

# Grafos

*Ancestrais*

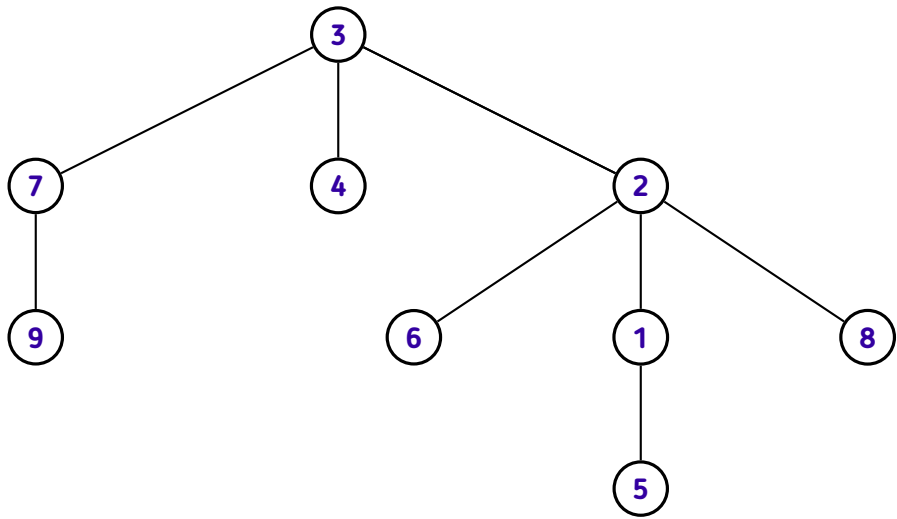
**Prof. Edson Alves**

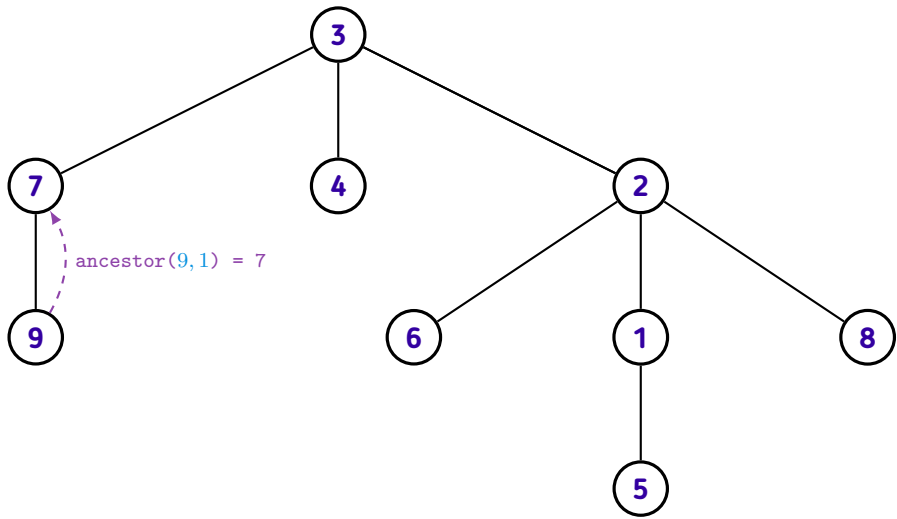
***Faculdade UnB Gama***

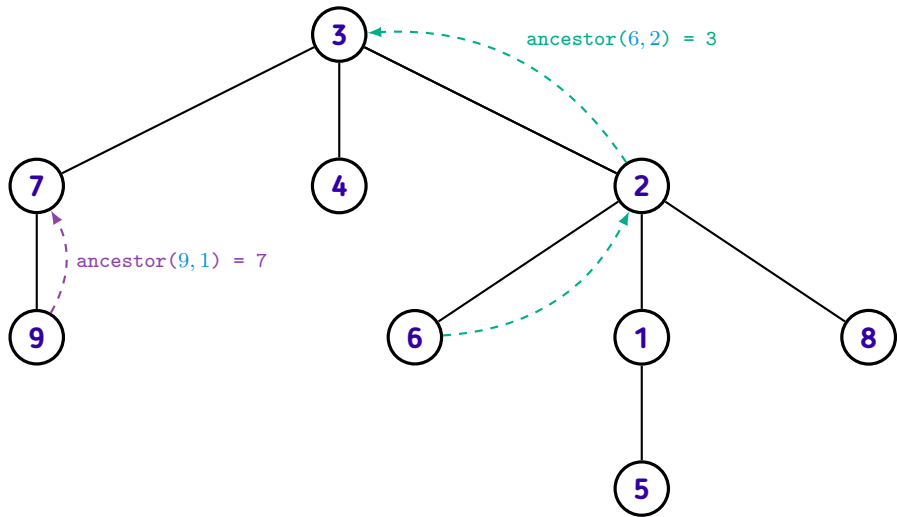
## $k$ -ésimo ancestral

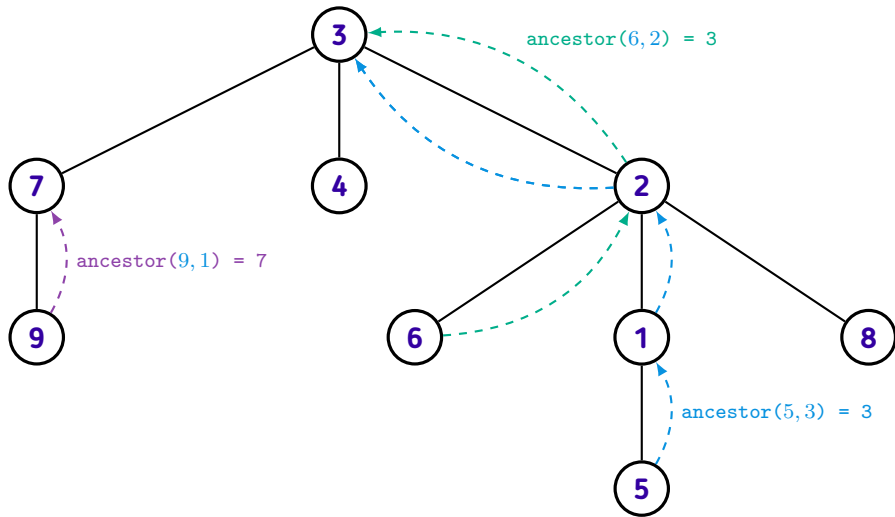
Seja  $T$  uma árvore enraizada,  $u$  um vértice de  $T$  e  $k$  um inteiro positivo.

O  $k$ -ésimo ancestral de  $u$  é o nó  $v$  que encerra o caminho que parte de  $u$  e segue  $k$  níveis, em direção à raiz. **Notação:**  $v = \text{ancestor}(u, k)$ .









**Identificação do  $k$ -ésimo ancestral em  $O(\log N)$**

## Identificação do $k$ -ésimo ancestral em $O(\log N)$

- ★ É possível identificar o  $k$ -ésimo ancestral em  $O(\log N)$ , onde  $N$  é o número de vértices da árvore, por meio de um algoritmo de programação dinâmica



## Identificação do $k$ -ésimo ancestral em $O(\log N)$

- ★ É possível identificar o  $k$ -ésimo ancestral em  $O(\log N)$ , onde  $N$  é o número de vértices da árvore, por meio de um algoritmo de programação dinâmica
- ★ Seja  $p(u)$  o ancestral direto de  $u$  na árvore enraizada

## Identificação do $k$ -ésimo ancestral em $O(\log N)$

★ É possível identificar o  $k$ -ésimo ancestral em  $O(\log N)$ , onde  $N$  é o número de vértices da árvore, por meio de um algoritmo de programação dinâmica

★ Seja  $p(u)$  o ancestral direto de  $u$  na árvore enraizada

★ O caso base ocorre com  $k = 2^0 = 1$ :  $\text{ancestor}(u, 1) = p(u)$

