



PECS | Programa de Pós-Graduação em  
Engenharia de Computação e Sistemas

# ALGORITMOS E ESTRUTURAS DE DADOS

Disciplina: ALGORITMOS E ESTRUTURAS DE DADOS

Carga Horária: 60h

Professor: **Dr. Reinaldo**

# Estruturas de dados do tipo árvore

As estruturas de dados do tipo árvore **são não lineares**, ou seja, os elementos que as compõem não estão armazenados de forma sequencial e também não estão todos encadeados.

# Conceito

- As árvores são estruturas de dados hierárquicas amplamente utilizadas em computação.
- Elas organizam informações em uma estrutura ramificada, com um nó raiz e vários nós filhos.



## Conceitos básicos da árvore

Uma árvore é composta por nós, que representam informações, conectados por arestas, que representam a relação hierárquica. O nó raiz é o nó superior, e os nós filhos são conectados a ele.

Nó raiz	O nó principal da árvore, que serve como ponto de partida para navegar na estrutura de dados.
Nós filhos	Nós que estão conectados a um nó pai. Cada nó pode ter vários nós filhos.
Nó Pai	O elemento superior na hierarquia, com nós filhos conectados.
Altura da árvore	O número de níveis na árvore, medido a partir do nó raiz.

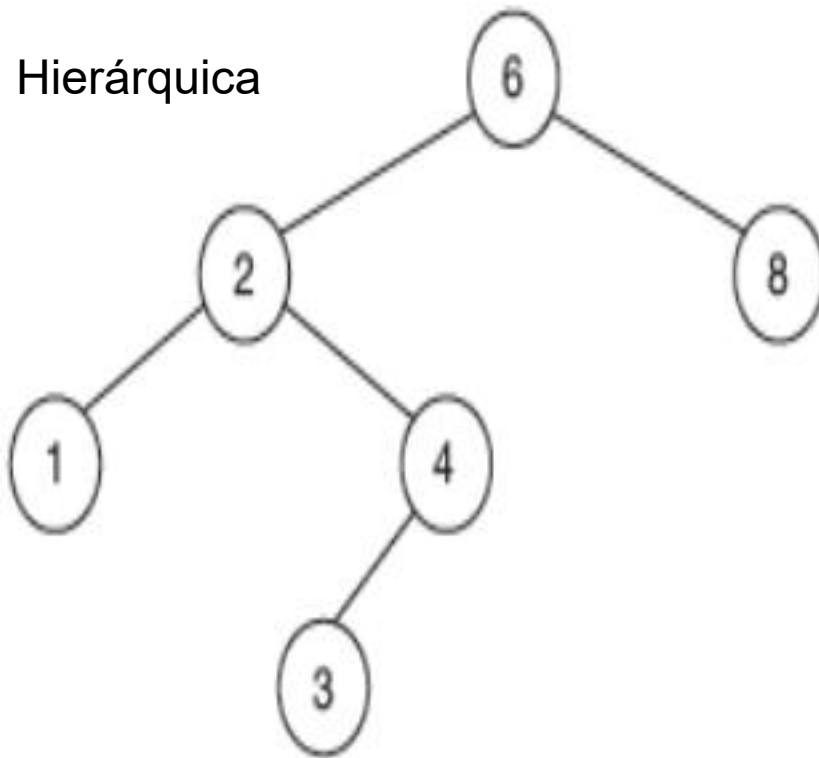
# Árvores na tecnologia: exemplos de uso

As árvores são uma estrutura de dados fundamental em várias áreas da tecnologia.

Sistemas de arquivos	Organização de arquivos e pastas, com pastas como nós filhos de pastas pai.
Algoritmos de busca	Árvores binárias e árvores AVL permitem buscas eficientes em conjuntos de dados ordenados.
Compiladores	Árvores sintáticas representam a estrutura gramatical de um programa, facilitando a análise e interpretação.
Banco de dados	Árvores B são usadas para indexar dados, garantindo um acesso rápido a registros específicos.

# Representações

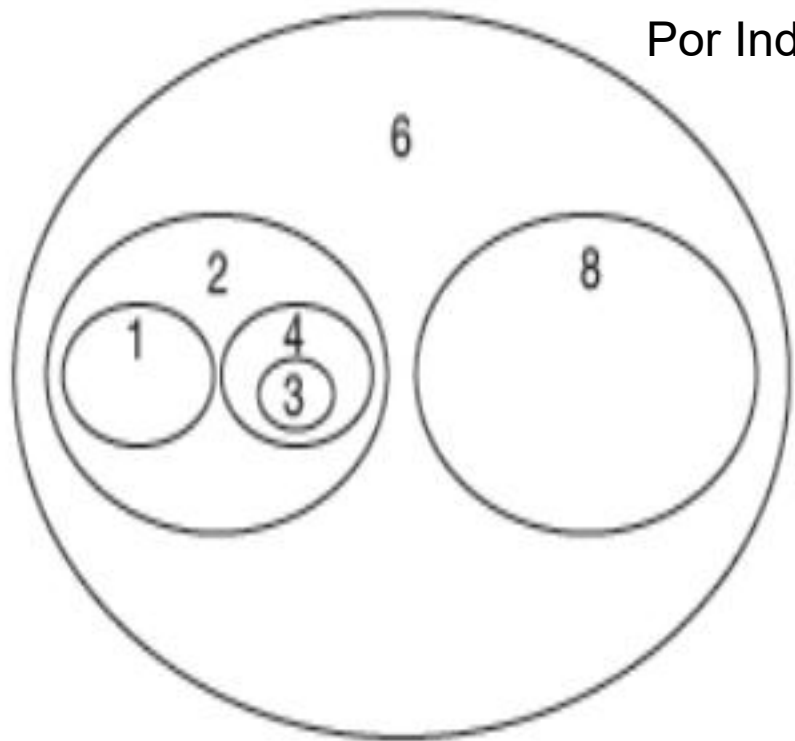
As representações podem ser ilustradas de três formas:



Por Parênteses Aninhados

6(2(1 4(3)) 8)

Por Indução





# Vantagens da utilização de árvores

As árvores oferecem diversas vantagens em comparação com outras estruturas de dados, tornando-as uma escolha popular em várias aplicações.

## **Eficiência de busca**

Árvores bem balanceadas permitem busca eficiente em tempo logarítmico, tornando-as ideais para grandes conjuntos de dados.

## **Organização hierárquica**

Sua estrutura hierárquica facilita a representação de dados relacionados e a navegação entre eles.

## **Flexibilidade**

As árvores podem ser usadas para representar uma ampla variedade de informações, desde estruturas de dados simples até sistemas complexos.





# Desafios e limitações no uso de árvores

## Complexidade

- Implementação e gerenciamento de árvores podem ser complexos, especialmente para árvores complexas e balanceadas.

