



# Árvores

Estruturas de Dados

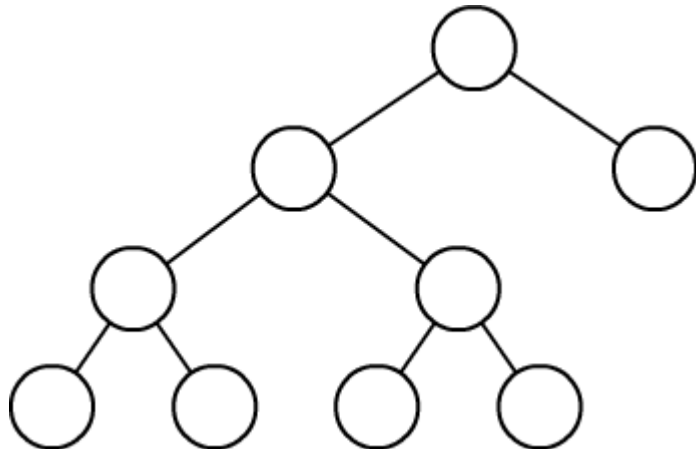
Prof. David Buzatto

---

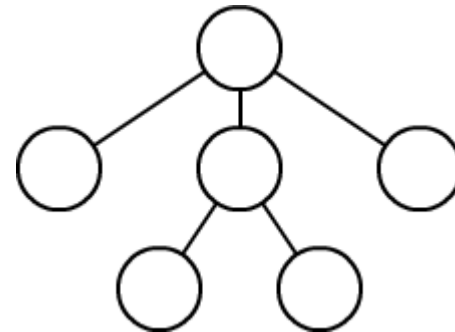
# Árvores

- Estruturas de dados não lineares;

Árvore 1



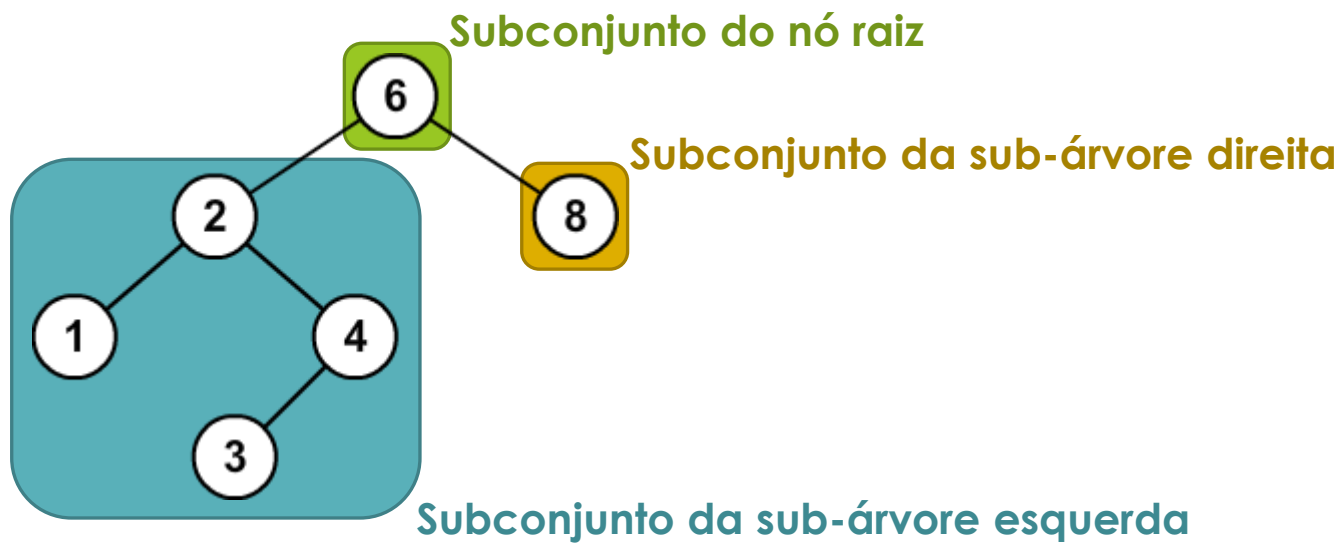
Árvore 2



# Árvore Binária

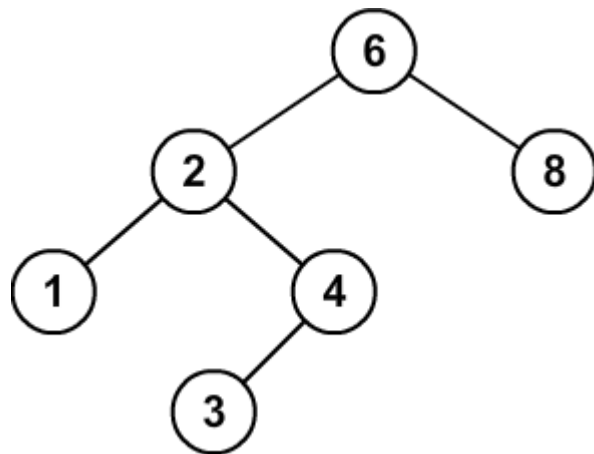
- Conjunto finito de elementos (nós);
- O primeiro nó é chamado de raiz;
- Formada por três subconjuntos:
  - Subconjunto do nó raiz;
  - Subconjunto da sub-árvore esquerda;
  - Subconjunto da sub-árvore direita;
- Implica em cada nó poder ter no máximo 2 filhos;

# Árvore Binária

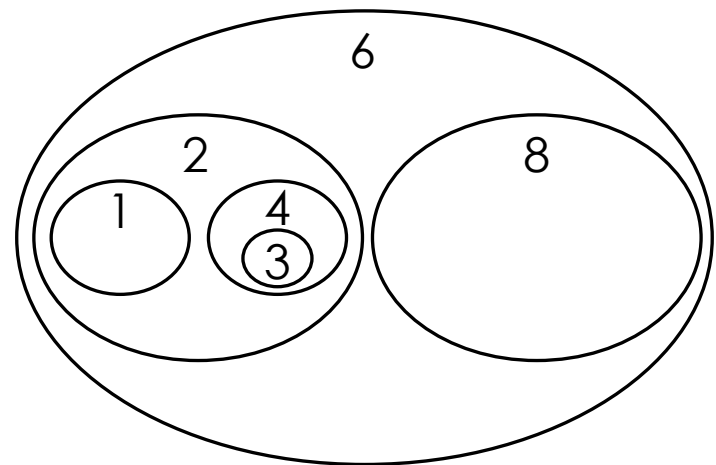


# Árvore Binária

Representação em Árvore



Representação em Conjunto



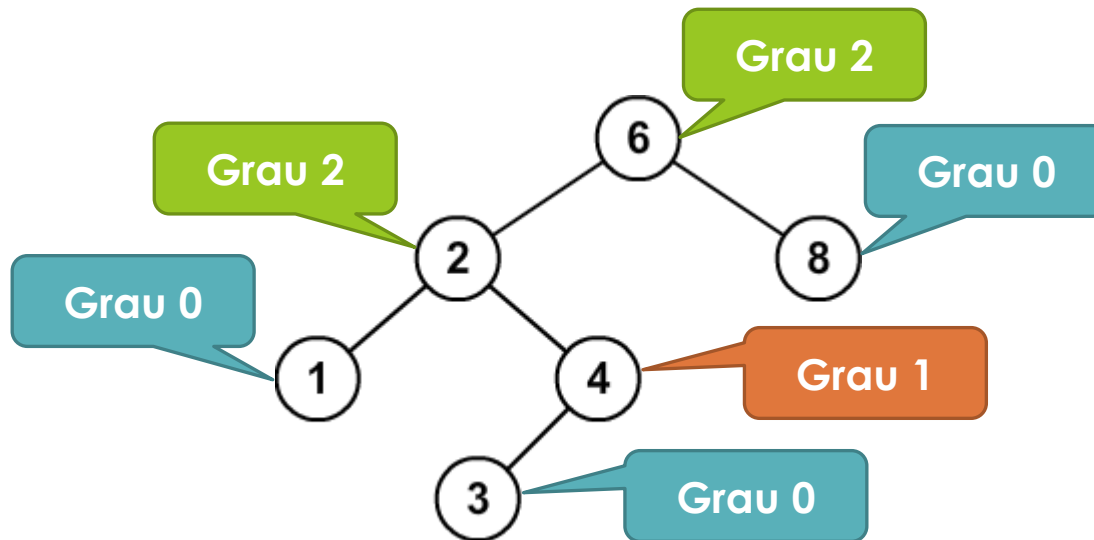
# Árvore Binária

- Árvore Binária de Busca
  - Todos os nós de uma sub-árvore direita são maiores ou iguais ao nó raiz;
  - Todos os nós de uma sub-árvore esquerda são menores que o nó raiz;
  - Cada sub-árvore é também uma árvore binária;

# Árvore Binária

## Propriedades

- O grau de um nó é igual ao seu número de sub-árvores;



# Árvore Binária

## Propriedades

- Em uma árvore binária, o grau máximo de um nó é 2;
- O grau de uma árvore é igual ao máximo dos graus de todos os seus nós;
- Uma árvore binária tem grau máximo igual a 2;



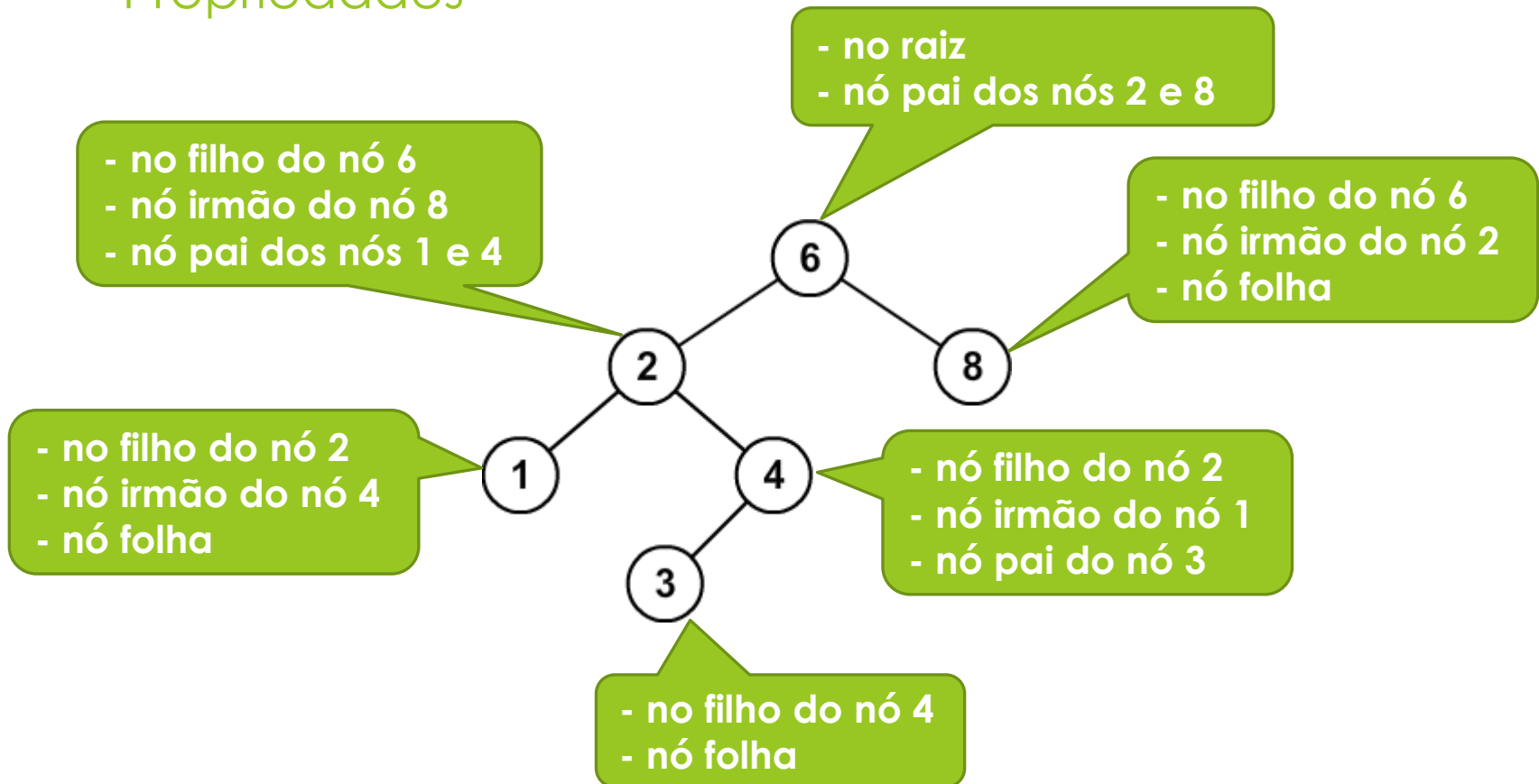
# Árvore Binária

## Propriedades

- Nó pai: nó acima e com ligação direta a outro nó;
- Nó filho: nó abaixo e com ligação direta a outro nó;
- Nós irmãos: nós que possuem o mesmo pai;
- Nó folha ou terminal: nó que não possui filhos;

# Árvore Binária

## Propriedades



# Árvore Binária

## Propriedades

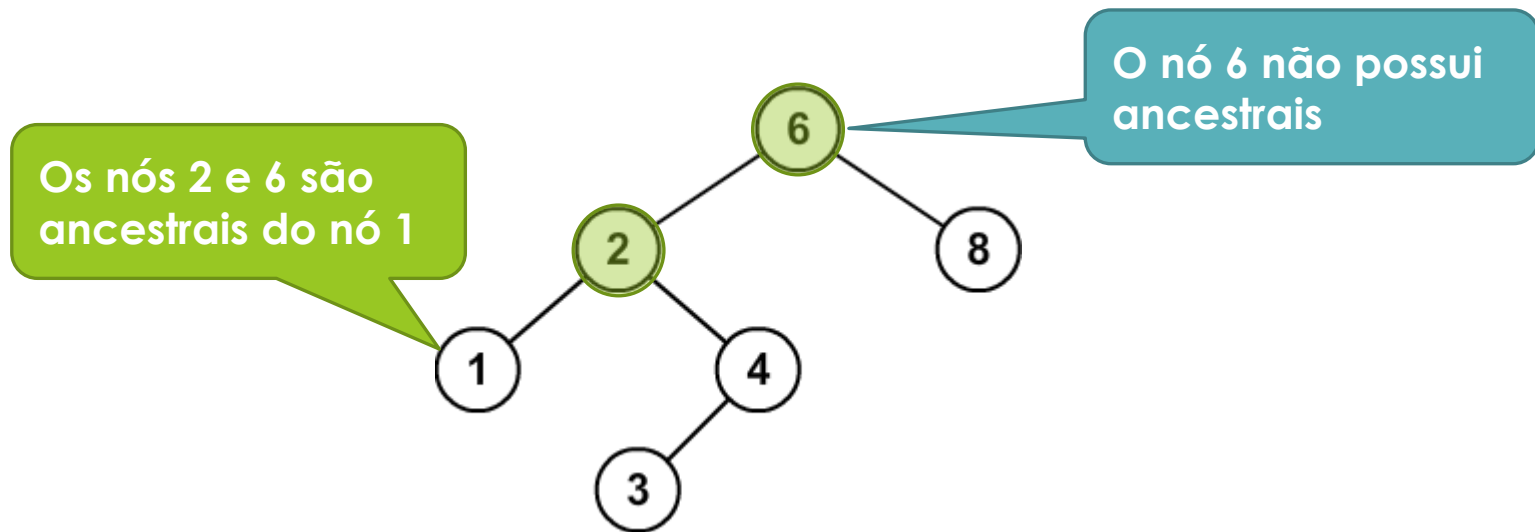
- Nó ancestral: nós que estão acima de um nó e possuem ligação direta ou indireta;



# Árvore Binária

## Propriedades

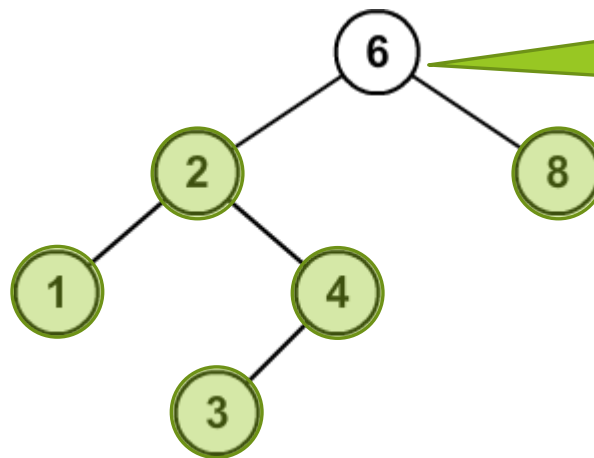
- Nó ancestral: nós que estão acima de um nó e possuem ligação direta ou indireta;



# Árvore Binária

## Propriedades

- **Nó descendente:** nós que estão abaixo de um nó e possuem ligação direta ou indireta.



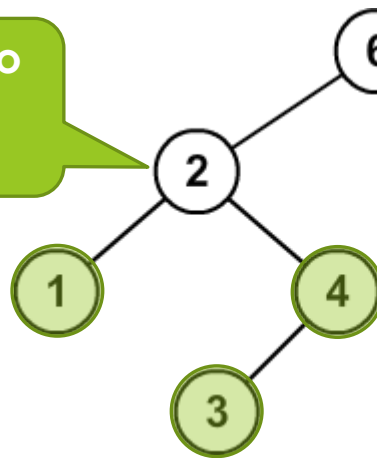
Os nós 2, 8, 1, 4 e 3  
são descendentes  
do nó 6

# Árvore Binária

## Propriedades

- **Nó descendente:** nós que estão abaixo de um nó e possuem ligação direta ou indireta.

