de T cuja chave seja k.

Árvore Binária de Busca: Busca por uma chave

- Dada uma árvore de busca T e um número k, encontrar um nó de T cuja chave seja k.
- Operações:

Árvore Binária de Busca: Busca por uma chave

- Dada uma árvore de busca T e um número k, encontrar um nó de T cuja chave seja k.
- Operações:
  - Buscar uma chave na árvore;

Árvore Binária de Busca: Busca por uma chave

- Dada uma árvore de busca T e um número k, encontrar um nó de T cuja chave seja k.
- Operações:
  - Buscar uma chave na árvore;
  - Inserir uma nova chave na árvore;

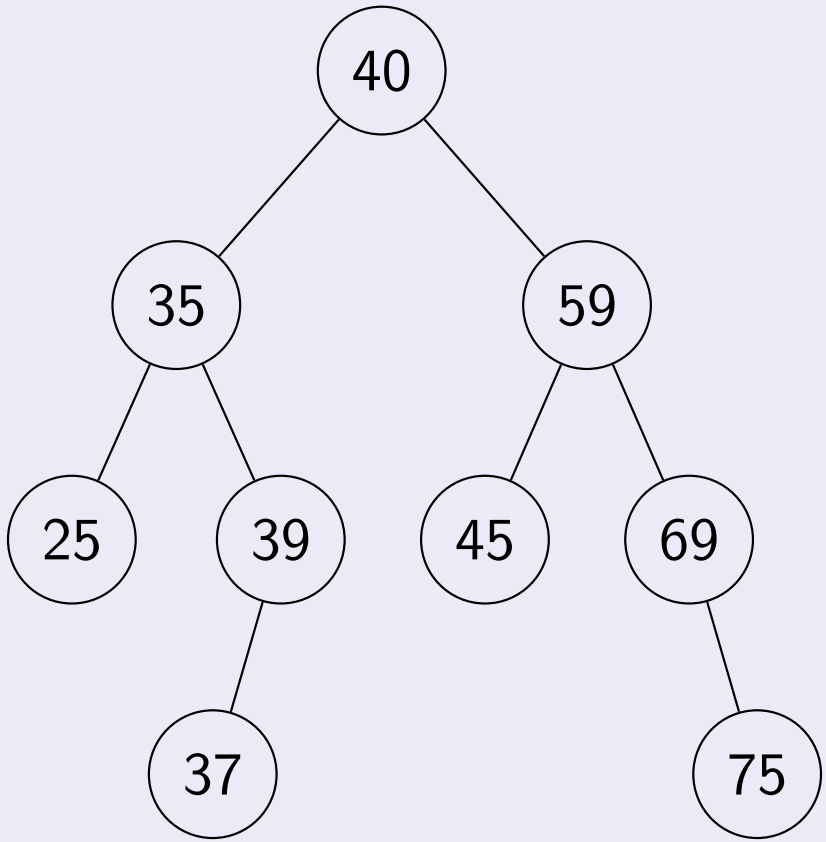
Árvore Binária de Busca: Busca por uma chave

- Dada uma árvore de busca T e um número k, encontrar um nó de T cuja chave seja k.
- Operações:
  - Buscar uma chave na árvore;
  - Inserir uma nova chave na árvore;
  - Excluir uma chave na árvore.

Árvore Binária de Busca: Busca por uma chave

- Exemplo: buscar o número 37 na árvore.

Exemplo



Árvore Binária de Busca: Busca por uma chave

- Vamos ver a implementação?

Árvore Binária de Busca: Inserção

- Considere o problema de inserir um novo nó, com chave k, em uma árvore de busca.

Árvore Binária de Busca: Inserção

- Considere o problema de inserir um novo nó, com chave k, em uma árvore de busca.
- A árvore resultante deve também ser de busca.

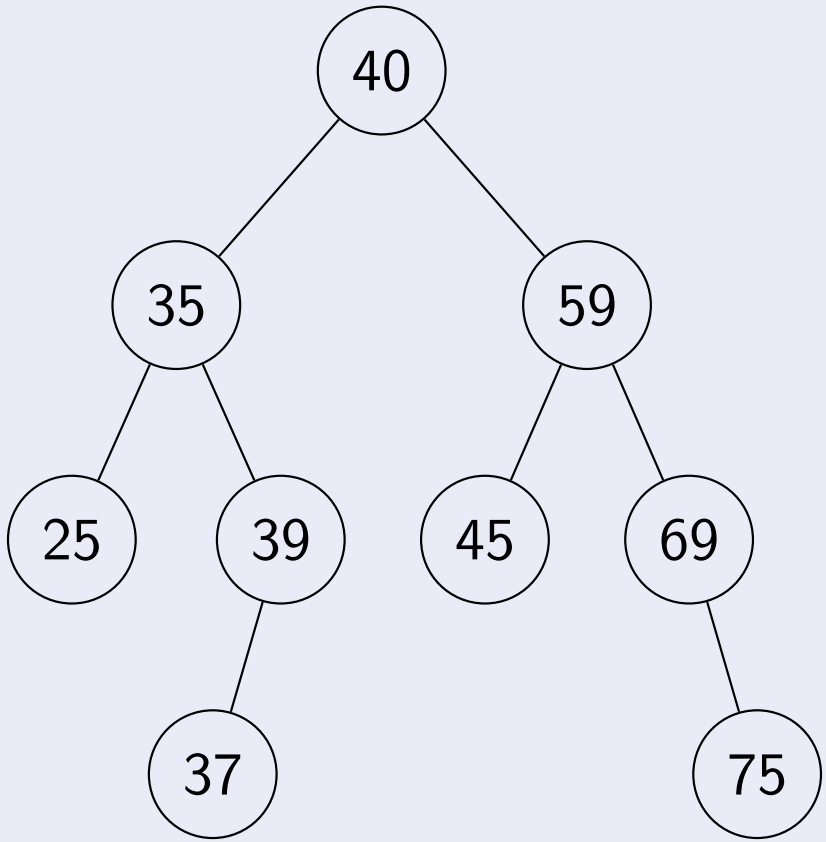
Árvore Binária de Busca: Inserção

- Considere o problema de inserir um novo nó, com chave k, em uma árvore de busca.
- A árvore resultante deve também ser de busca.
- O novo nó será uma folha da árvore resultante.

Árvore Binária de Busca: Inserção

- Exemplo: inserir o número 42 na árvore.

Exemplo



Árvore Binária de Busca

- Vamos ver a implementação?

Dúvidas?