## Desempenho

- Árvores desbalanceadas podem levar a tempos de busca longos, impactando o desempenho do sistema.

## Uso de memória

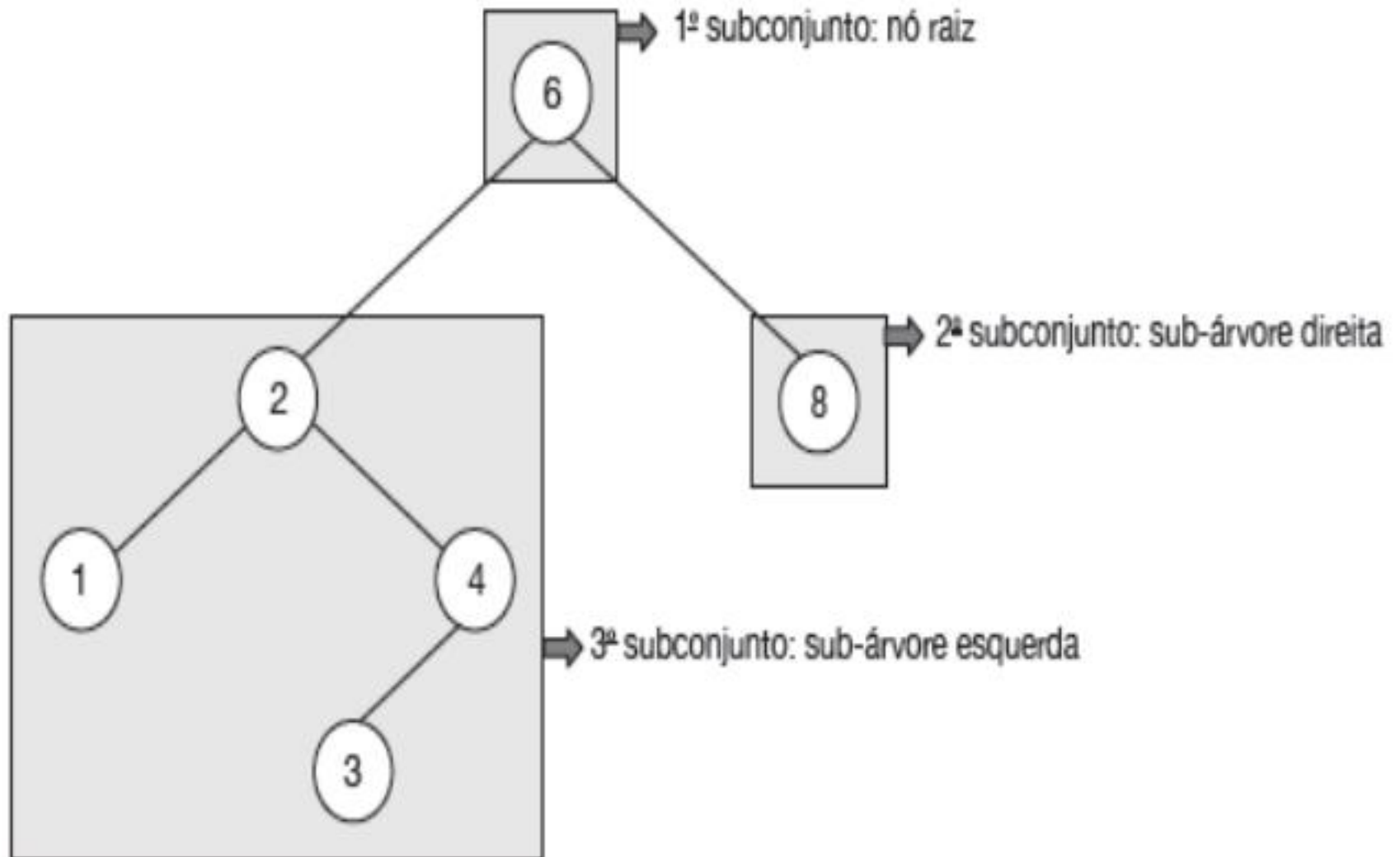
- As árvores podem ocupar uma quantidade significativa de memória, especialmente em conjuntos de dados grandes



# Árvore binária

Conjunto finito de elementos, em que cada um é denominado **nó** e o primeiro é conhecido como **raiz**. Pode estar vazio ou ser particionado em três subconjuntos: 1<sup>o</sup> subconjunto (**nó raiz**), 2<sup>o</sup> subconjunto (**sub-árvore direita**) e 3<sup>o</sup> subconjunto (**sub-árvore esquerda**).