Os nós 1, 4 e 3 são descendentes do nó 2

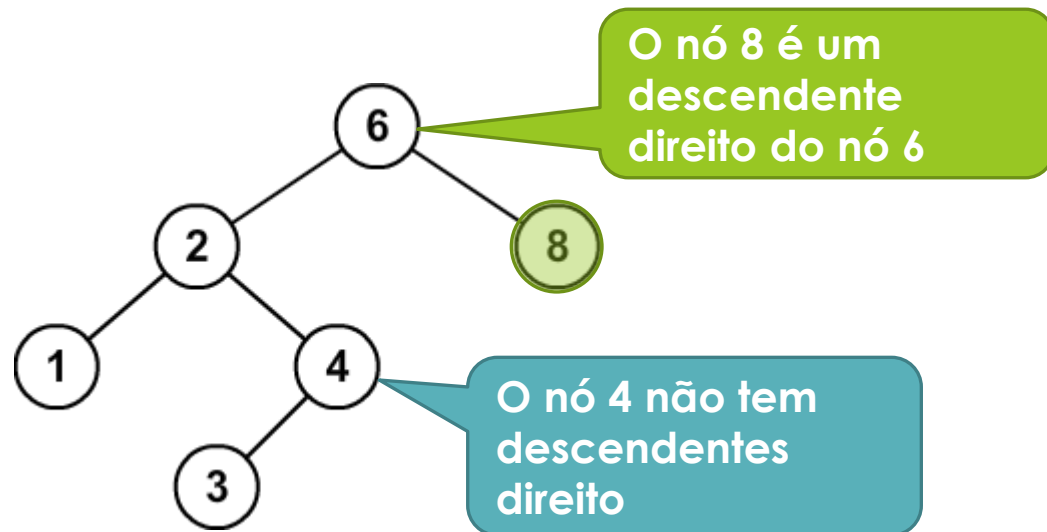


O nó 8 não possui descendentes

# Árvore Binária

## Propriedades

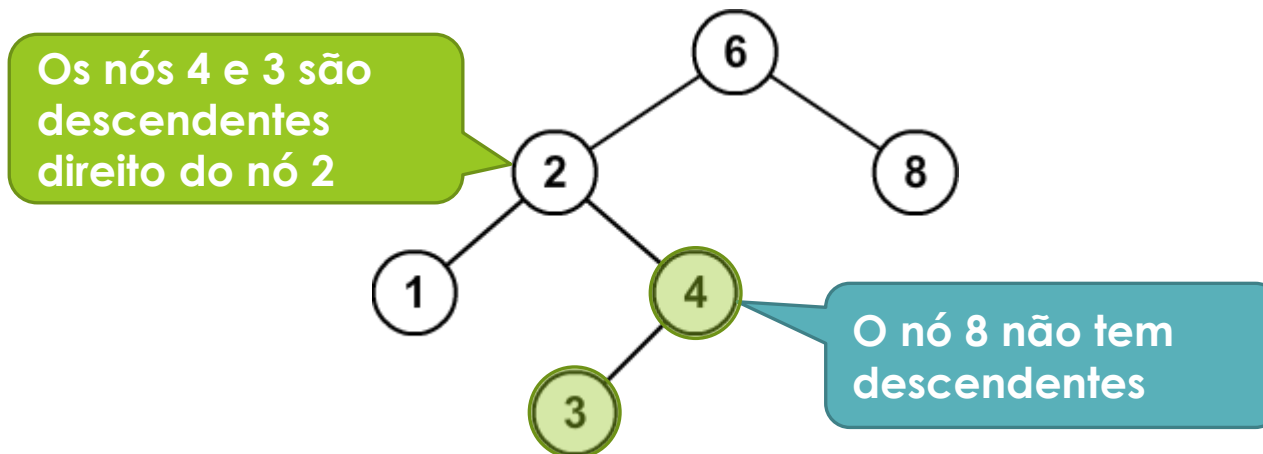
- Nó descendente direito: nós que estão abaixo de um nó, possuem ligação direta ou indireta e fazem parte da sub-árvore direita;



# Árvore Binária

## Propriedades

- Nó descendente direito: nós que estão abaixo de um nó, possuem ligação direta ou indireta e fazem parte da sub-árvore direita;

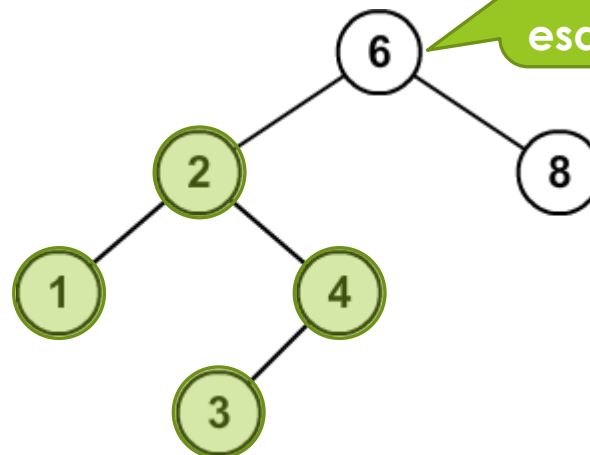




# Árvore Binária

## Propriedades

- Nó descendente esquerdo: nós que estão abaixo de um nó, possuem ligação direta ou indireta e fazem parte da sub-árvore esquerda;



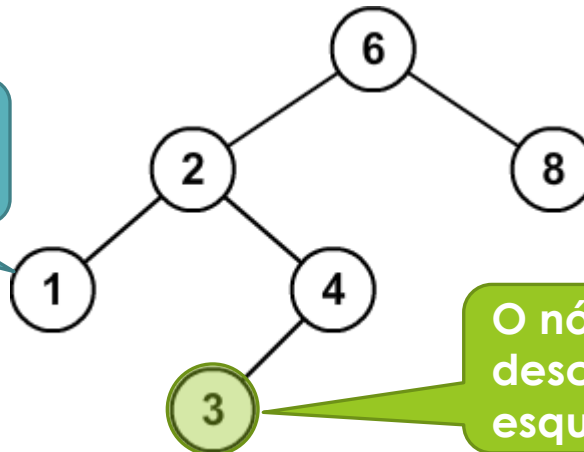
Os nós 2, 1, 4 e 3  
são descendentes  
esquerdo do nó 6

# Árvore Binária

## Propriedades

- Nó descendente esquerdo: nós que estão abaixo de um nó, possuem ligação direta ou indireta e fazem parte da sub-árvore esquerda;

O nó 1 não tem descendentes

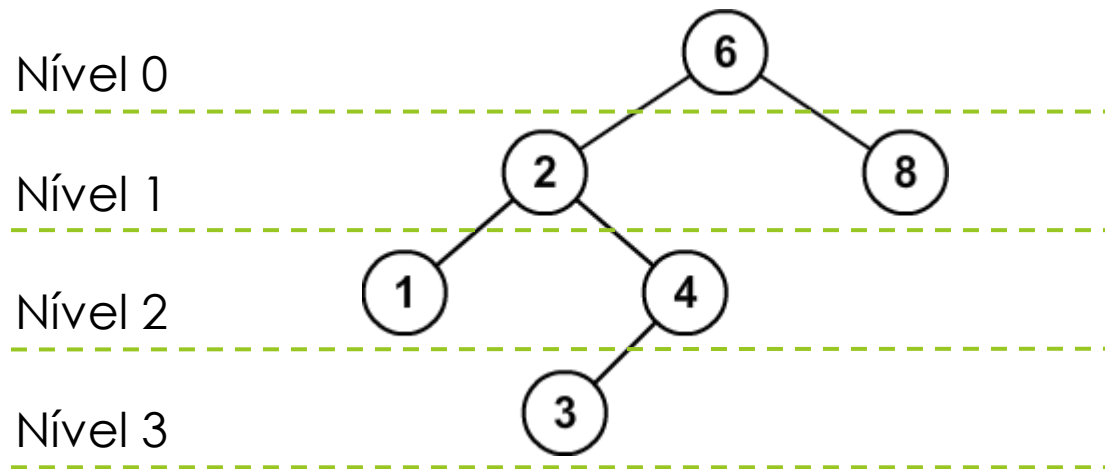


O nó 3 é descendente esquerdo do nó 4

# Árvore Binária

## Propriedades

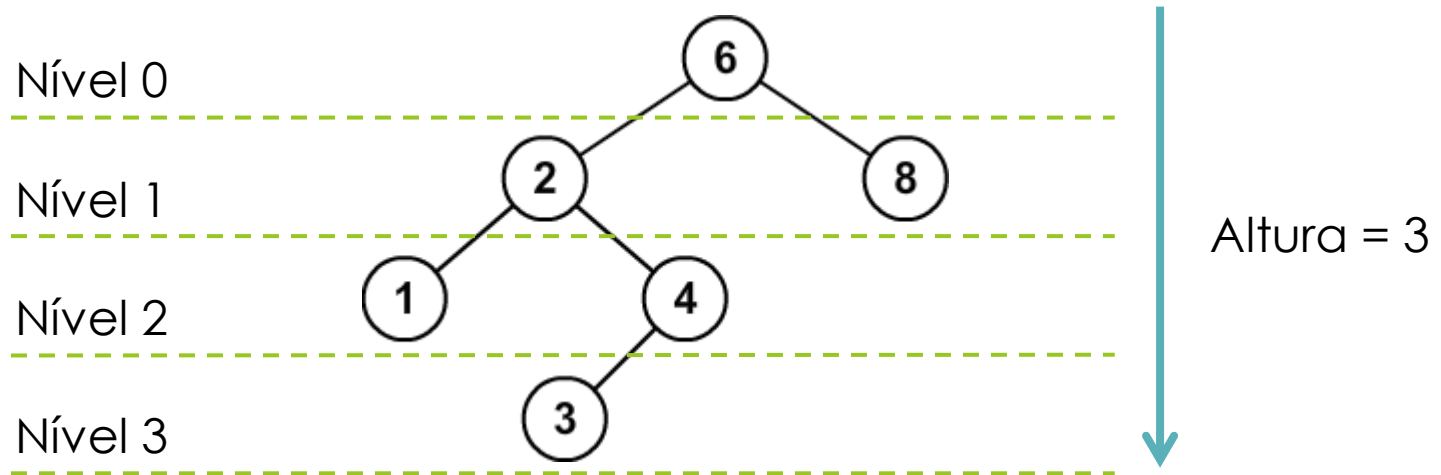
- Nível de um nó: é a sua distância em relação ao nó raiz. O nível da raiz é sempre zero;



# Árvore Binária

## Propriedades

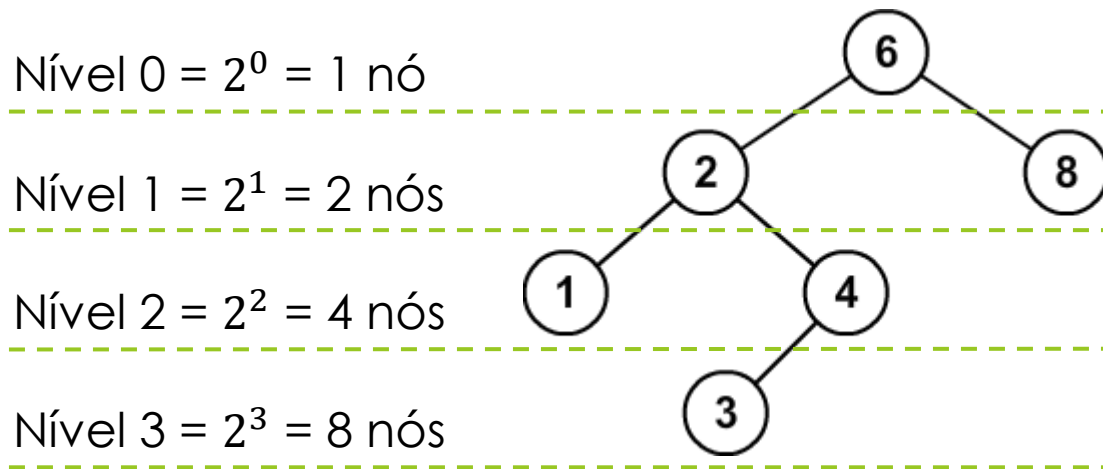
- Altura ou profundidade de uma árvore: é o nível do nó mais distante da raiz;



# Árvore Binária

## Propriedades

- Número máximo de nós em um nível:  
 $2^n$ , onde  $n$  é o nível em questão

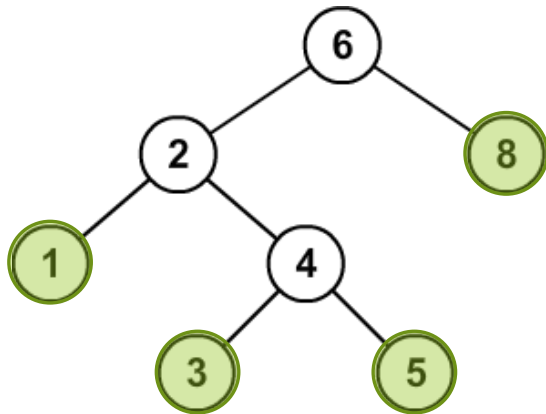


# Árvore Binária

## Propriedades

- Árvore estritamente binária: árvore em que todos os nós tem 0 ou 2 filhos;
- Número de nós de uma árvore estritamente binária:

$2n - 1$ , onde  $n$  é a quantidade de nós folha



Quantidade de nós folha: 4

Os nós folha são: 1, 3, 5, e 8

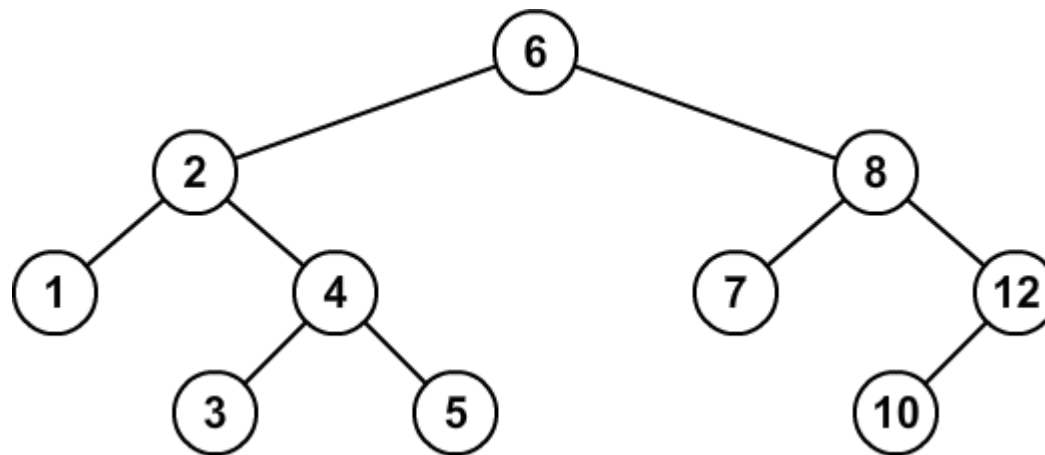
Número de nós da árvore estritamente binária:

$$2n - 1 = 2 \cdot 4 - 1 = 8 - 1 = 7$$

# Árvore Binária

## Propriedades

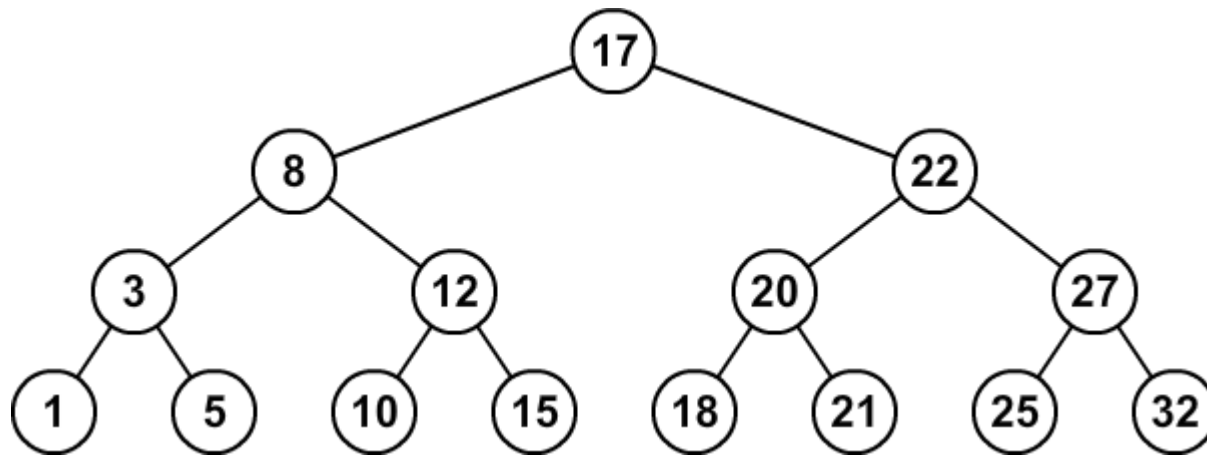
- Árvore completa: árvore em que todos os nós com menos de dois nós filhos ficam no último e no penúltimo nível;



# Árvore Binária

## Propriedades

- Árvore cheia: árvore estritamente binária e completa;





# Árvore Binária de Busca

## Inserção

- Relembrando...
  - Todos os nós de uma sub-árvore direita são maiores ou iguais ao nó raiz;
  - Todos os nós de uma sub-árvore esquerda são menores que o nó raiz;