## Identificação do $k$ -ésimo ancestral em $O(\log N)$

★ É possível identificar o  $k$ -ésimo ancestral em  $O(\log N)$ , onde  $N$  é o número de vértices da árvore, por meio de um algoritmo de programação dinâmica

★ Seja  $p(u)$  o ancestral direto de  $u$  na árvore enraizada

★ O caso base ocorre com  $k = 2^0 = 1$ :  $\text{ancestor}(u, 1) = p(u)$

★ A transição é dada por:

$$\text{ancestor}(u, 2^k) = \text{ancestor}(\text{ancestor}(u, 2^{k-1}), 2^{k-1})$$

**Identificação do  $k$ -ésimo ancestral em  $O(\log N)$**

## Identificação do $k$ -ésimo ancestral em $O(\log N)$

★ Seja  $k$  um inteiro positivo

## Identificação do $k$ -ésimo ancestral em $O(\log N)$

- ★ Seja  $k$  um inteiro positivo
- ★ É possível escrever  $k$  como a soma de potências distintas de 2:

$$k = 2^\alpha + 2^\beta + \dots + 2^\omega$$

## Identificação do $k$ -ésimo ancestral em $O(\log N)$

- ★ Seja  $k$  um inteiro positivo
- ★ É possível escrever  $k$  como a soma de potências distintas de 2:

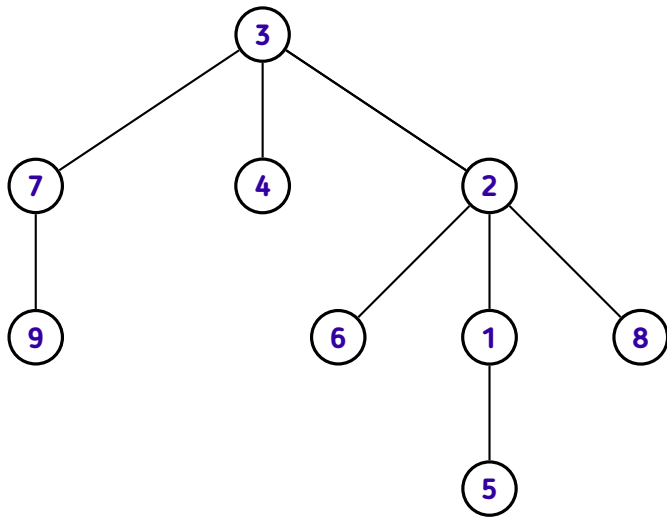
$$k = 2^\alpha + 2^\beta + \dots + 2^\omega$$

- ★ Deste modo,

$$\text{ancestor}(u, k) = \text{ancestor}(\text{ancestor}(\text{ancestor}(\text{ancestor}(u, 2^\alpha), 2^\beta), \dots), 2^\omega)$$







	$2^0$	$2^1$	$2^2$
1			
2			
3			
4			
5			
6			
7			
8			
9			
$\text{succ}(u, 2^k)$			

## Referências

1. HALIM, Felix; HALIM, Steve. *Competitive Programming 3*, 2010.
2. LAAKSONEN, Antti. *Competitive Programmer's Handbook*, 2018.