

# **Algoritmos e Estruturas de Dados II**

Busca em profundidade em grafos e  
ordenação topológica

**Prof. Flávio José Mendes Coelho**  
[fcoelho@uea.edu.br](mailto:fcoelho@uea.edu.br)

# **Plano de aula**

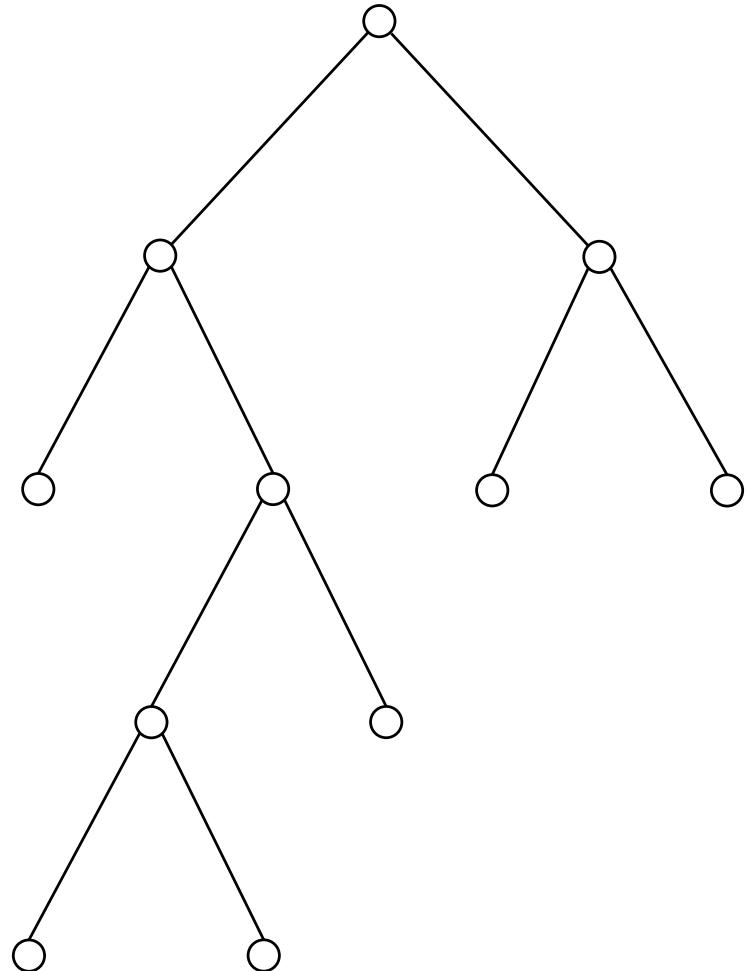
1. Busca em profundidade em grafos.
2. Ordenação topológica.
3. Exercícios.