# Árvore Binária de Busca

Inserção

raiz  
↓

# Árvore Binária de Busca

## Inserção

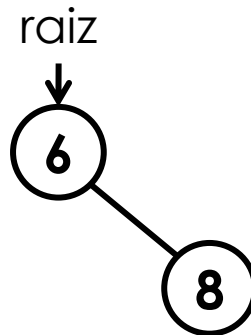
- inserir( 6 )



# Árvore Binária de Busca

## Inserção

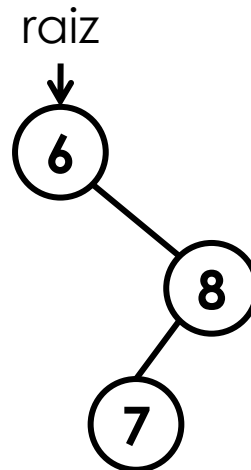
- inserir( 8 )



# Árvore Binária de Busca

## Inserção

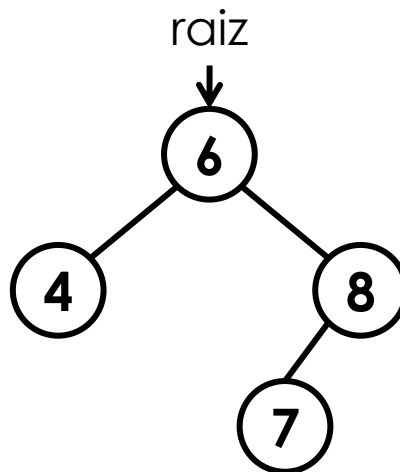
● inserir( 7 )



# Árvore Binária de Busca

## Inserção

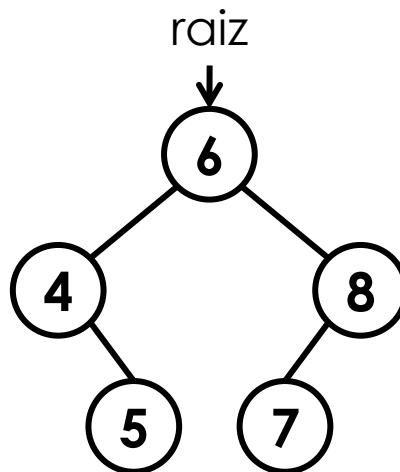
● inserir( 4 )



# Árvore Binária de Busca

## Inserção

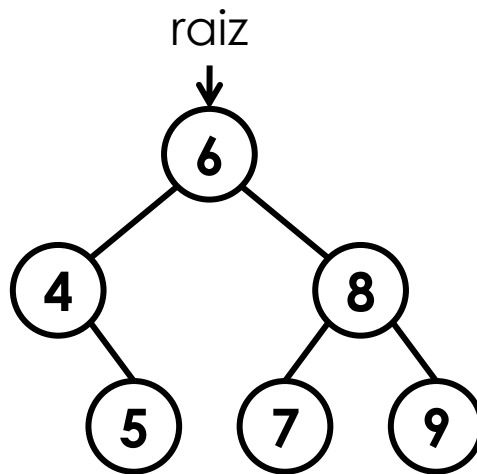
● inserir( 5 )



# Árvore Binária de Busca

## Inserção

● inserir( 9 )

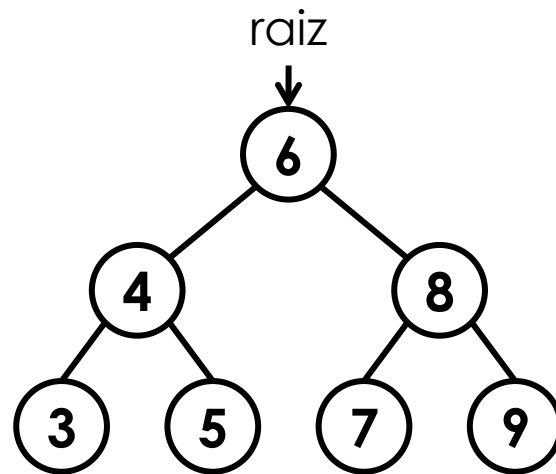




# Árvore Binária de Busca

## Inserção

- inserir( 3 )



# Árvore Binária de Busca

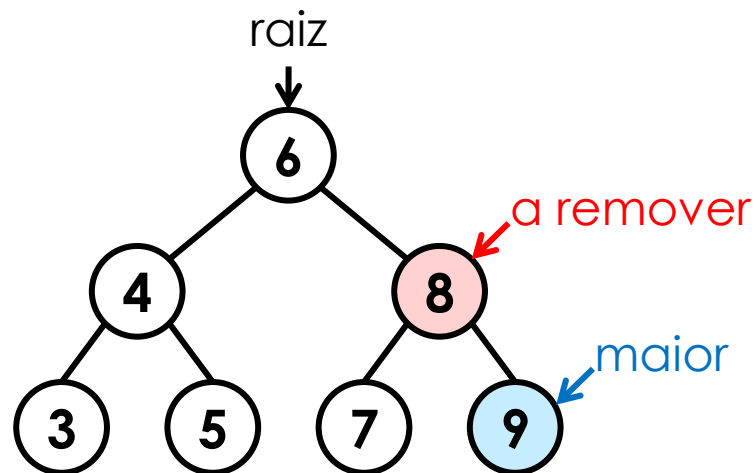
## Remoção

- De novo...
  - Todos os nós de uma sub-árvore direita são maiores ou iguais ao nó raiz;
  - Todos os nós de uma sub-árvore esquerda são menores que o nó raiz;
  - Sendo assim:
    - A remoção de um nó com filhos faz com que o nó maior que ele (à sua direita) ocupe sua posição;
    - Se o nó à direita não existe, então o nó à esquerda ocupa sua posição;
    - Caso o nó seja um nó folha, não há a necessidade de reestruturação da árvore;

# Árvore Binária de Busca

## Remoção

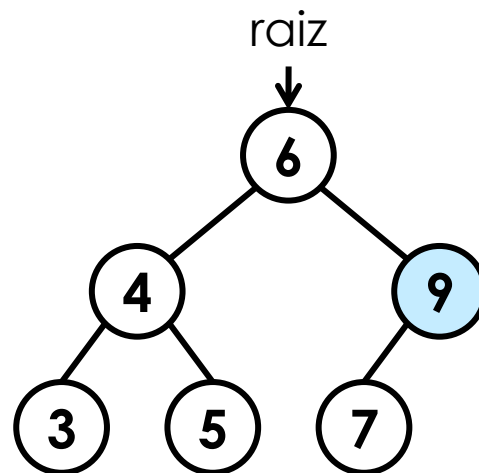
● remover( 8 )



# Árvore Binária de Busca

## Remoção

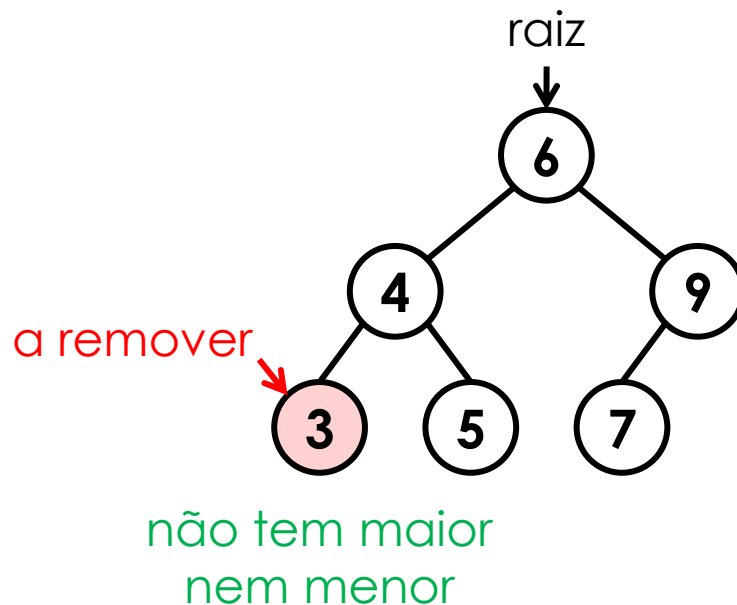
● remover( 8 )



# Árvore Binária de Busca

## Remoção

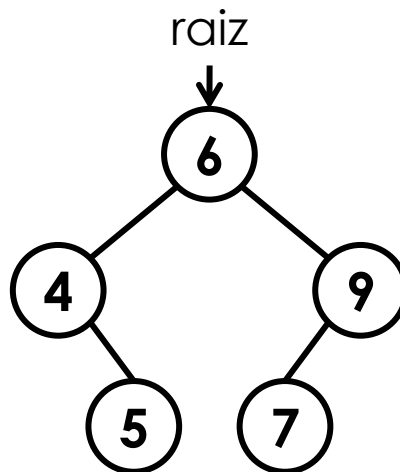
● remover( 3 )



# Árvore Binária de Busca

## Remoção

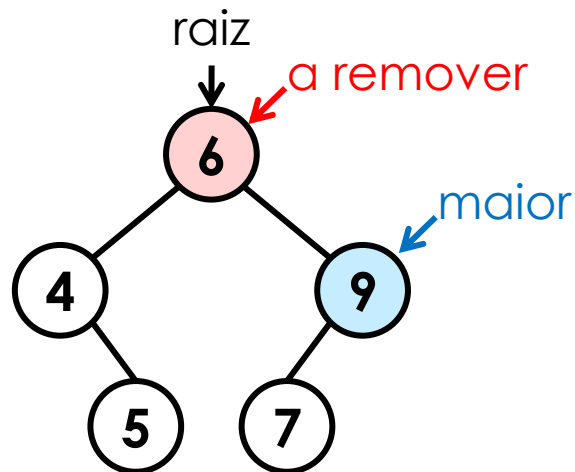
● `remove( 3 )`



# Árvore Binária de Busca

## Remoção

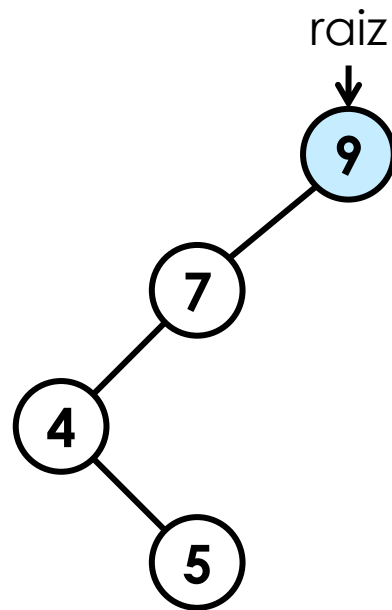
● remover( 6 )



# Árvore Binária de Busca

## Remoção

● remover( 6 )

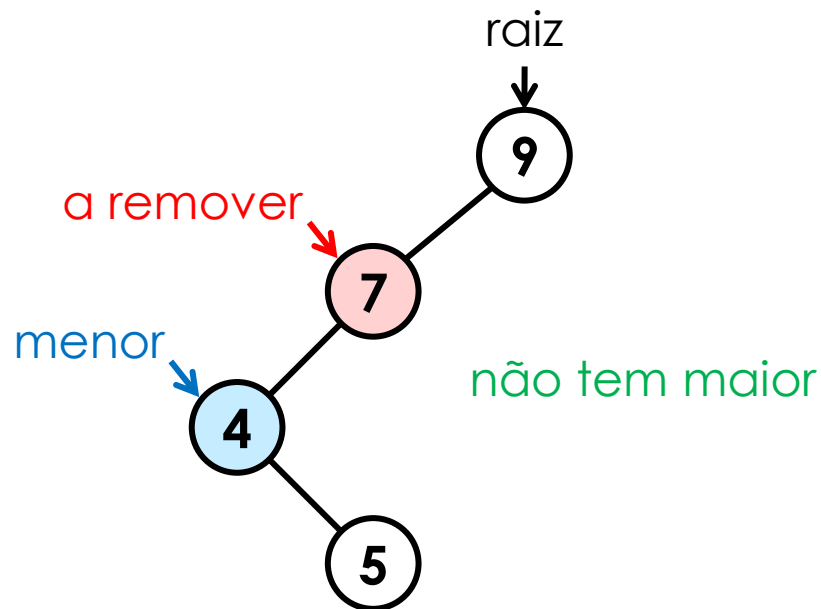




# Árvore Binária de Busca

## Remoção

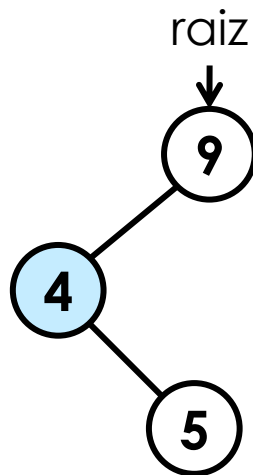
● remover( 7 )



# Árvore Binária de Busca

## Remoção

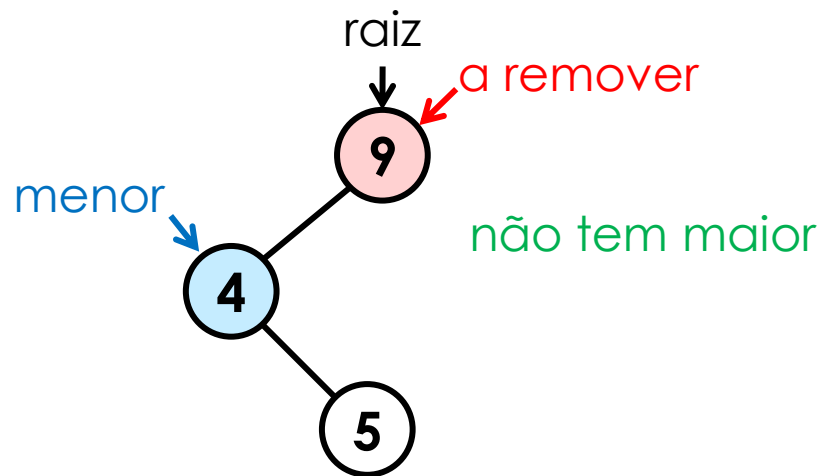
● `remove( 7 )`



# Árvore Binária de Busca

## Remoção

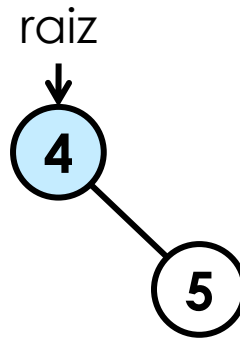
● remover( 9 )



# Árvore Binária de Busca

## Remoção

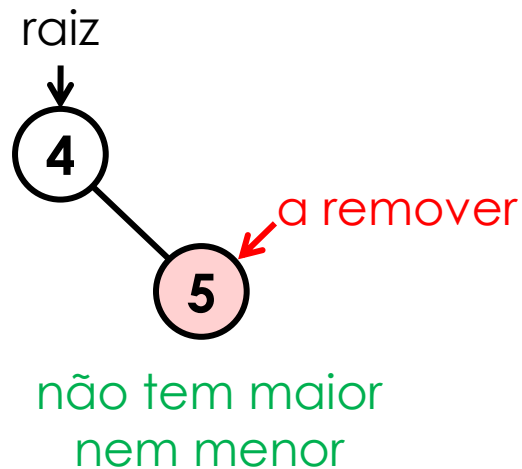
● `remove( 9 )`



# Árvore Binária de Busca

## Remoção

● remover( 5 )



# Árvore Binária de Busca

Remoção

- `remove( 5 )`



# Árvore Binária de Busca

## Remoção

● remover( 4 )



# Árvore Binária de Busca

Remoção

- `remover( 4 )`

raiz  
↓



# Árvore Binária

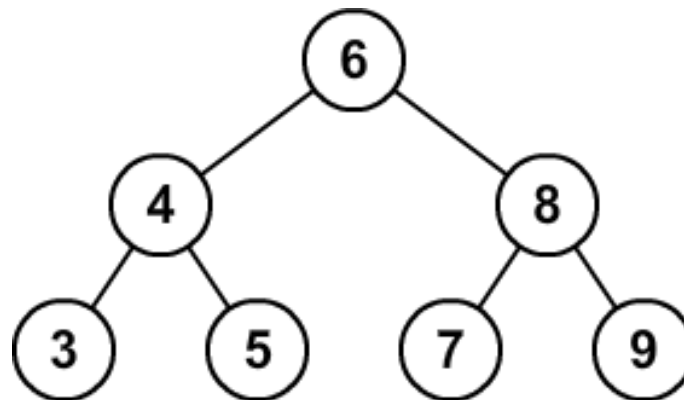
## Percursos

- Percurso: forma de percorrer uma estrutura de dados, no caso, em uma árvore;
- Tipos de percurso:
  1. Pré-ordem;
  2. Em ordem (ordem simétrica);
  3. Pós-ordem (ordem final);