# Árvore binária



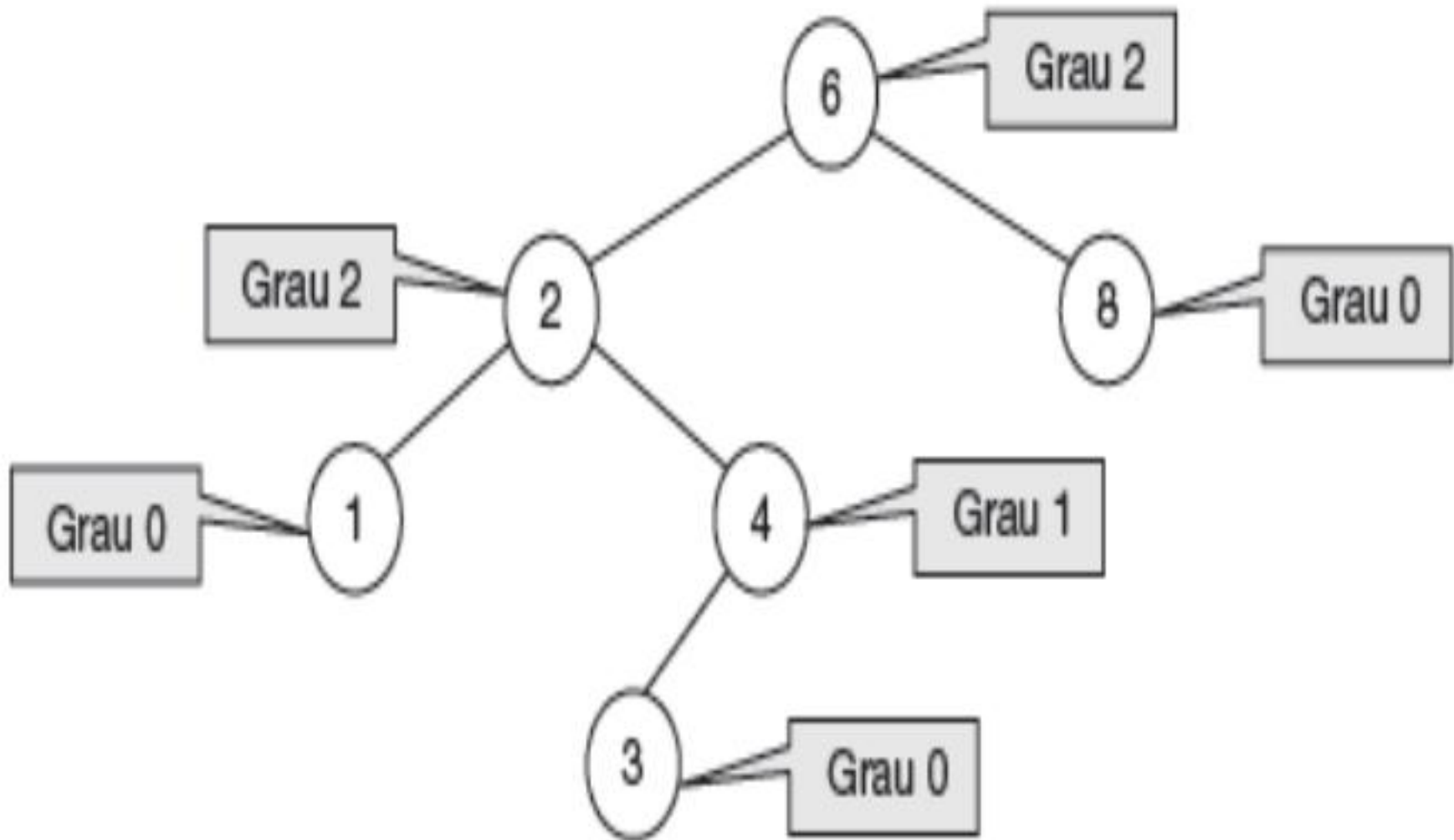
# Propriedades da árvore binária

- a) Todos os nós de uma sub-árvore direita são maiores que o nó raiz.
- b) Todos os nós de uma sub-árvore esquerda são menores que o nó raiz.
- c) Cada sub-árvore é também uma árvore binária.
- d) O grau de um nó representa o seu número de sub-árvores.

# Propriedades da árvore binária

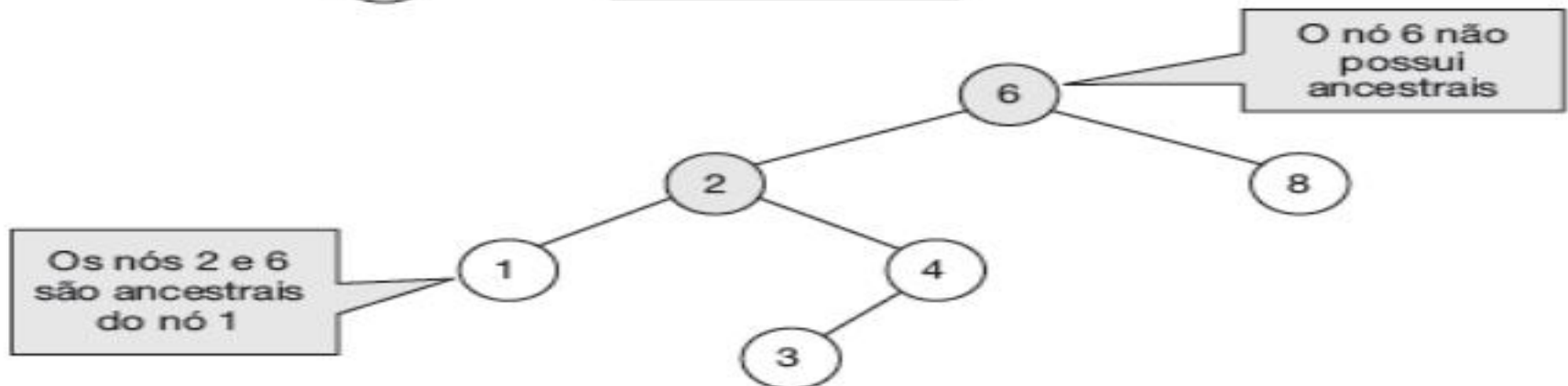
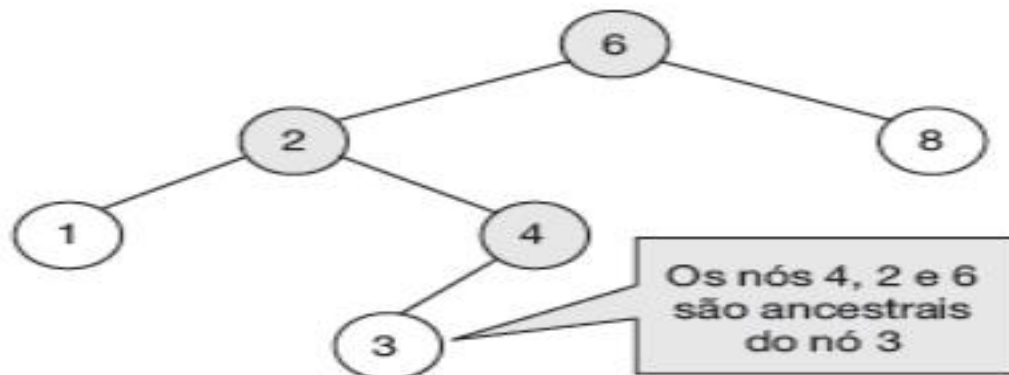
- e) Na árvore binária, o grau máximo de um nó é 2.
- f) O grau de uma árvore é igual ao máximo dos graus de todos os seus nós.
- g) Uma árvore binária tem grau máximo igual a 2.
- h) Nó pai: nó acima e com ligação direta a outro nó.
- i) Nó filho: nó abaixo e com ligação direta a outro nó. São os nós raízes das sub-árvores.
- j) Nós irmãos: são que possuem o mesmo nó pai.
- k) Nó folha ou terminal: nó que não possui filhos.