# **Busca em profundidade**

*(depth-first search - DFS)*

# Busca em profundidade

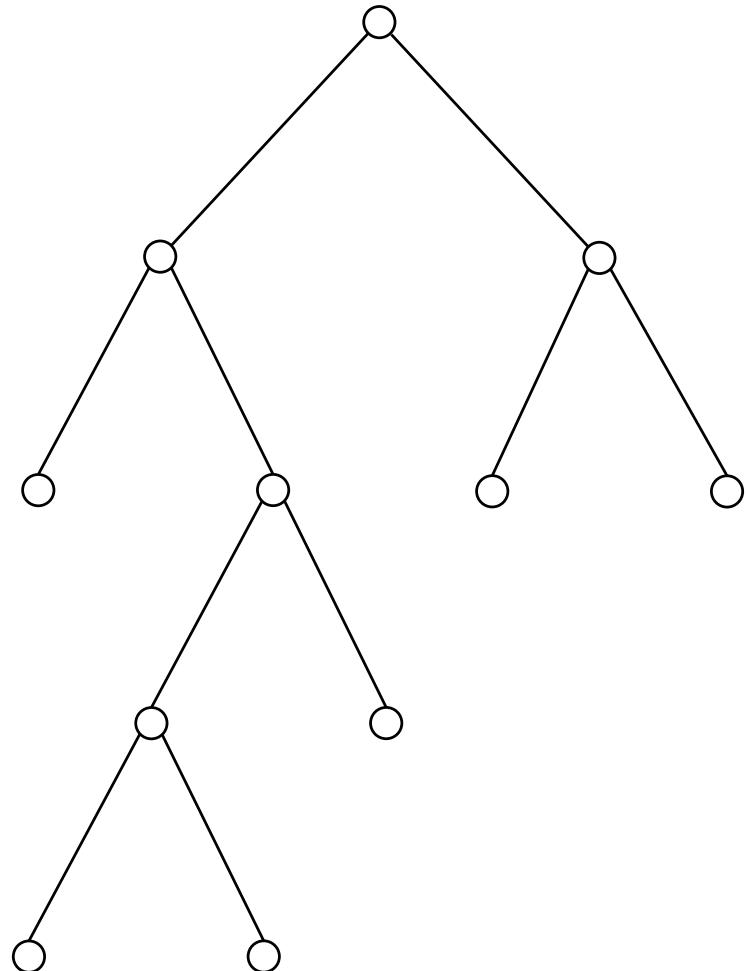
Ideia em árvores



# Busca em profundidade

## Ideia em árvores

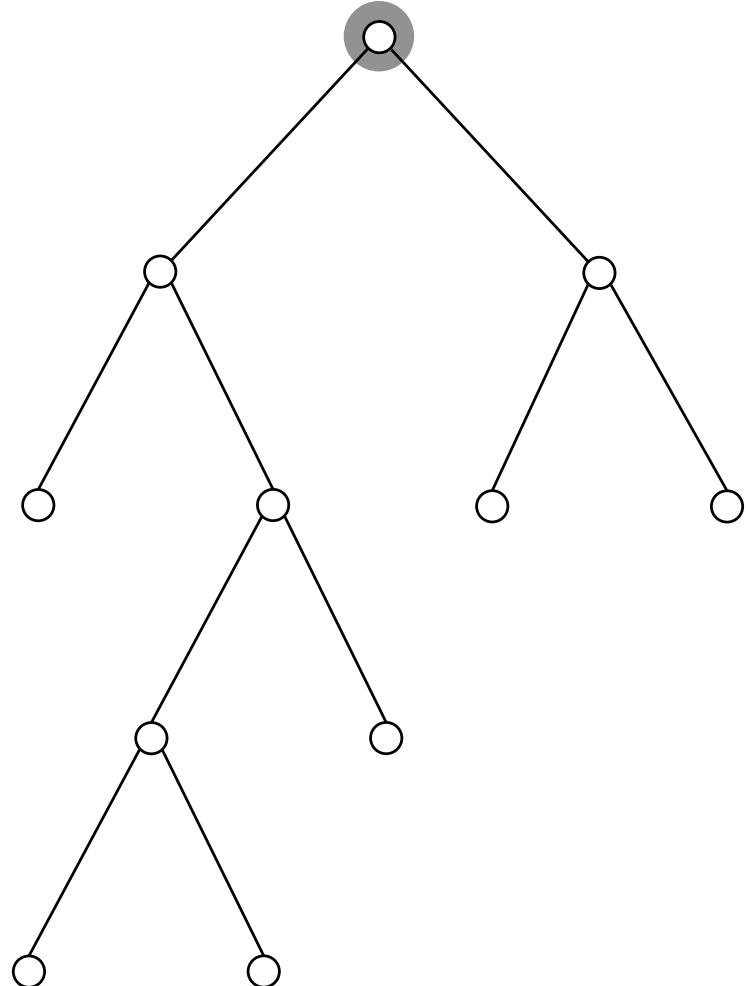
1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo w**.