# Árvore Binária

Percursos – Pré-ordem

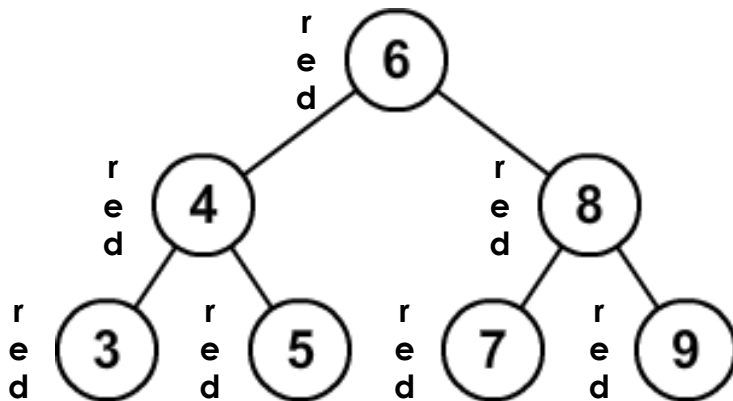
● Raiz → Esquerda → Direita



# Árvore Binária

## Percursos – Pré-ordem

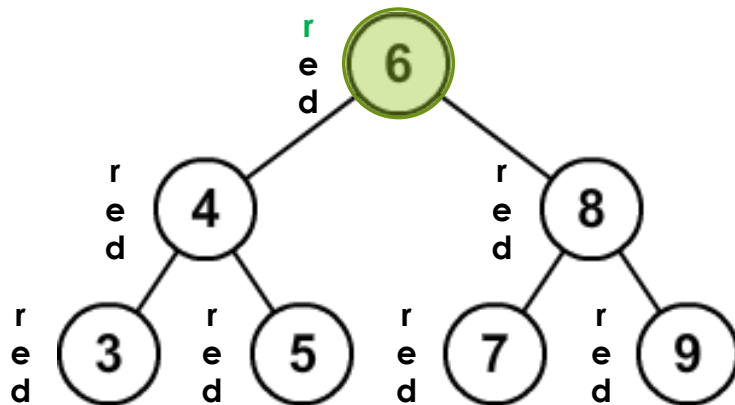
● Raiz → Esquerda → Direita



# Árvore Binária

## Percursos – Pré-ordem

● Raiz → Esquerda → Direita

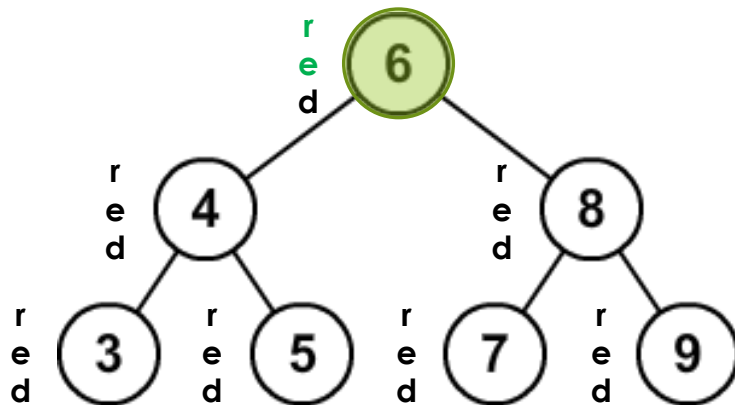


6

# Árvore Binária

## Percursos – Pré-ordem

● Raiz → Esquerda → Direita

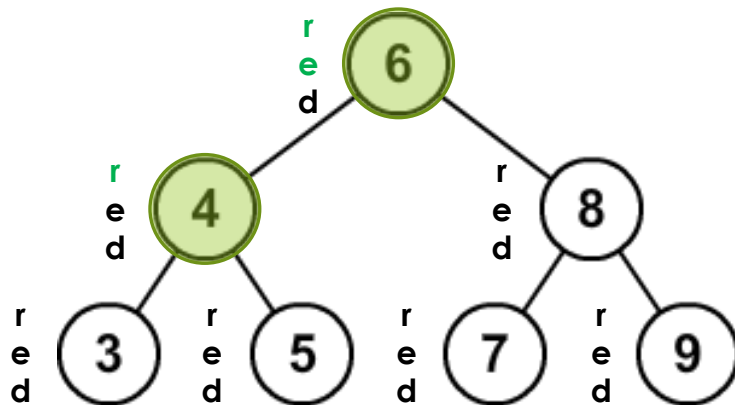


6

# Árvore Binária

## Percursos – Pré-ordem

● Raiz → Esquerda → Direita

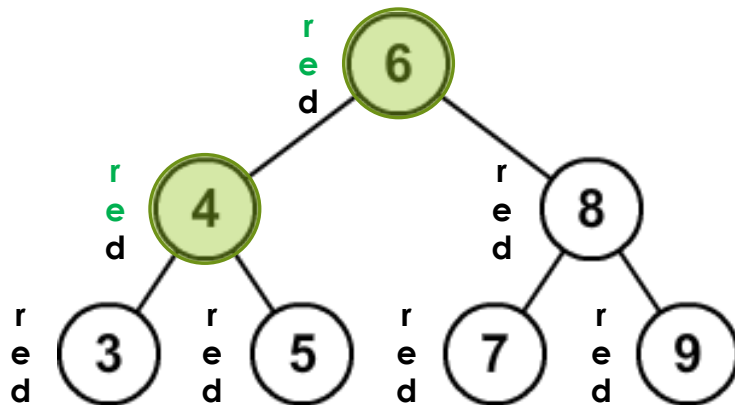


6 4

# Árvore Binária

## Percursos – Pré-ordem

● Raiz → Esquerda → Direita

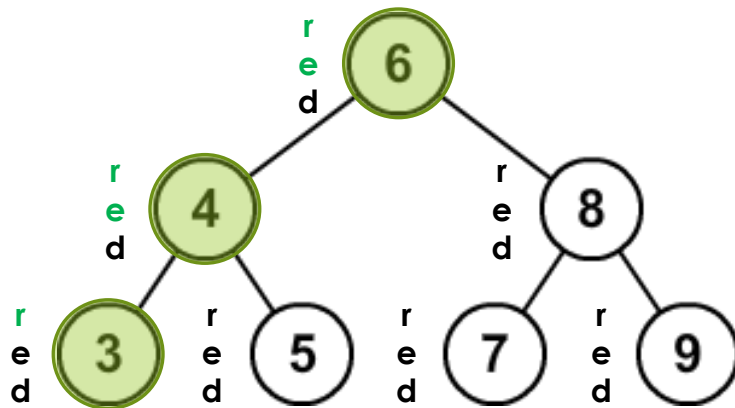


6 4

# Árvore Binária

## Percursos – Pré-ordem

● Raiz → Esquerda → Direita



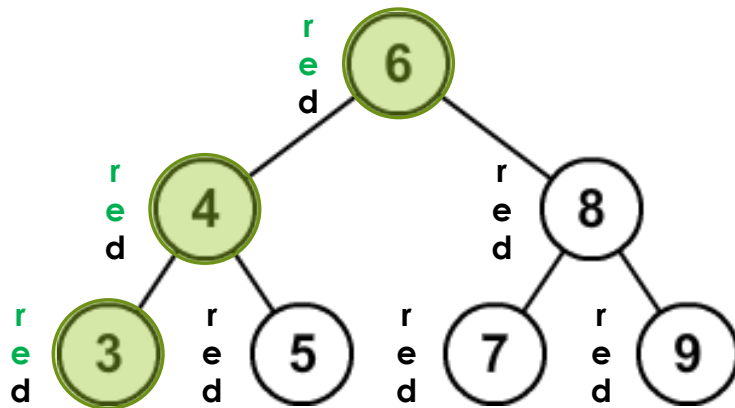
6 4 3



# Árvore Binária

## Percursos – Pré-ordem

● Raiz → Esquerda → Direita

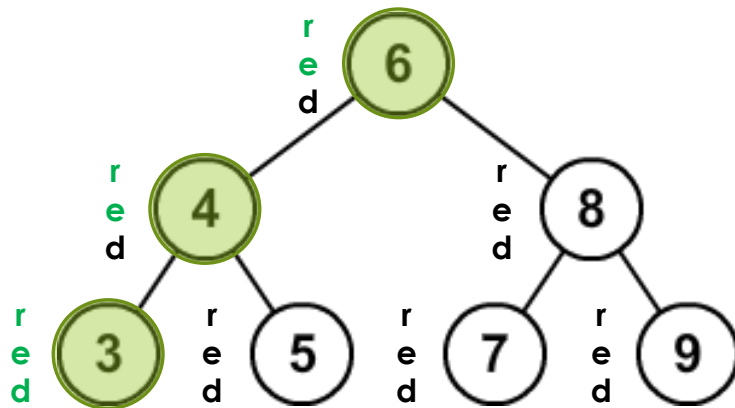


6 4 3

# Árvore Binária

## Percursos – Pré-ordem

● Raiz → Esquerda → Direita

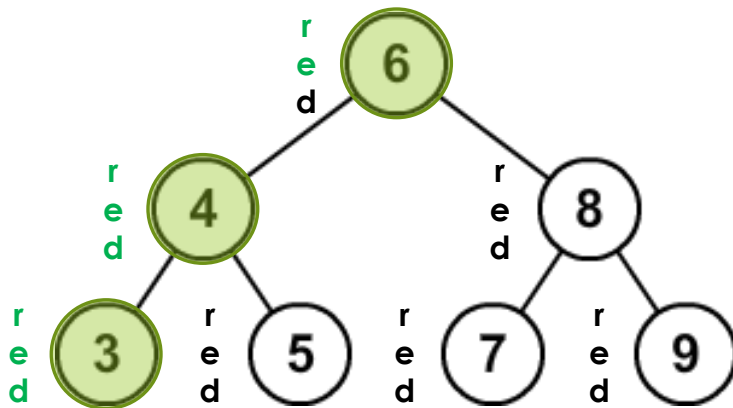


6 4 3

# Árvore Binária

## Percursos – Pré-ordem

● Raiz → Esquerda → Direita

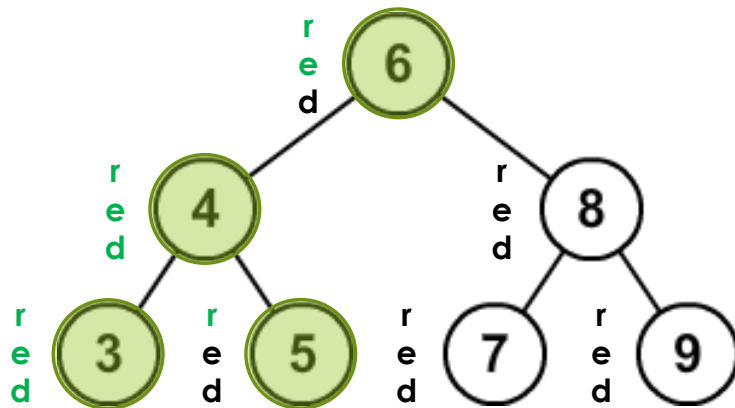


6 4 3

# Árvore Binária

## Percursos – Pré-ordem

● Raiz → Esquerda → Direita

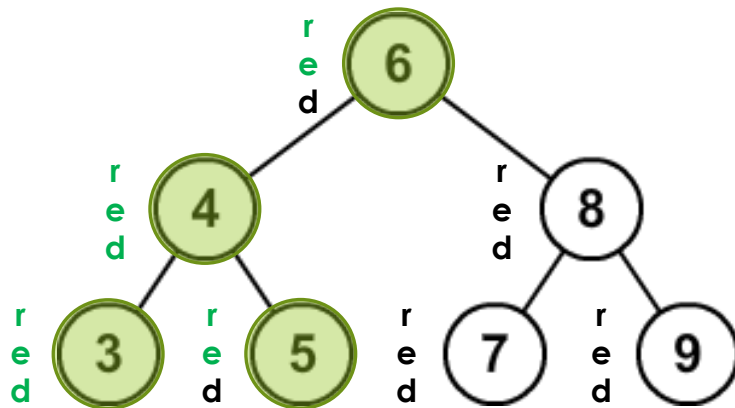


6 4 3 5

# Árvore Binária

## Percursos – Pré-ordem

● Raiz → Esquerda → Direita

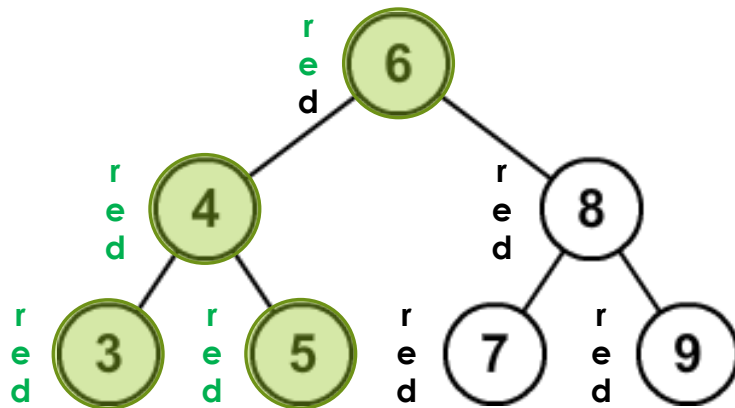


6 4 3 5

# Árvore Binária

## Percursos – Pré-ordem

● Raiz → Esquerda → Direita

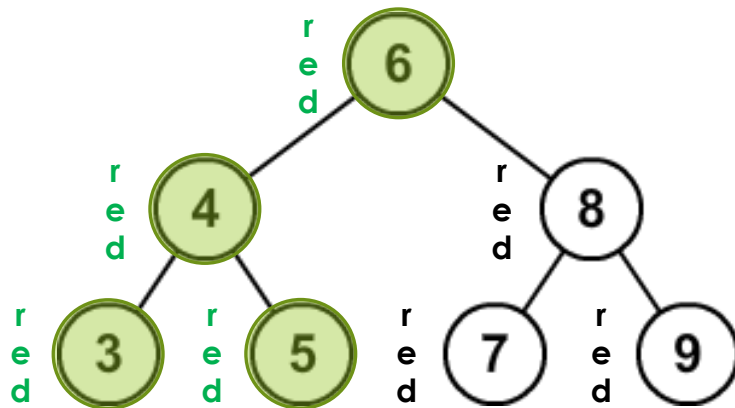


**6 4 3 5**

# Árvore Binária

## Percursos – Pré-ordem

● Raiz → Esquerda → Direita

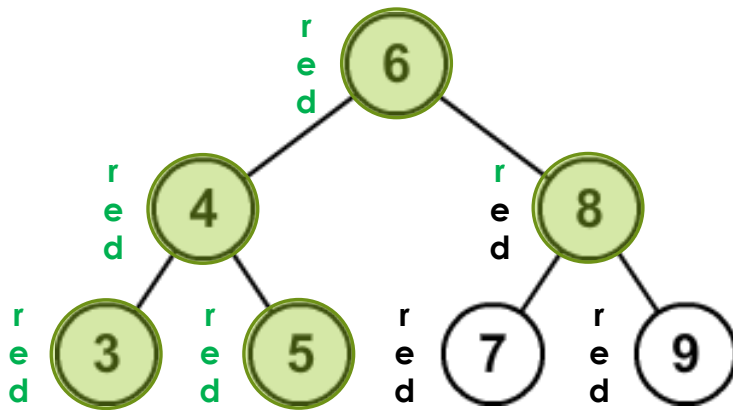


**6 4 3 5**

# Árvore Binária

## Percursos – Pré-ordem

● Raiz → Esquerda → Direita



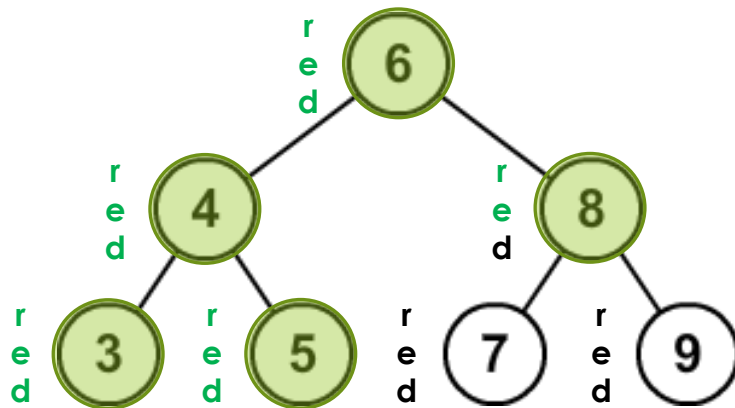
6 4 3 5 8



# Árvore Binária

## Percursos – Pré-ordem

● Raiz → Esquerda → Direita

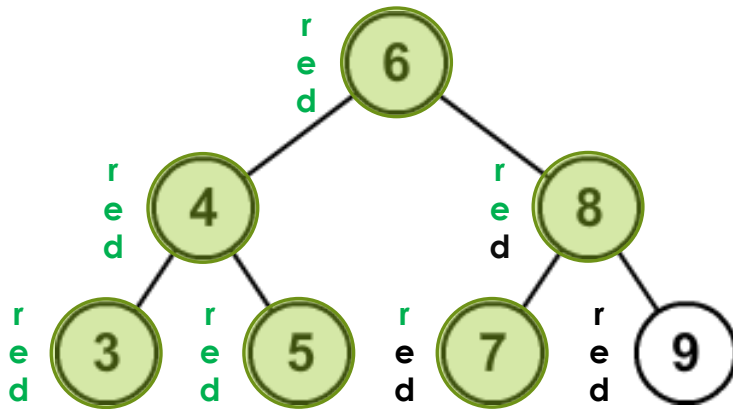


6 4 3 5 8

# Árvore Binária

## Percursos – Pré-ordem

● Raiz → Esquerda → Direita

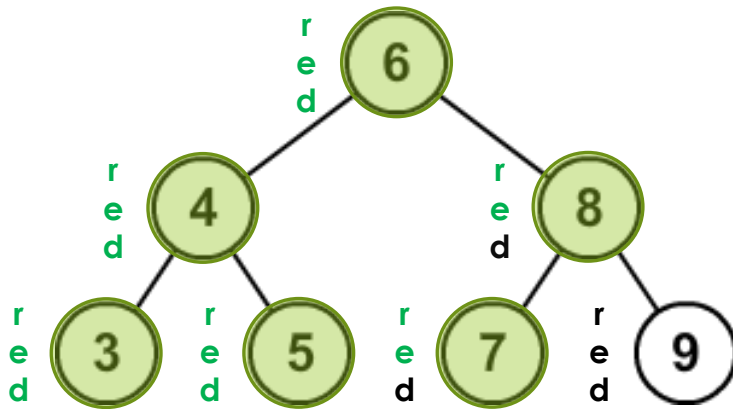