# Graus dos nós de uma árvore binária



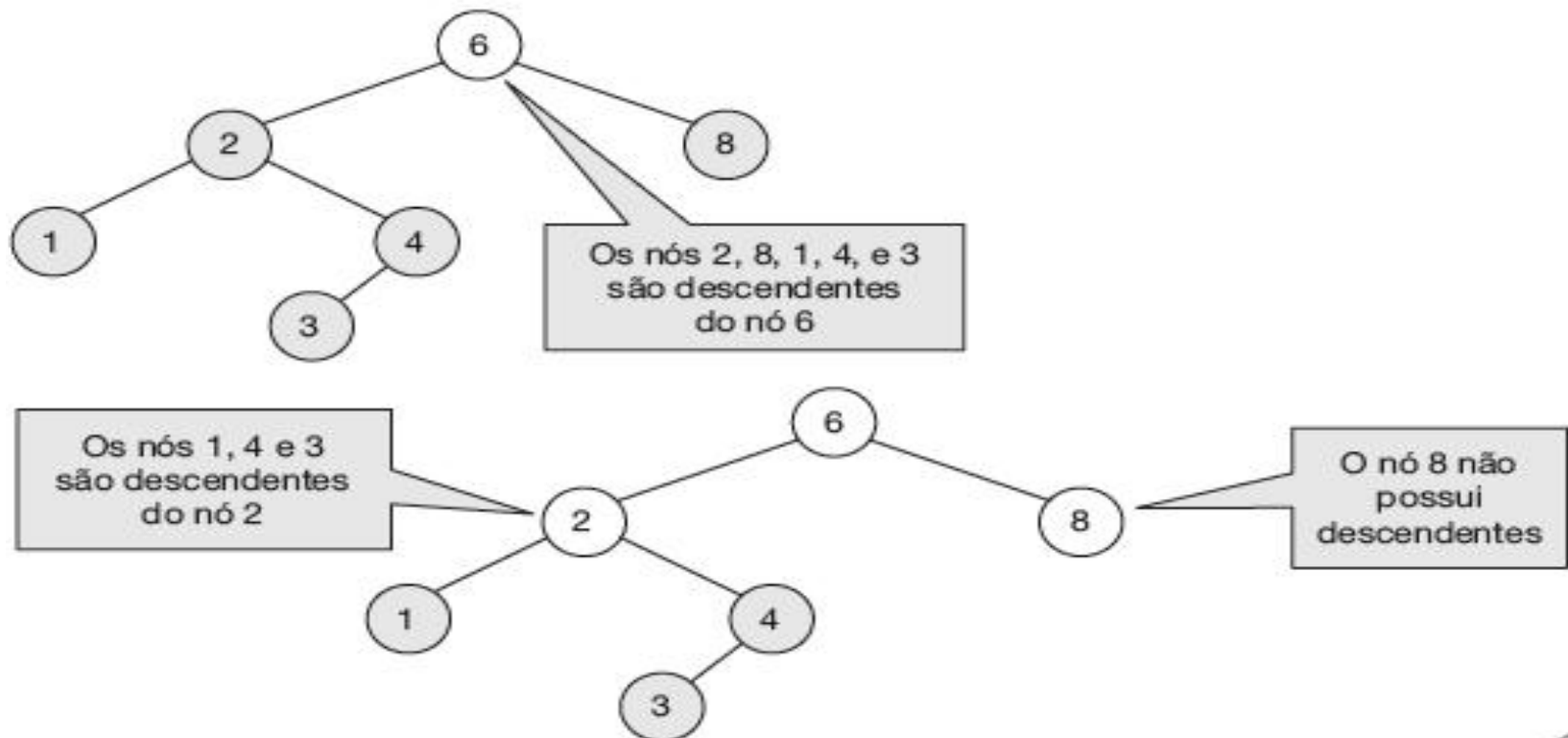
# Propriedades da árvore binária

1) Nós ancestrais: estão acima de um nó e têm ligação direta ou indireta



# Propriedades da árvore binária

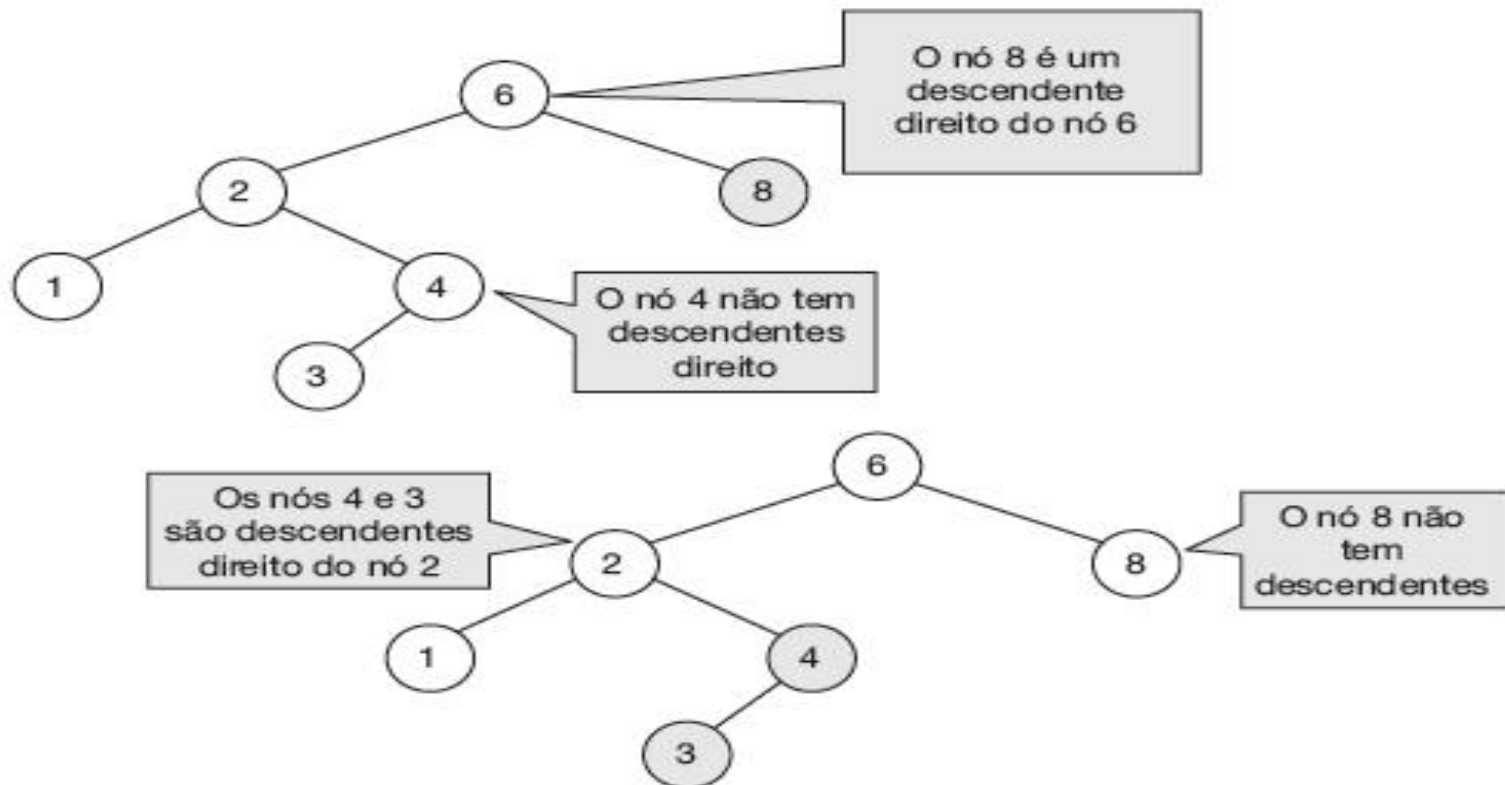
m) Nós descendentes: estão abaixo de um nó e possuem ligação direta ou indireta.