6 4 3 5 8 7

# Árvore Binária

## Percursos – Pré-ordem

● Raiz → Esquerda → Direita

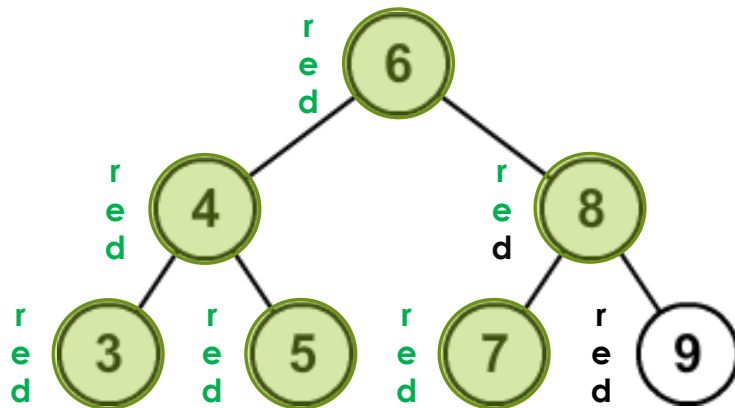


6 4 3 5 8 7

# Árvore Binária

## Percursos – Pré-ordem

● Raiz → Esquerda → Direita

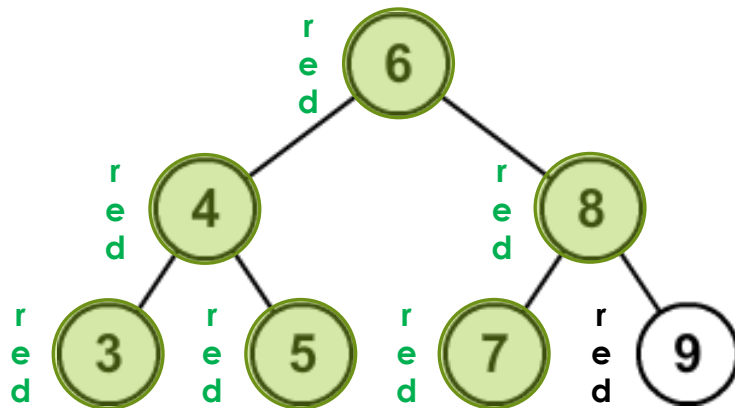


6 4 3 5 8 7

# Árvore Binária

## Percursos – Pré-ordem

● Raiz → Esquerda → Direita

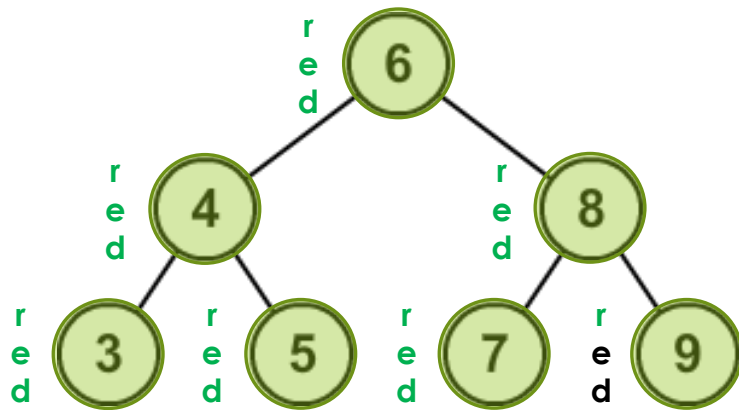


6 4 3 5 8 7

# Árvore Binária

Percursos – Pré-ordem

● Raiz → Esquerda → Direita

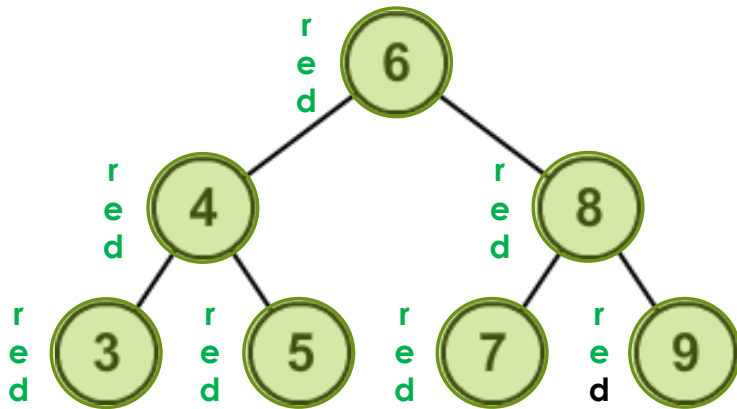


6 4 3 5 8 7 9

# Árvore Binária

Percursos – Pré-ordem

● Raiz → Esquerda → Direita

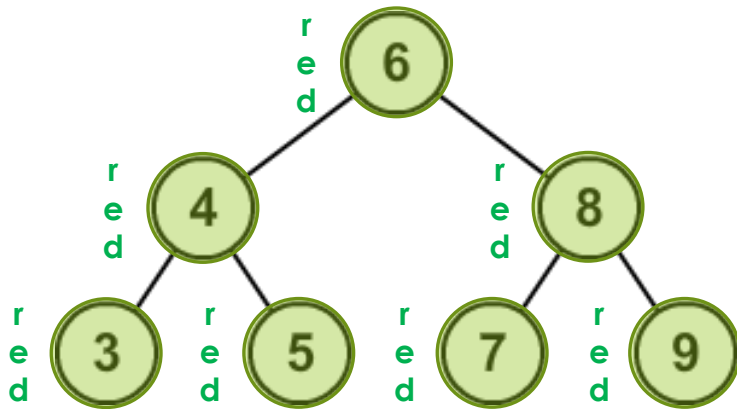


6 4 3 5 8 7 9

# Árvore Binária

Percursos – Pré-ordem

● Raiz → Esquerda → Direita



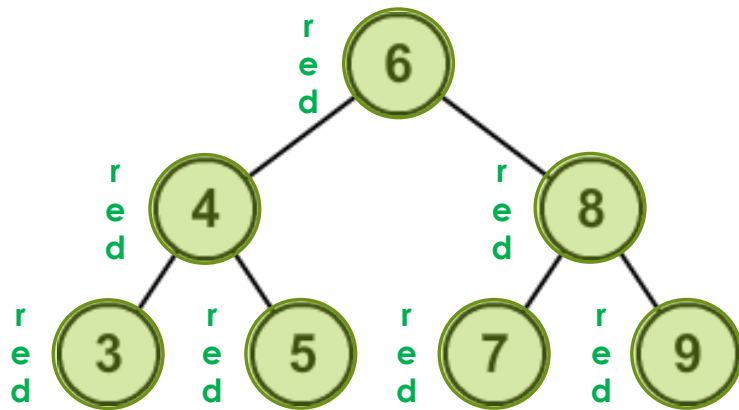
6 4 3 5 8 7 9



# Árvore Binária

## Percursos – Pré-ordem

● Raiz → Esquerda → Direita

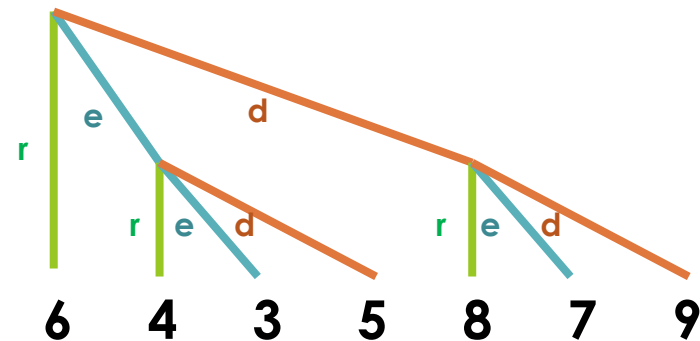
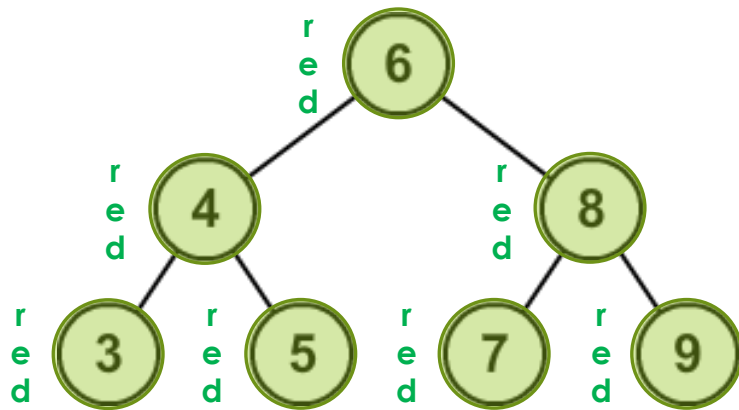


6 4 3 5 8 7 9

# Árvore Binária

Percursos – Pré-ordem

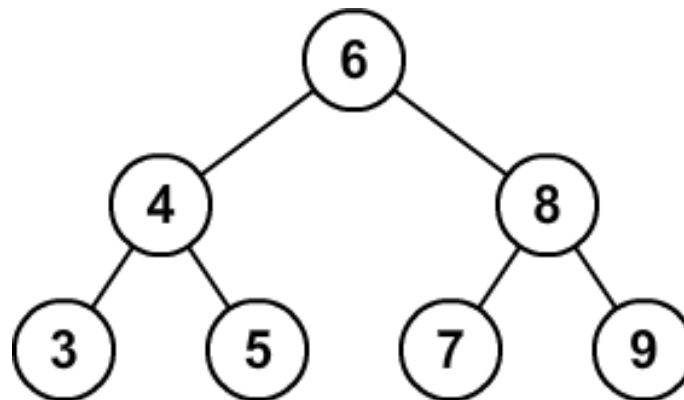
● Raiz → Esquerda → Direita



# Árvore Binária

Percursos – Em ordem (ordem simétrica)

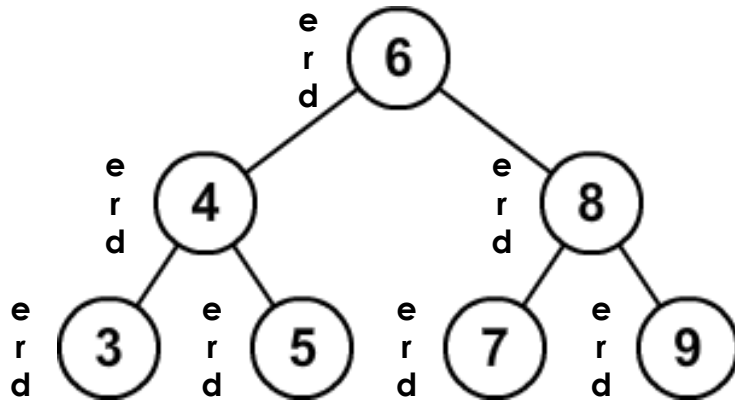
- Esquerda → Raiz → Direita



# Árvore Binária

Percursos – Em ordem (ordem simétrica)

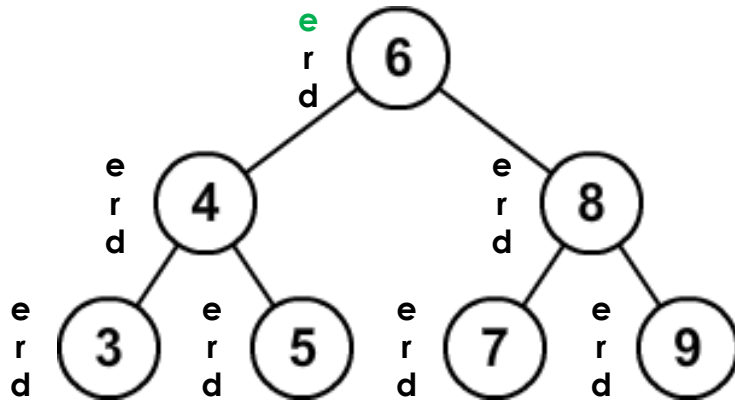
● Esquerda → Raiz → Direita



# Árvore Binária

Percursos – Em ordem (ordem simétrica)

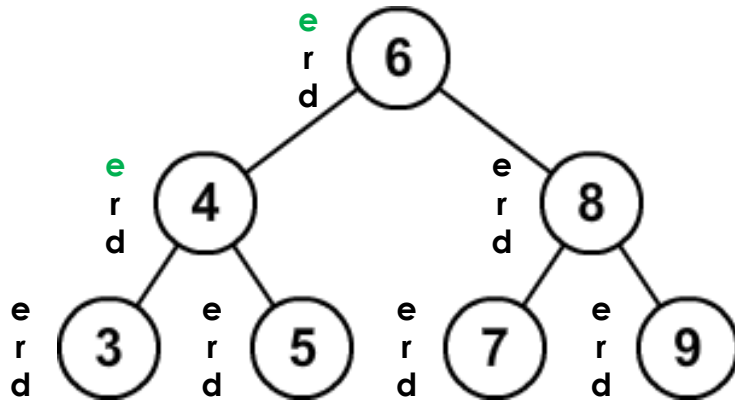
○ Esquerda → Raiz → Direita



# Árvore Binária

Percursos – Em ordem (ordem simétrica)

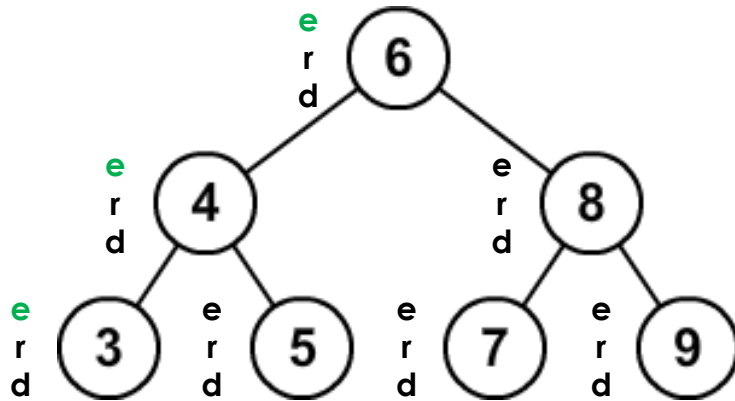
○ Esquerda → Raiz → Direita



# Árvore Binária

Percursos – Em ordem (ordem simétrica)

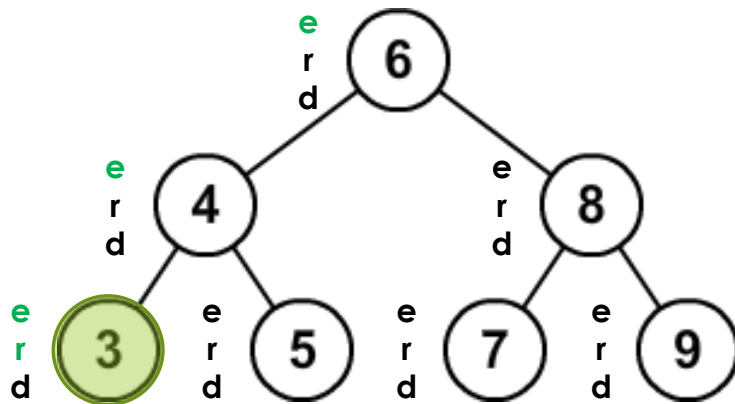
○ Esquerda → Raiz → Direita



# Árvore Binária

Percursos – Em ordem (ordem simétrica)

○ Esquerda → Raiz → Direita



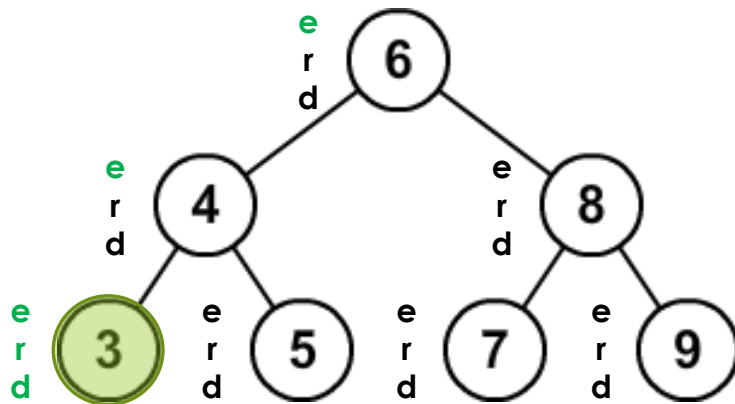
3



# Árvore Binária

Percursos – Em ordem (ordem simétrica)

○ Esquerda → Raiz → Direita

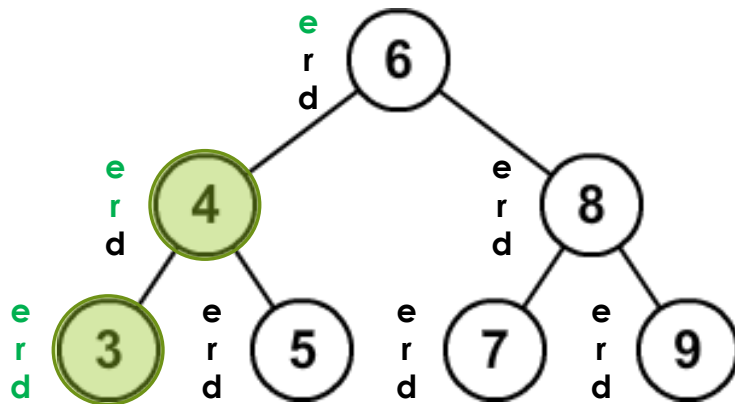


3

# Árvore Binária

Percursos – Em ordem (ordem simétrica)

○ Esquerda → Raiz → Direita

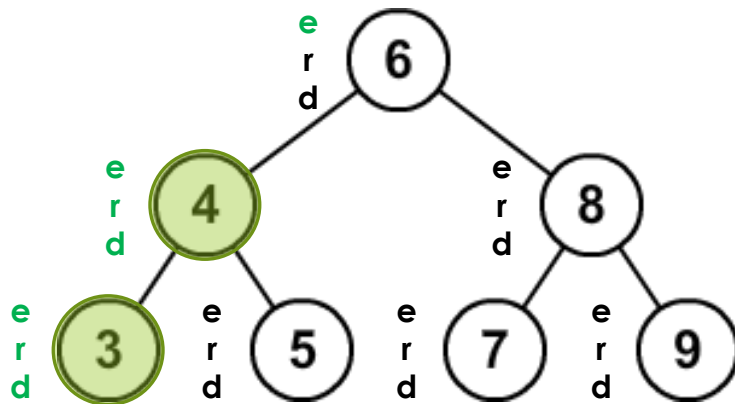


3 4

# Árvore Binária

Percursos – Em ordem (ordem simétrica)

○ Esquerda → Raiz → Direita

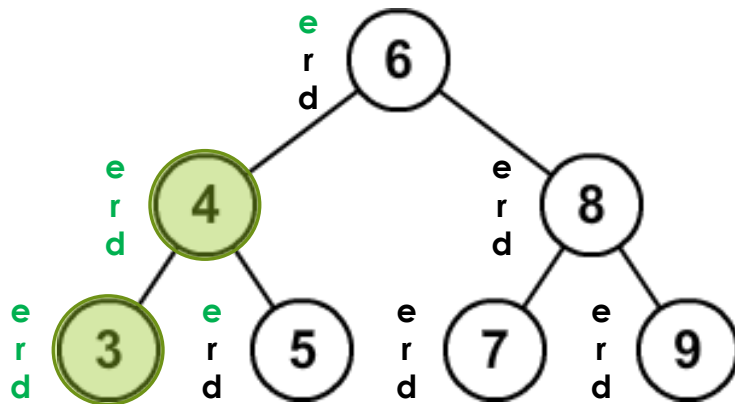


3 4

# Árvore Binária

Percursos – Em ordem (ordem simétrica)

○ Esquerda → Raiz → Direita

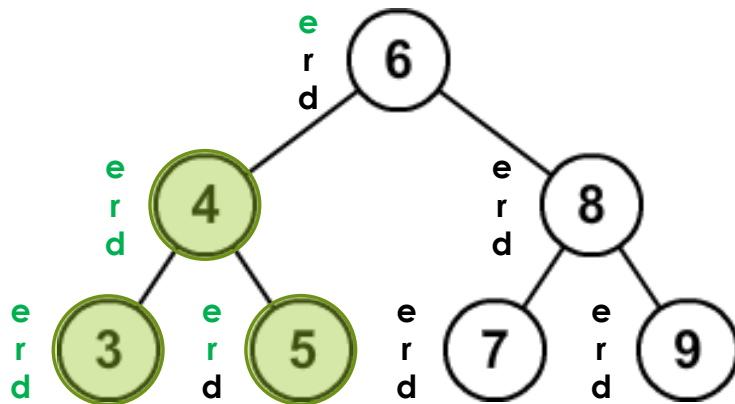


3 4

# Árvore Binária

Percursos – Em ordem (ordem simétrica)

○ Esquerda → Raiz → Direita

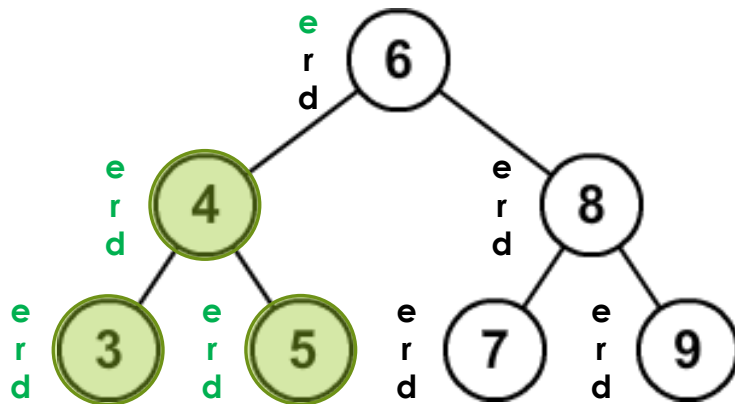


3 4 5

# Árvore Binária

Percursos – Em ordem (ordem simétrica)

- Esquerda → Raiz → Direita

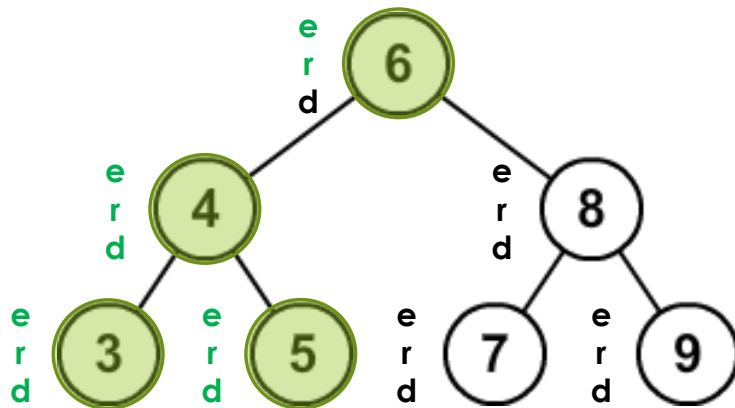


3 4 5

# Árvore Binária

Percursos – Em ordem (ordem simétrica)

- Esquerda → Raiz → Direita

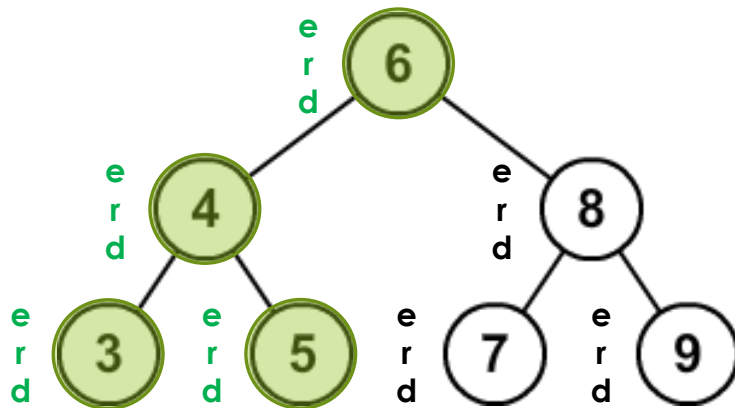


3 4 5 6

# Árvore Binária

Percursos – Em ordem (ordem simétrica)

- Esquerda → Raiz → Direita



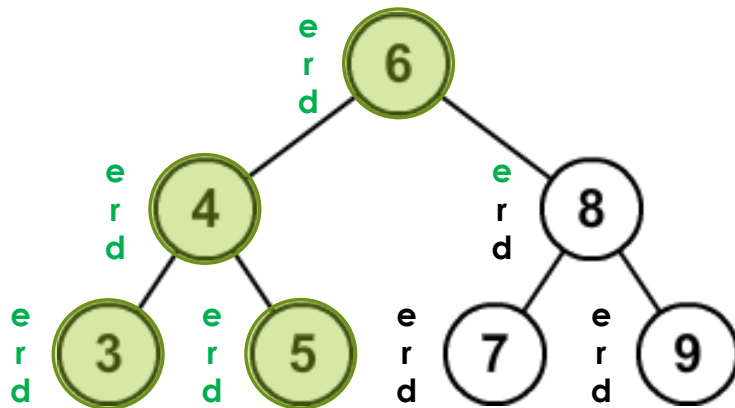
3 4 5 6



# Árvore Binária

Percursos – Em ordem (ordem simétrica)

○ Esquerda → Raiz → Direita

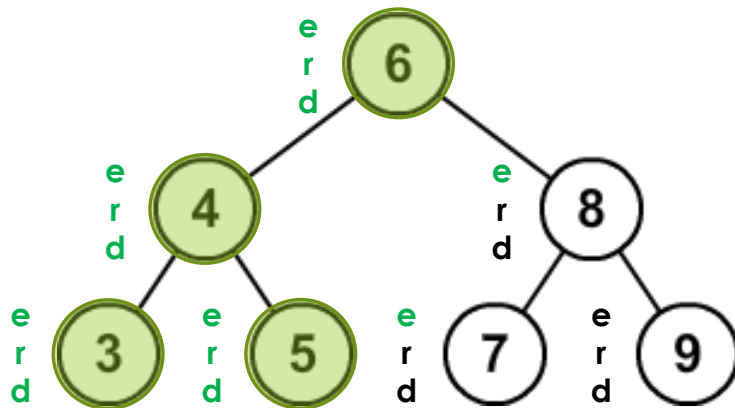


3 4 5 6

# Árvore Binária

Percursos – Em ordem (ordem simétrica)

○ Esquerda → Raiz → Direita

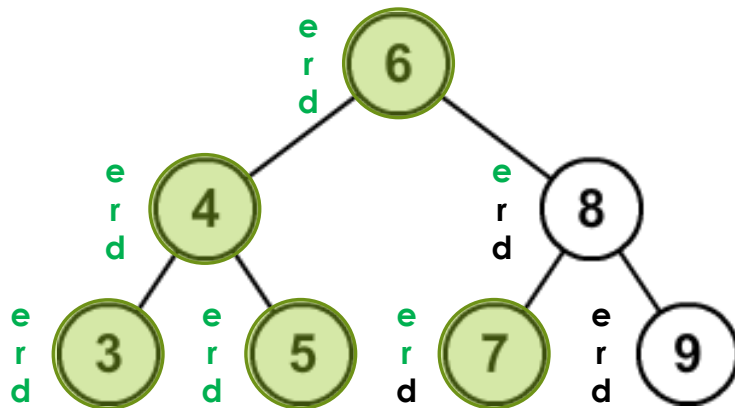


3 4 5 6

# Árvore Binária

Percursos – Em ordem (ordem simétrica)

- Esquerda → Raiz → Direita

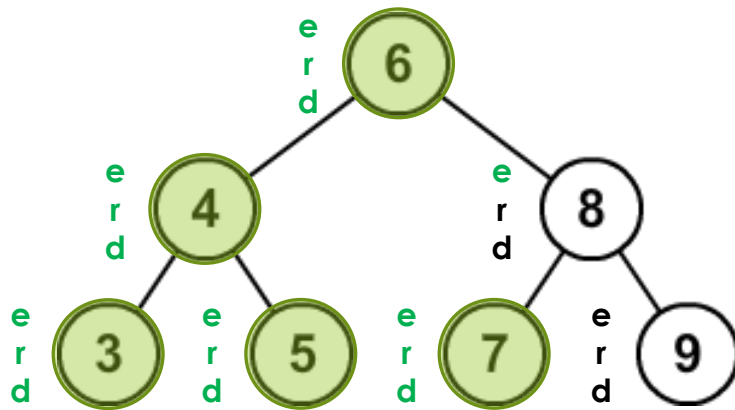


3 4 5 6 7

# Árvore Binária

Percursos – Em ordem (ordem simétrica)

- Esquerda → Raiz → Direita

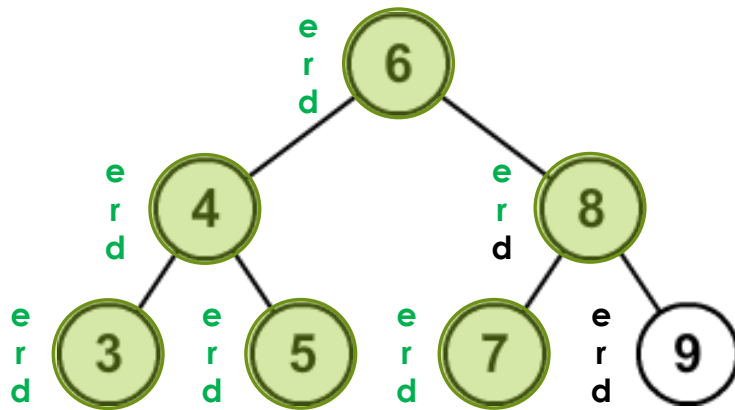


3 4 5 6 7

# Árvore Binária

Percursos – Em ordem (ordem simétrica)

- Esquerda → Raiz → Direita

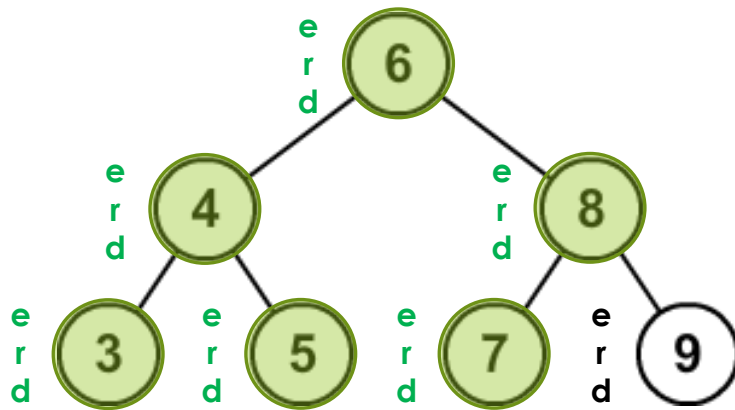


3 4 5 6 7 8

# Árvore Binária

Percursos – Em ordem (ordem simétrica)

- Esquerda → Raiz → Direita

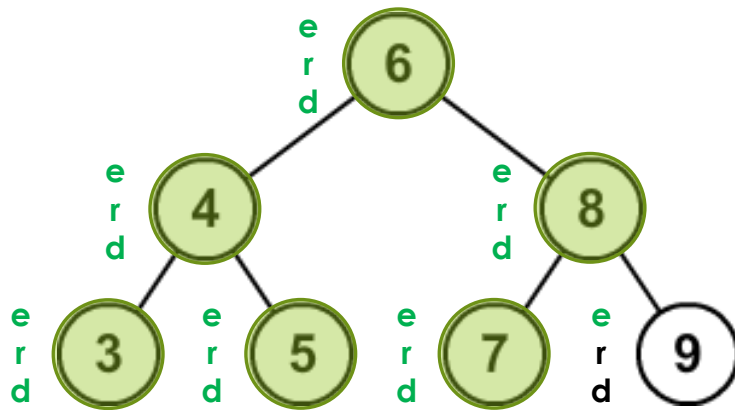


3 4 5 6 7 8

# Árvore Binária

Percursos – Em ordem (ordem simétrica)

- Esquerda → Raiz → Direita

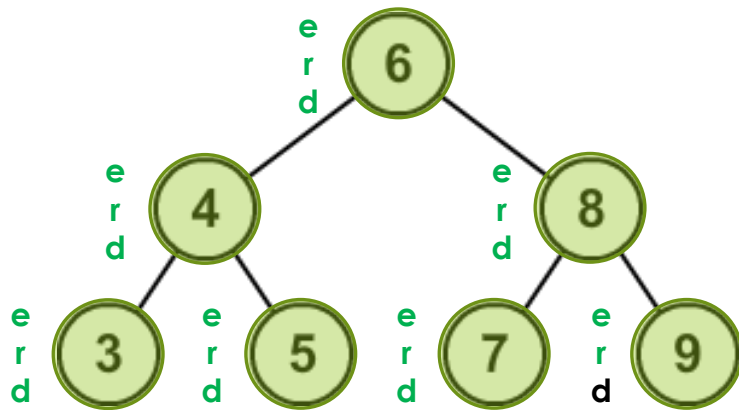


3 4 5 6 7 8

# Árvore Binária

Percursos – Em ordem (ordem simétrica)

- Esquerda → Raiz → Direita



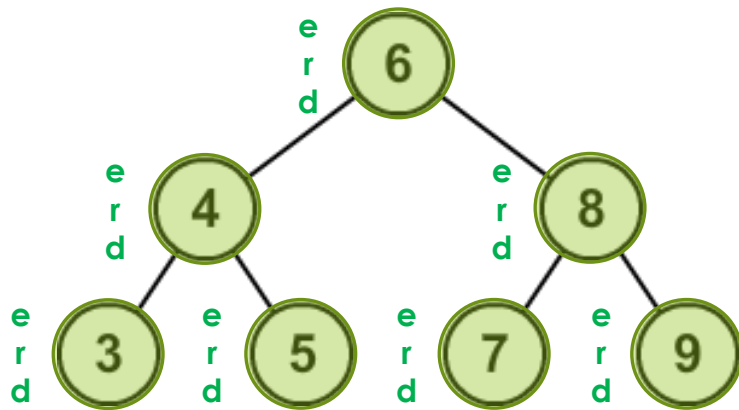
3 4 5 6 7 8 9



# Árvore Binária

Percursos – Em ordem (ordem simétrica)