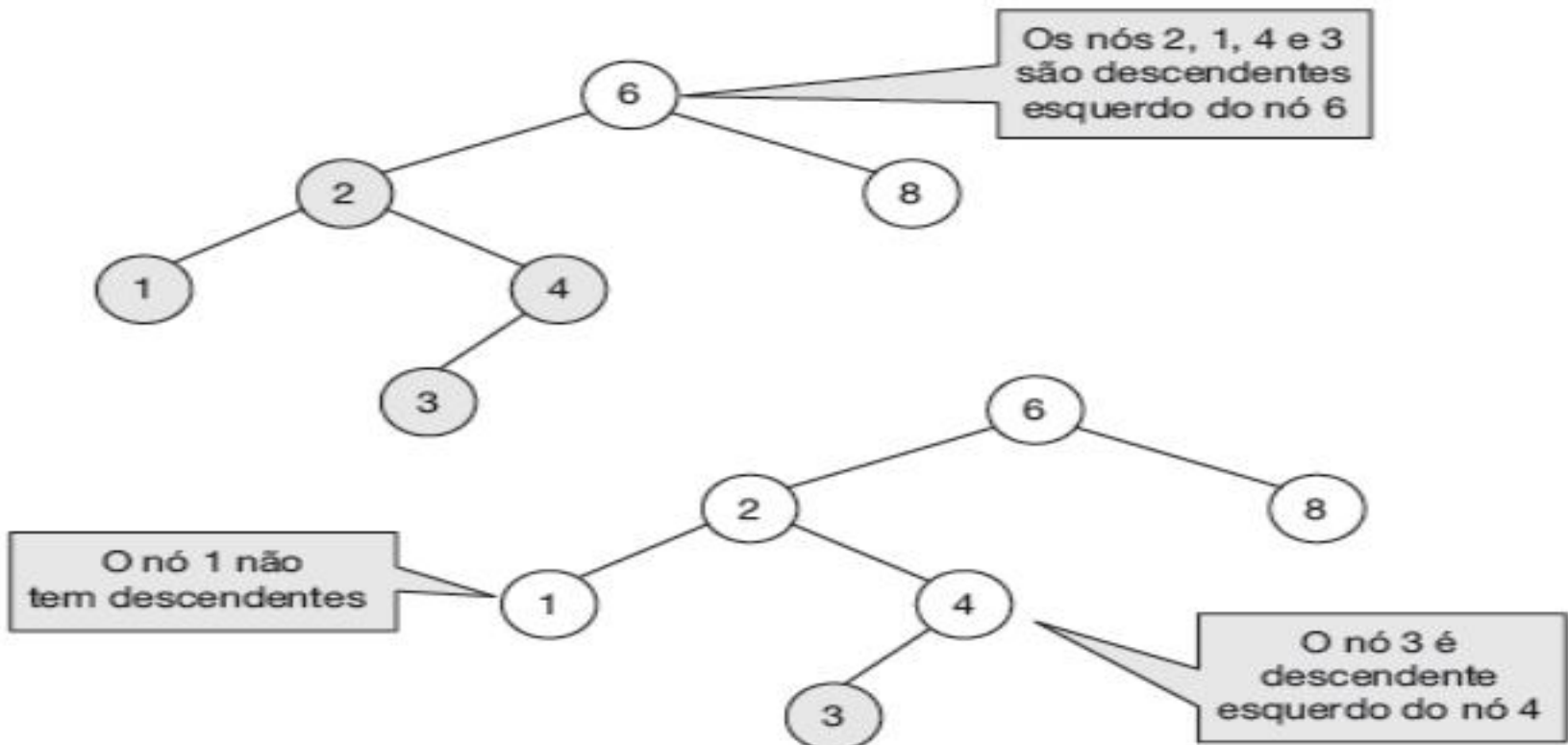
# Propriedades da árvore binária

n) Nós descendentes direito: estão abaixo de um nó, possuem ligação direta ou indireta e fazem parte da sub-árvore direita.



# Propriedades da árvore binária

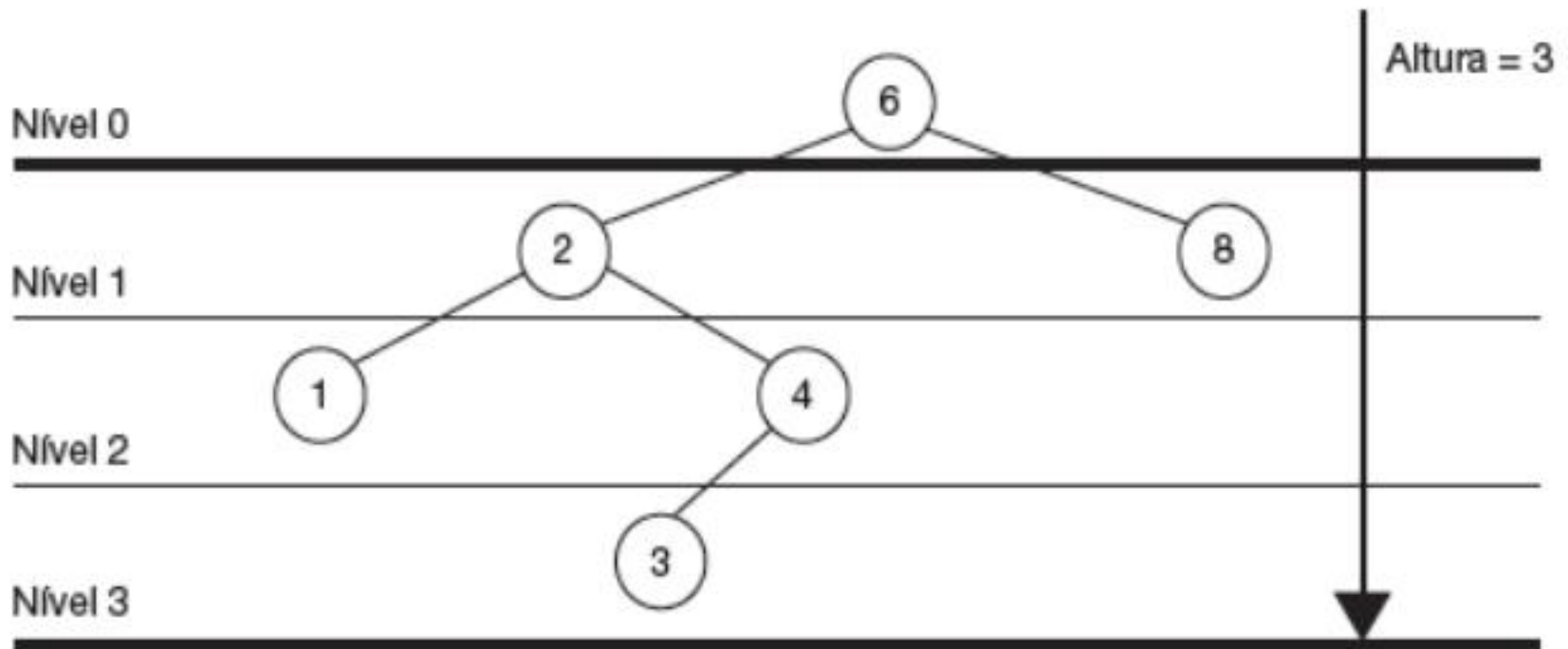
o) Nós descendentes esquerdo: estão abaixo de um nó, possuem ligação direta ou indireta e fazem parte da sub-árvore esquerda.