○ Esquerda → Raiz → Direita

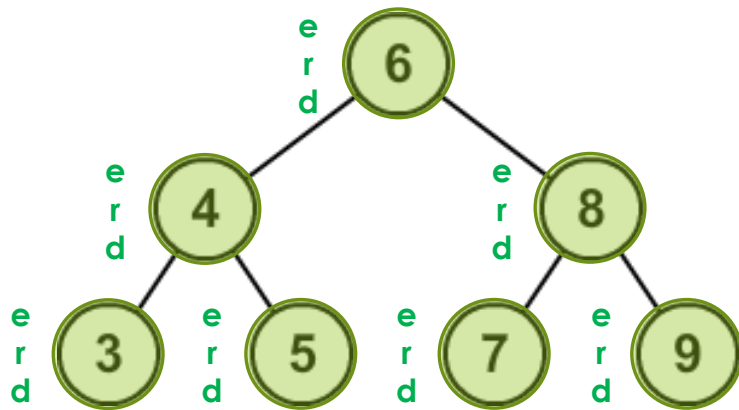


3 4 5 6 7 8 9

# Árvore Binária

Percursos – Em ordem (ordem simétrica)

- Esquerda → Raiz → Direita

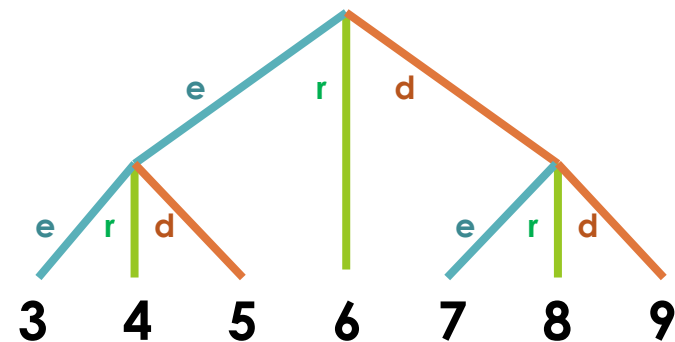
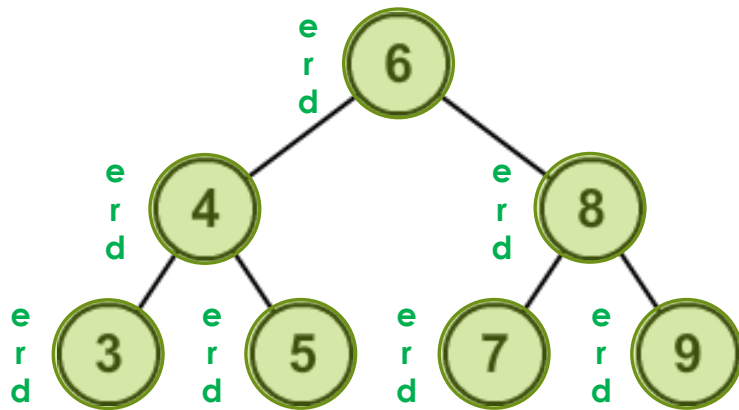


3 4 5 6 7 8 9

# Árvore Binária

Percursos – Em ordem (ordem simétrica)

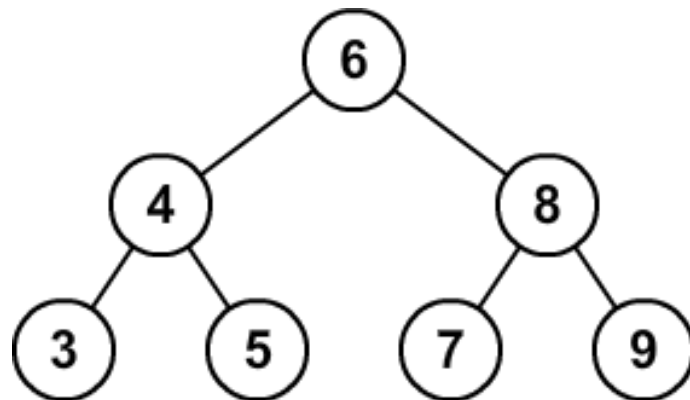
- Esquerda → Raiz → Direita



# Árvore Binária

Percursos – Pós-ordem (ordem final)

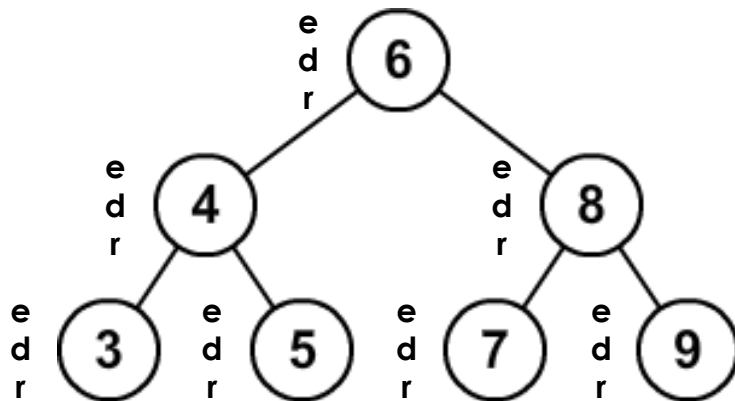
● Esquerda → Direita → Raiz



# Árvore Binária

Percursos – Pós-ordem (ordem final)

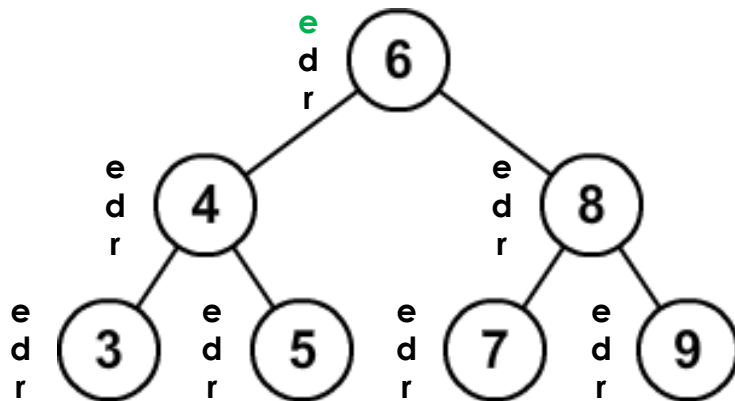
● Esquerda → Direita → Raiz



# Árvore Binária

Percursos – Pós-ordem (ordem final)

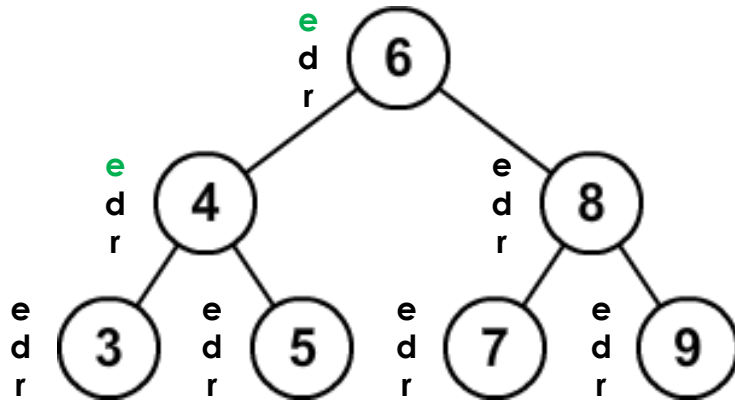
● Esquerda → Direita → Raiz



# Árvore Binária

Percursos – Pós-ordem (ordem final)

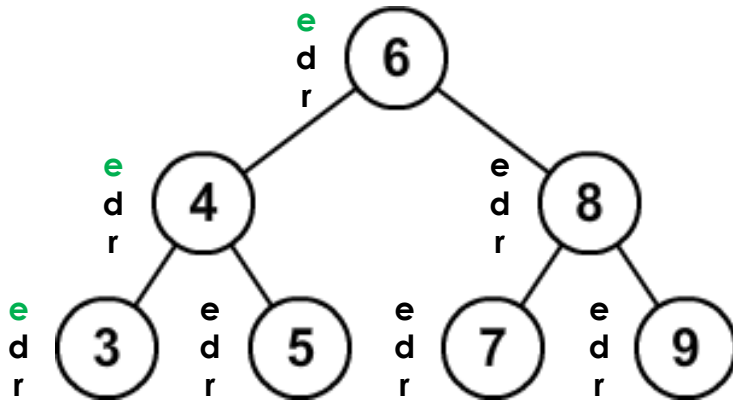
● Esquerda → Direita → Raiz



# Árvore Binária

Percursos – Pós-ordem (ordem final)

● Esquerda → Direita → Raiz

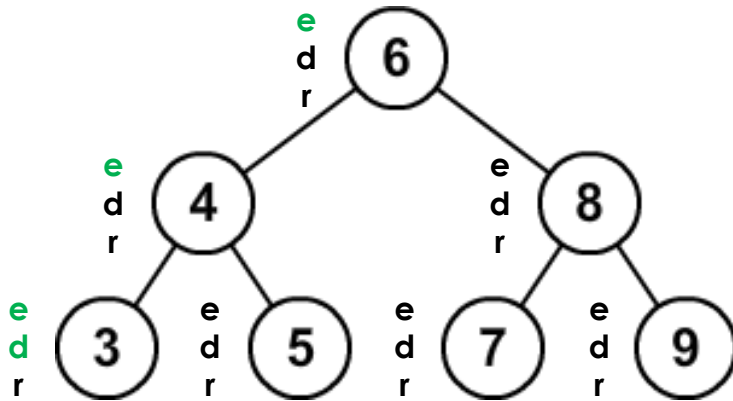




# Árvore Binária

Percursos – Pós-ordem (ordem final)

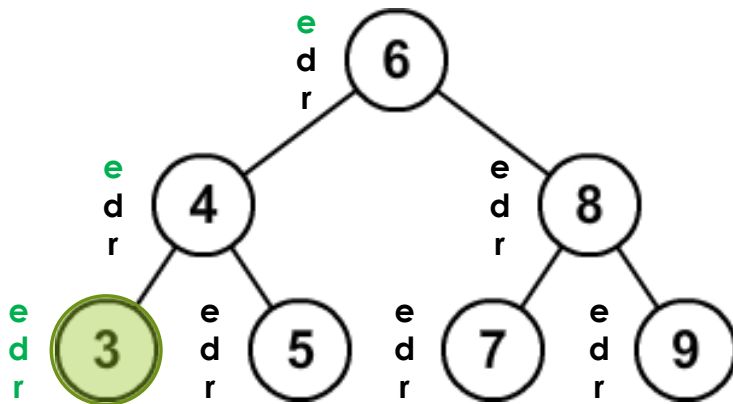
● Esquerda → Direita → Raiz



# Árvore Binária

Percursos – Pós-ordem (ordem final)

○ Esquerda → Direita → Raiz

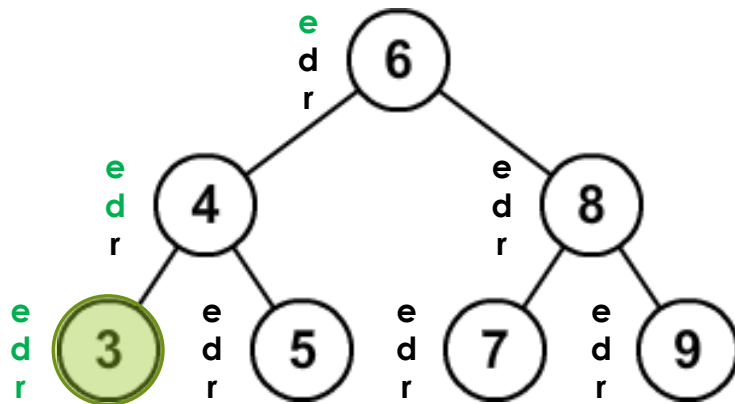


3

# Árvore Binária

Percursos – Pós-ordem (ordem final)

○ Esquerda → Direita → Raiz

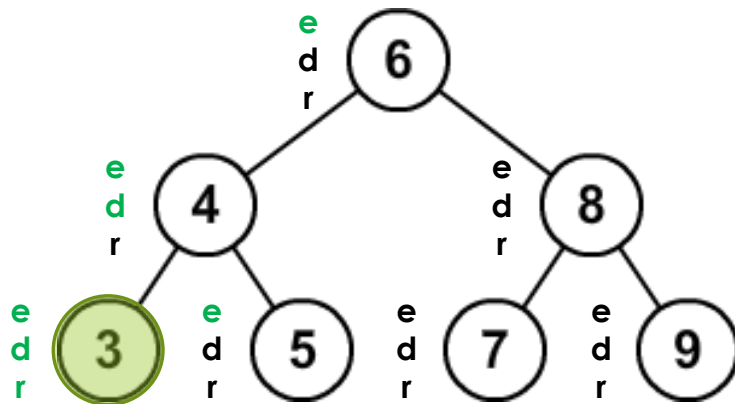


3

# Árvore Binária

Percursos – Pós-ordem (ordem final)

○ Esquerda → Direita → Raiz

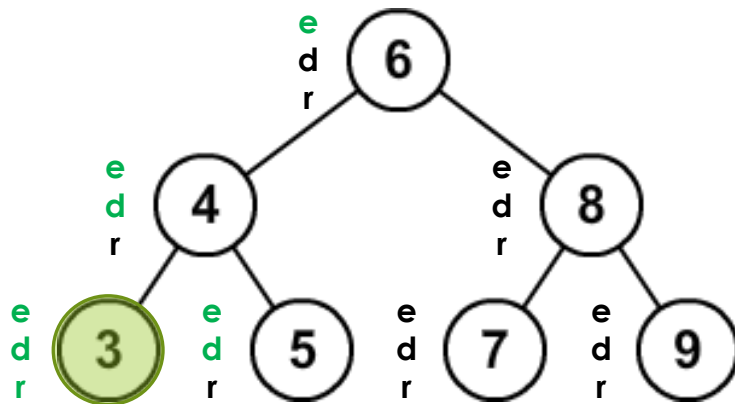


3

# Árvore Binária

Percursos – Pós-ordem (ordem final)

○ Esquerda → Direita → Raiz

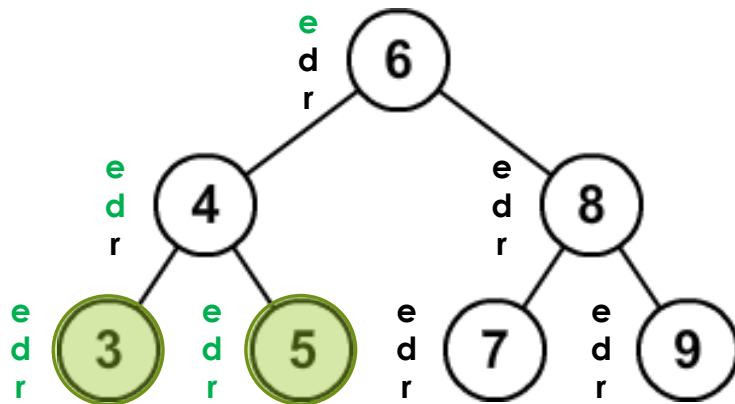


3

# Árvore Binária

Percursos – Pós-ordem (ordem final)

● Esquerda → Direita → Raiz

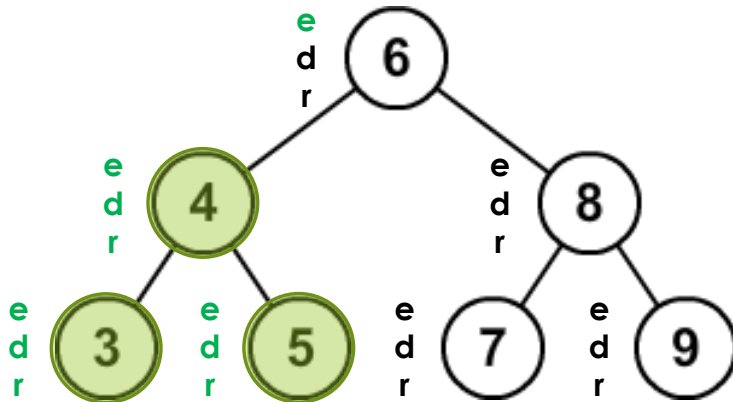


3 5

# Árvore Binária

Percursos – Pós-ordem (ordem final)

● Esquerda → Direita → Raiz

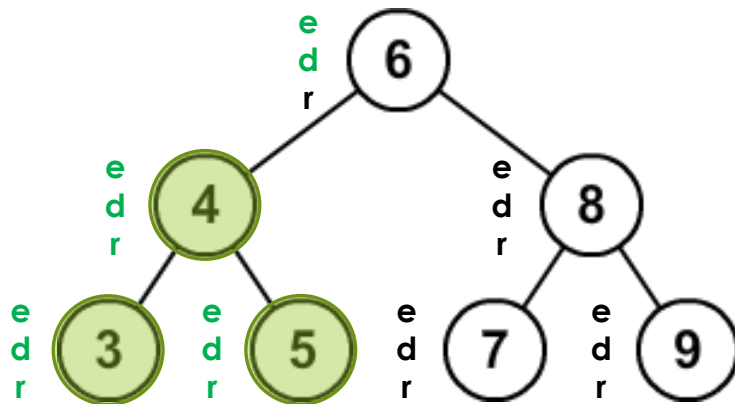


**3 5 4**

# Árvore Binária

Percursos – Pós-ordem (ordem final)

○ Esquerda → Direita → Raiz



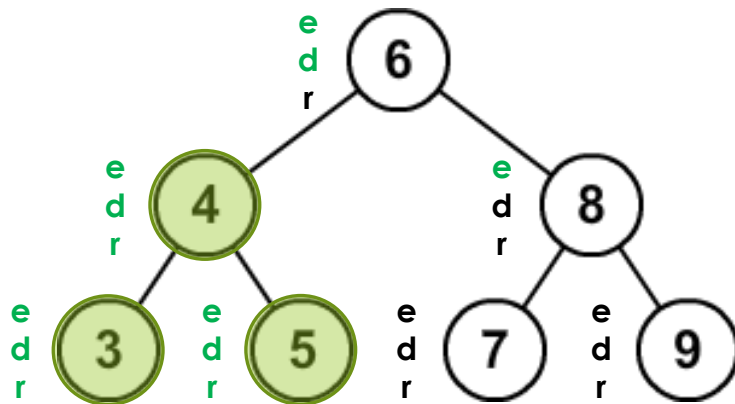
**3 5 4**



# Árvore Binária

Percursos – Pós-ordem (ordem final)

● Esquerda → Direita → Raiz

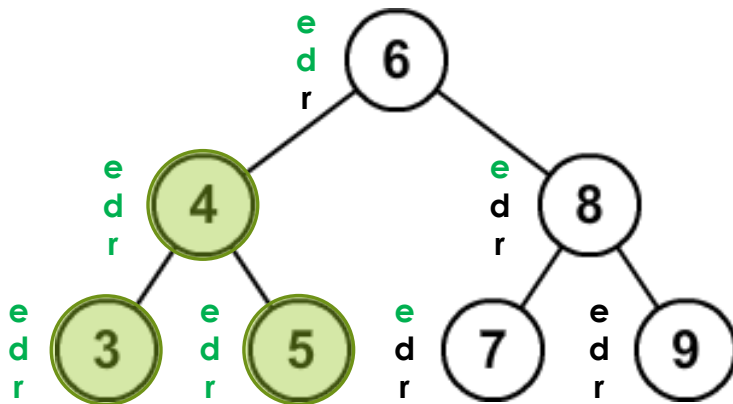


**3 5 4**

# Árvore Binária

Percursos – Pós-ordem (ordem final)

○ Esquerda → Direita → Raiz

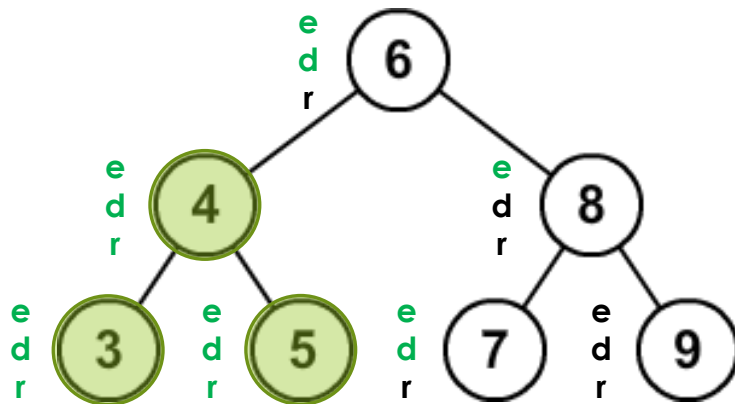


**3 5 4**

# Árvore Binária

Percursos – Pós-ordem (ordem final)

● Esquerda → Direita → Raiz

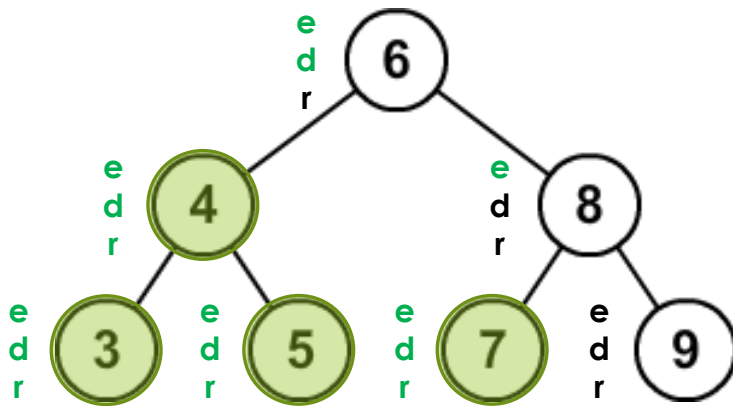


**3 5 4**

# Árvore Binária

Percursos – Pós-ordem (ordem final)

● Esquerda → Direita → Raiz

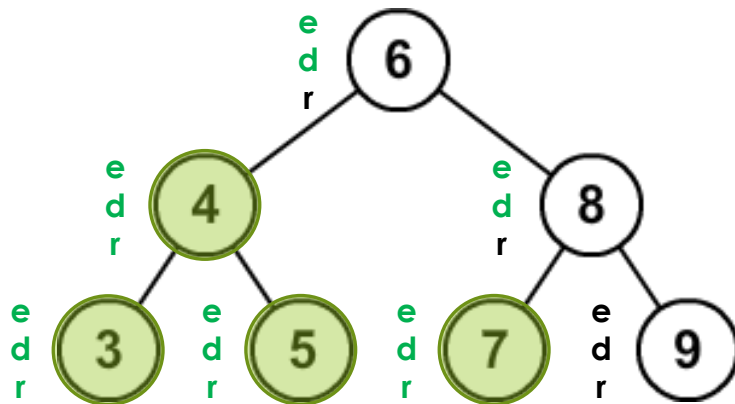


3 5 4 7

# Árvore Binária

Percursos – Pós-ordem (ordem final)

● Esquerda → Direita → Raiz

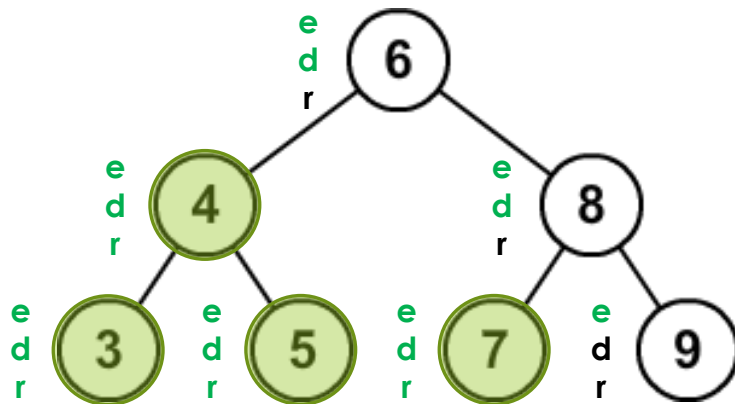


3 5 4 7

# Árvore Binária

Percursos – Pós-ordem (ordem final)

● Esquerda → Direita → Raiz

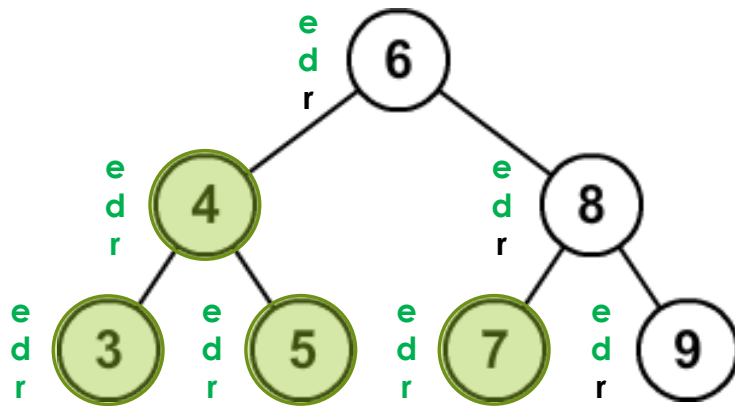


3 5 4 7

# Árvore Binária

Percursos – Pós-ordem (ordem final)

● Esquerda → Direita → Raiz

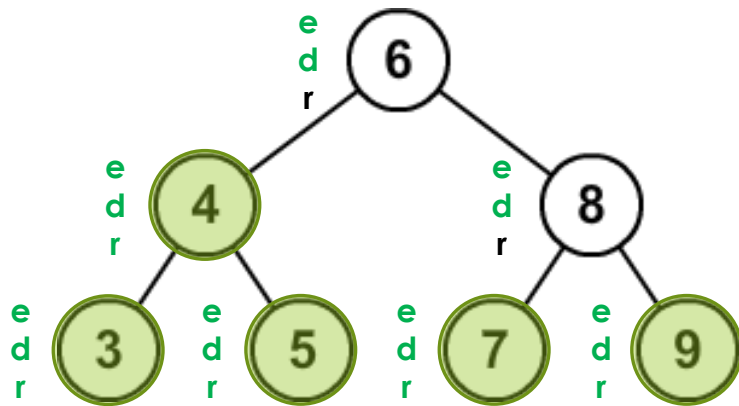


3 5 4 7

# Árvore Binária

Percursos – Pós-ordem (ordem final)

● Esquerda → Direita → Raiz



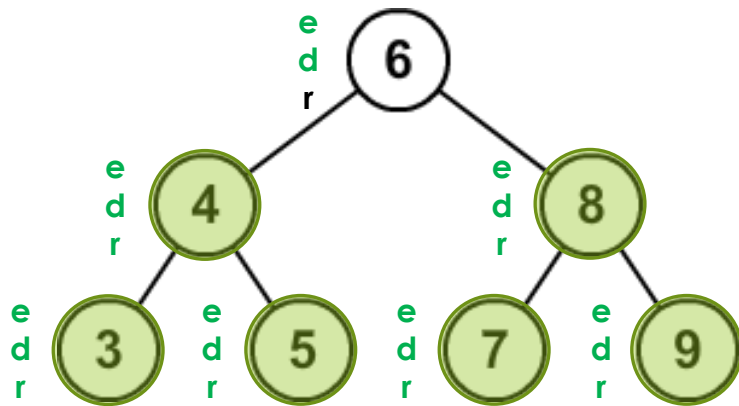
3 5 4 7 9



# Árvore Binária

Percursos – Pós-ordem (ordem final)

● Esquerda → Direita → Raiz

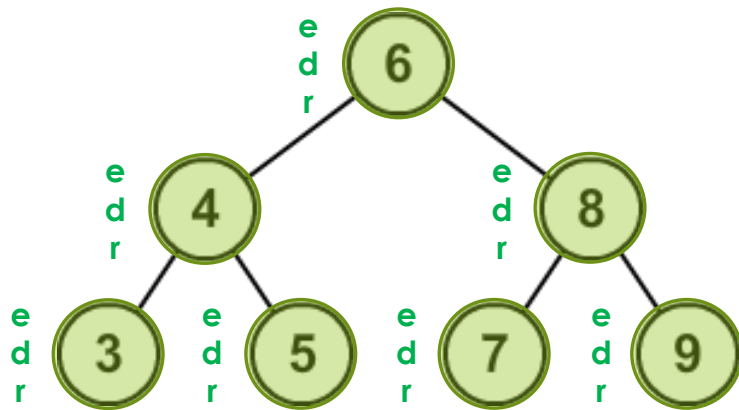


3 5 4 7 9 8

# Árvore Binária

Percursos – Pós-ordem (ordem final)

● Esquerda → Direita → Raiz

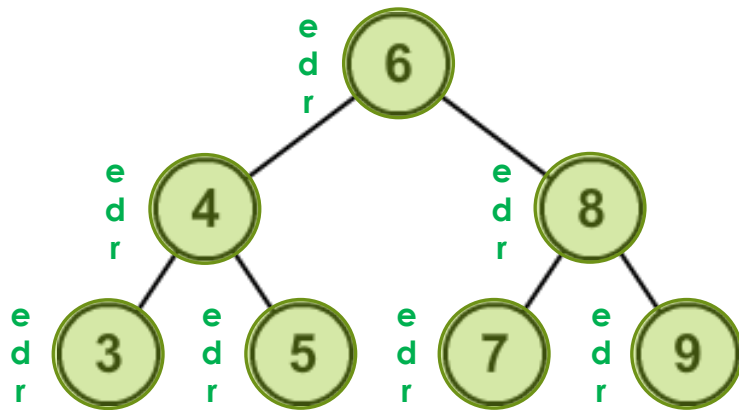


3 5 4 7 9 8 6

# Árvore Binária

Percursos – Pós-ordem (ordem final)

● Esquerda → Direita → Raiz

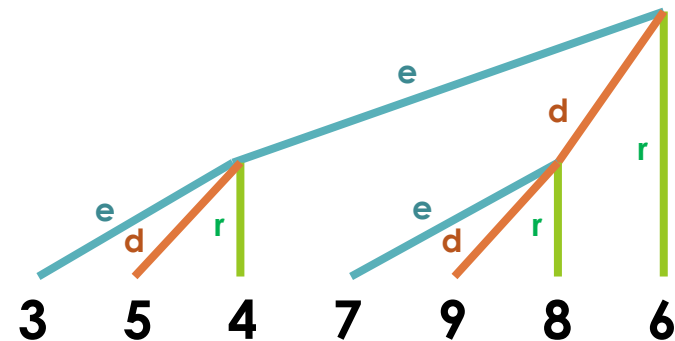
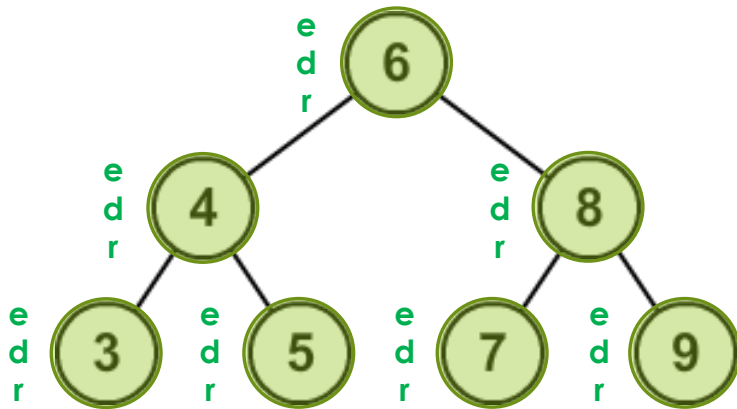


3 5 4 7 9 8 6

# Árvore Binária

Percursos – Pós-ordem (ordem final)

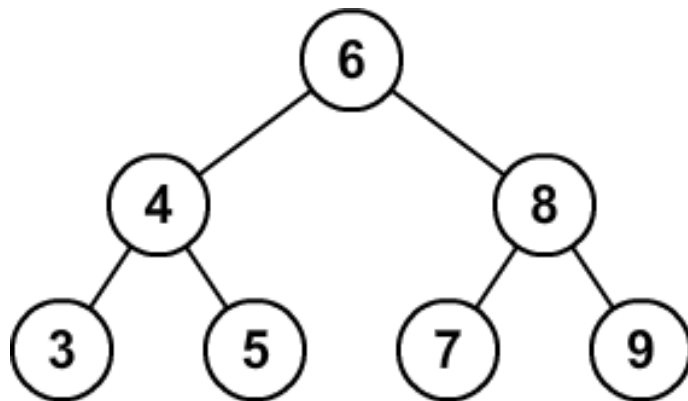
○ Esquerda → Direita → Raiz



# Árvore Binária

Percursos – Pré-ordem inverso

● Raiz → Direita → Esquerda

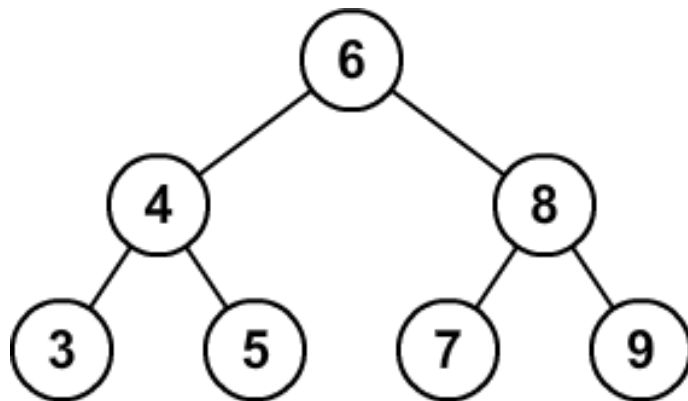


**6 8 9 7 4 5 3**

# Árvore Binária

Percursos – Em ordem (ordem simétrica) inverso

● Direita → Raiz → Esquerda

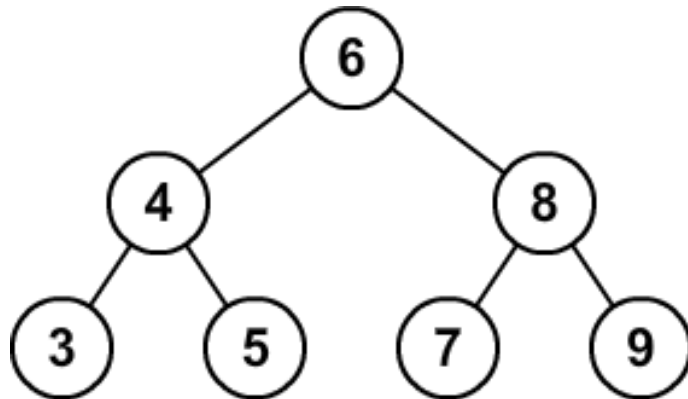


**9 8 7 6 5 4 3**

# Árvore Binária

Percursos – Pós-ordem (ordem final) inverso

● Direita → Esquerda → Raiz



9 7 8 5 3 4 6

# Bibliografia

- ASCENCIO, A. F. G.; ARAÚJO, G. S. **Estruturas de Dados**. São Paulo: Pearson, 2011. 432 p.
- TENENBAUM, A. M.; LANGSAM, Y.; AUGENSTEIN, M. J. **Estrutura de dados usando C**. São Paulo: Pearson Makron Books, 1995. 884 p.
- VILLAS, M. V.; FERREIRA, A. G. M.; LEROY, P. G.; MIRANDA, C.; BOCKMAN, C. L. **Estruturas de Dados: Conceitos e Implementação**. Rio de Janeiro: Editora Campus, 1993. 298 p.