# Propriedades da árvore binária

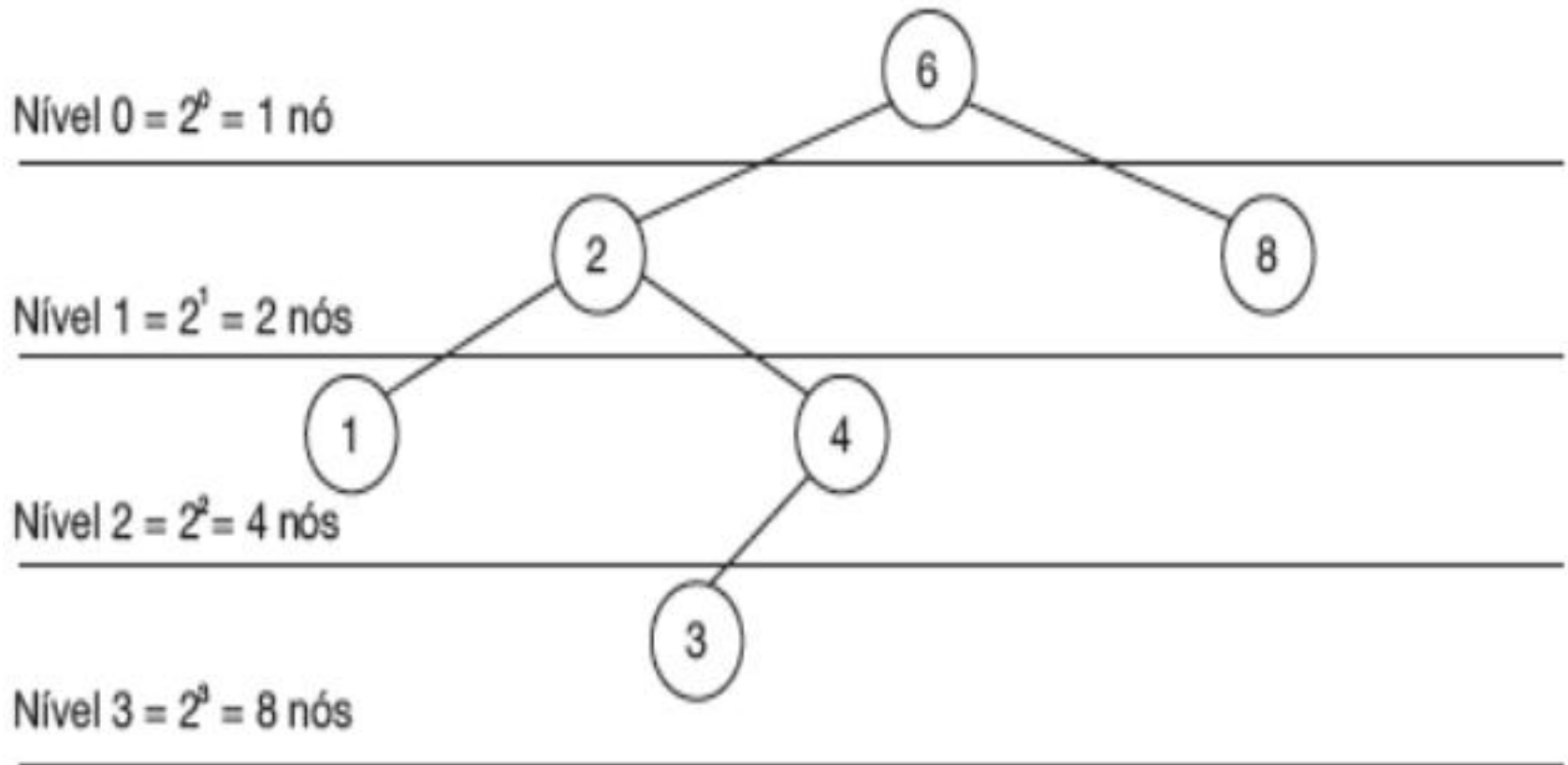
p) Nível de um nó: distância do nó raiz.

q) Altura ou profundidade da árvore: nível mais distante da raiz.



# Propriedades da árvore binária

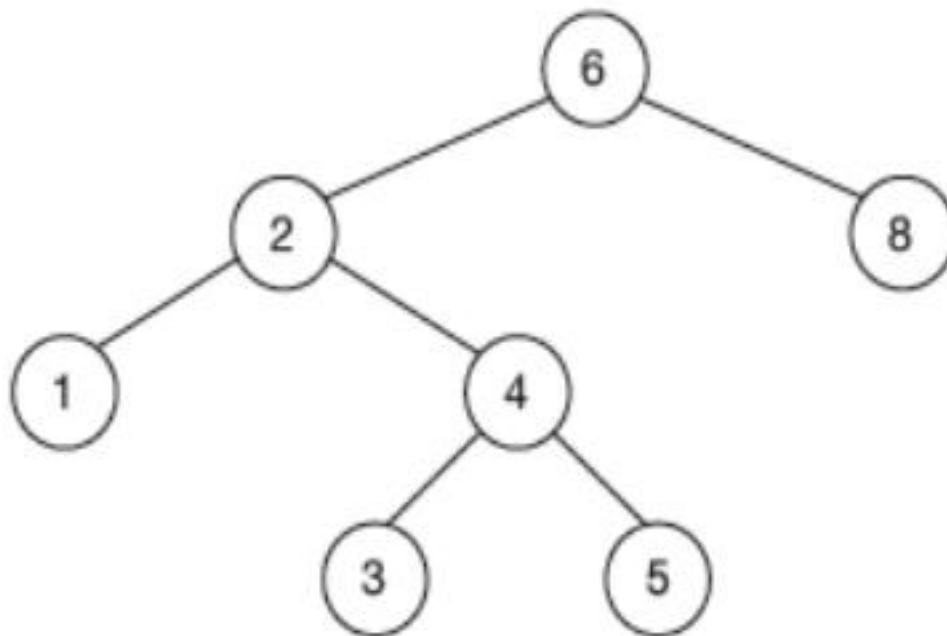
r) Expressão que representa o número máximo de nós em um nível da árvore binária =  $2^n$ , onde  $n$  é o nível em questão.



# Propriedades da árvore binária

s) Árvore estritamente binária: árvore em que todos os nós têm 0 ou 2 filhos.

t) Expressão que representa o número de nós de uma árvore estritamente binária =  $2n - 1$ , onde  $n$  é o número de nós folha.



Quantidade de nós  
folha = 4.

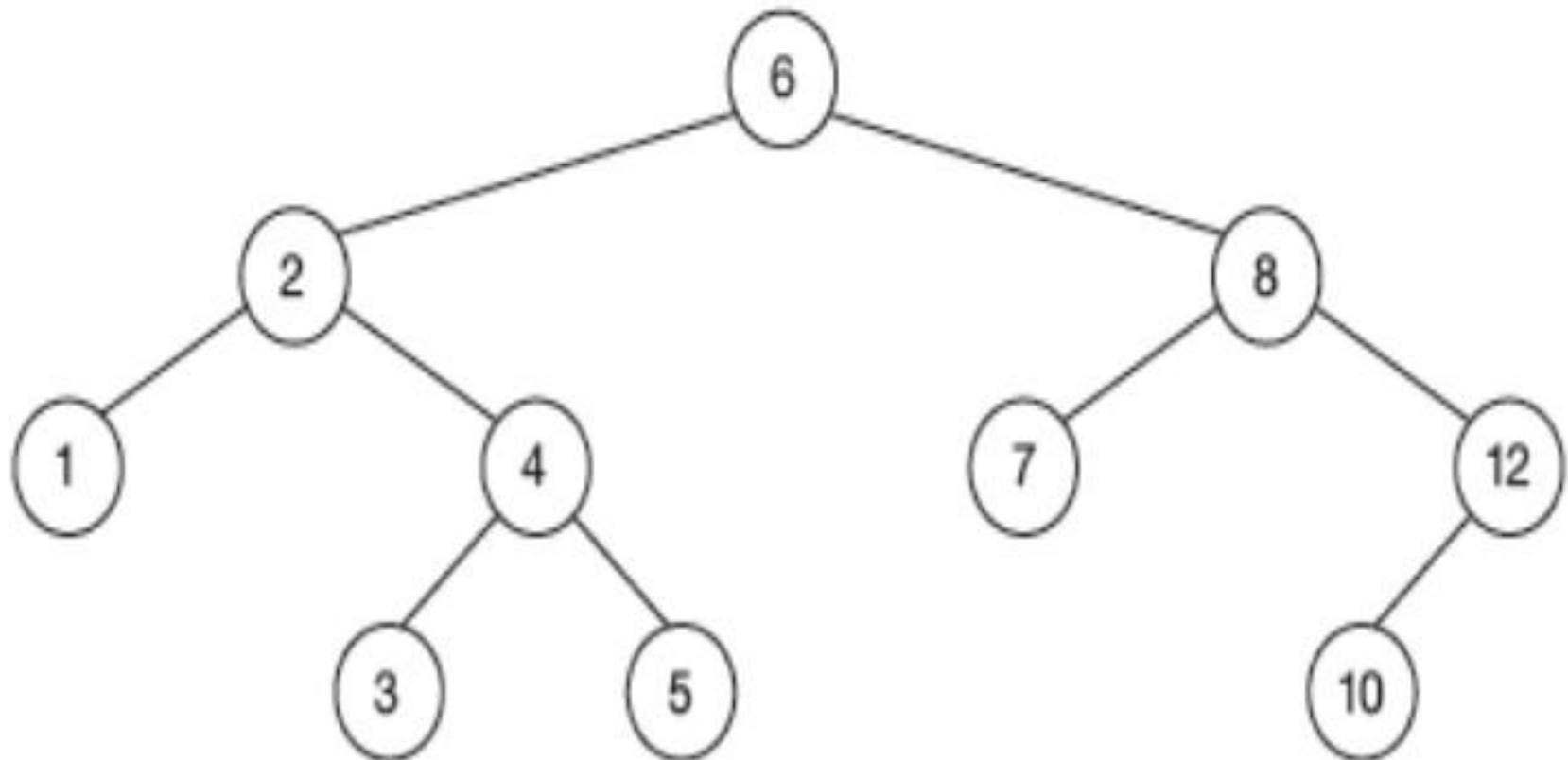
Os nós folha são:

1, 3, 5 e 8.

Número de nós desta  
árvore estritamente  
binária =  $2.n - 1$ , onde  
 $n$  é o número de folhas  
 $2.4 - 1 = 7$  nós

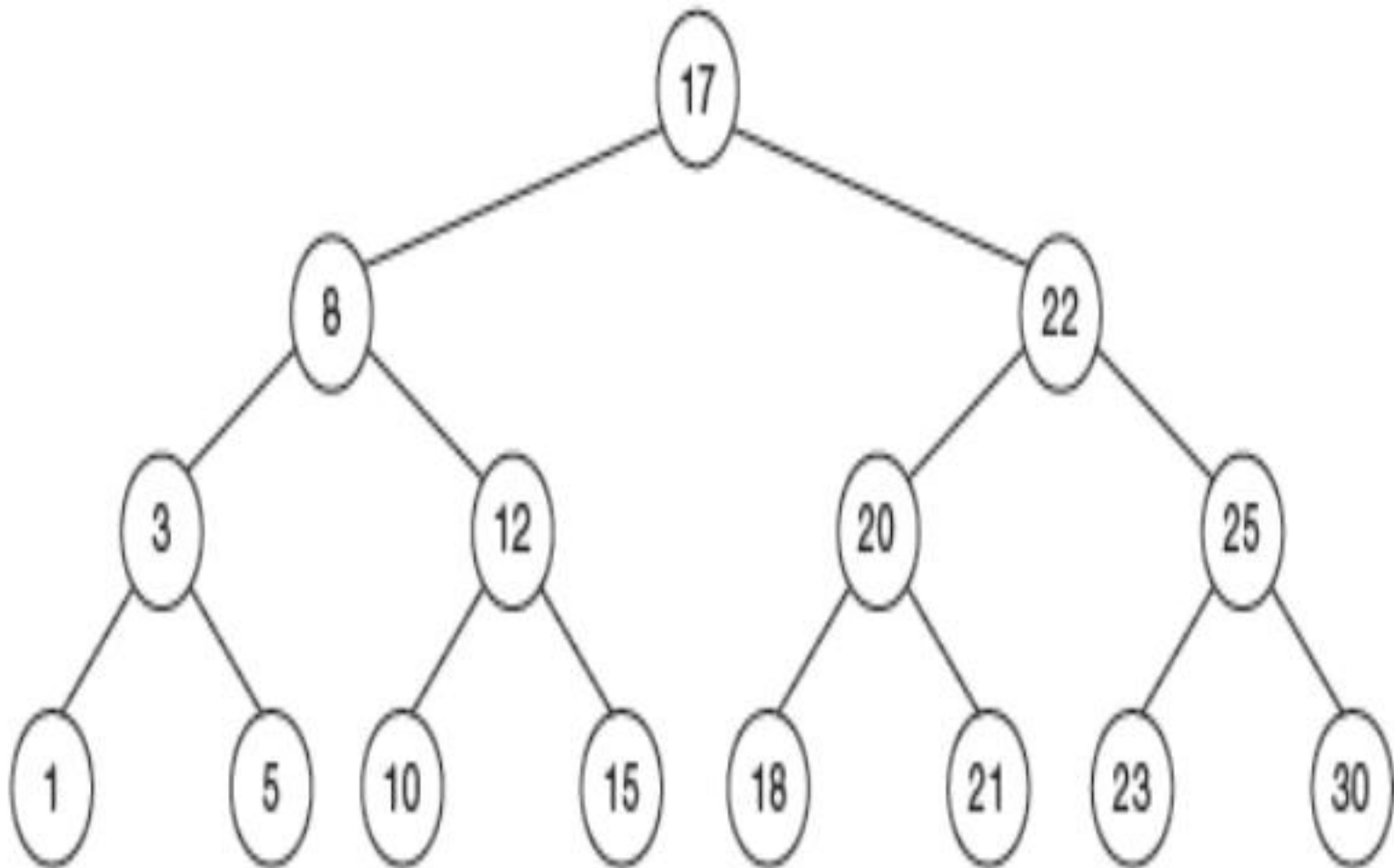
# Propriedades da árvore binária

u) Árvore completa: todos os nós com menos de dois filhos ficam no último e no penúltimo nível.



# Propriedades da árvore binária

v) Árvore cheia: árvore estritamente binária e completa.





# Árvore binária

- Na inserção, as propriedades da árvore devem ser obedecidas e todo novo nó é sempre uma folha.
- Na remoção, o filho da direita, que é o mais velho, assume o lugar do nó pai.
- Na consulta (em ordem, pré-ordem e pós-ordem), todos os nós são listados, alterando-se apenas a ordem.

# Árvore binária

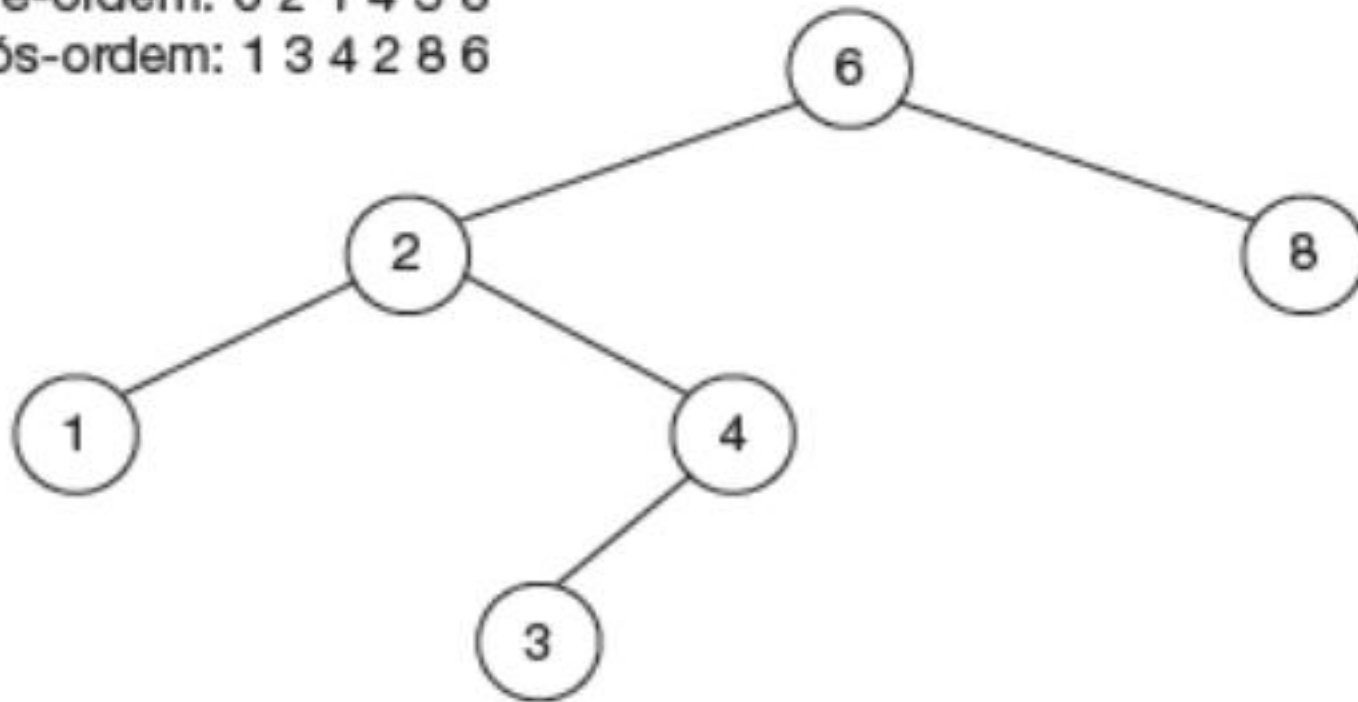
- Consulta em ordem: cada árvore é mostrada com o ramo da esquerda, a raiz e posteriormente o ramo da direita.
- Consulta pré-ordem: cada árvore é mostrada com a raiz, o ramo da esquerda e posteriormente o ramo da direita.
- Consulta pós-ordem: cada árvore é mostrada com o ramo da esquerda, o ramo da direita e posteriormente a raiz.

# Consultas em um árvore binária

Em ordem: 1 2 3 4 6 8

Pré-ordem: 6 2 1 4 3 8

Pós-ordem: 1 3 4 2 8 6



# Busca

O processo de busca em árvores é normalmente feito a partir da raiz na direção de alguma de suas folhas

Naturalmente, são de especial interesse as árvores com a menor altura possível

A altura mínima de uma árvore binária com  $n > 0$  nós é  $h = 1 + \lfloor \log_2 n \rfloor$

# Implementação

## Usando vetores

A idéia é armazenar níveis sucessivos da árvore seqüencialmente no vetor