# Busca em profundidade

## Ideia em árvores

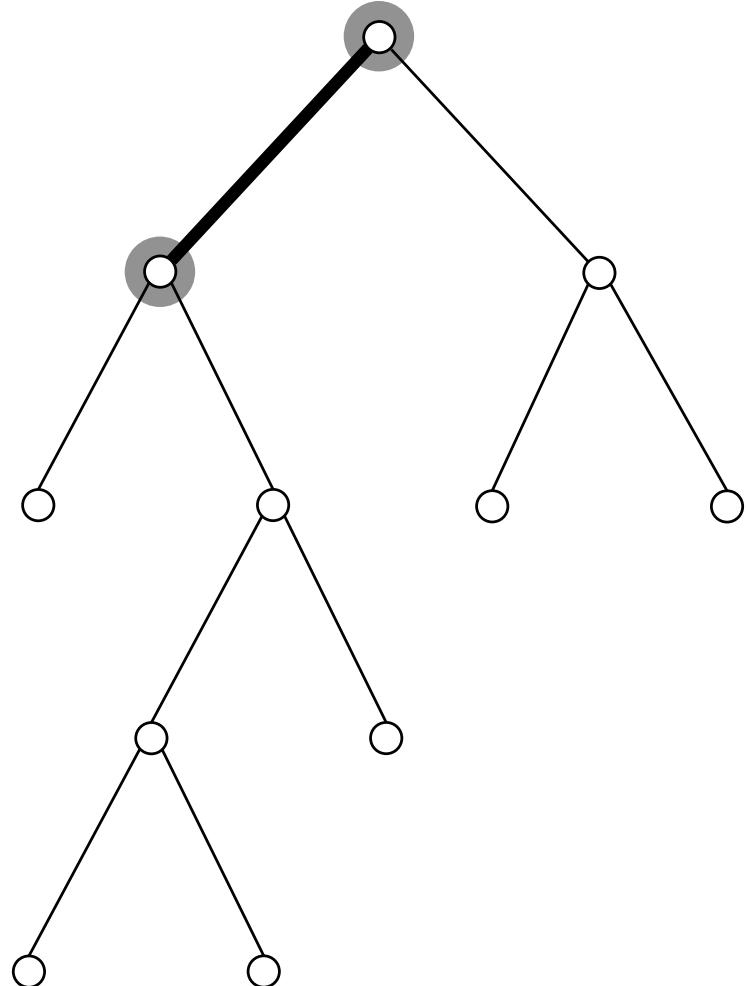
1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo w**.



# Busca em profundidade

## Ideia em árvores

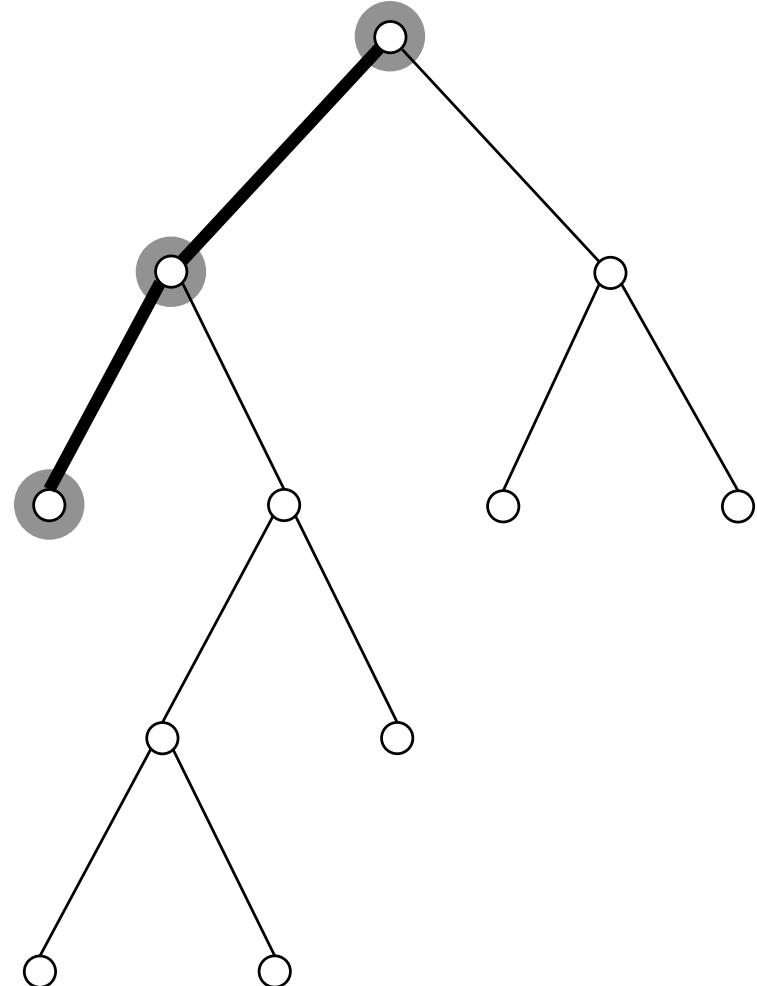
1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo w**.



# Busca em profundidade

## Ideia em árvores

1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo w**.

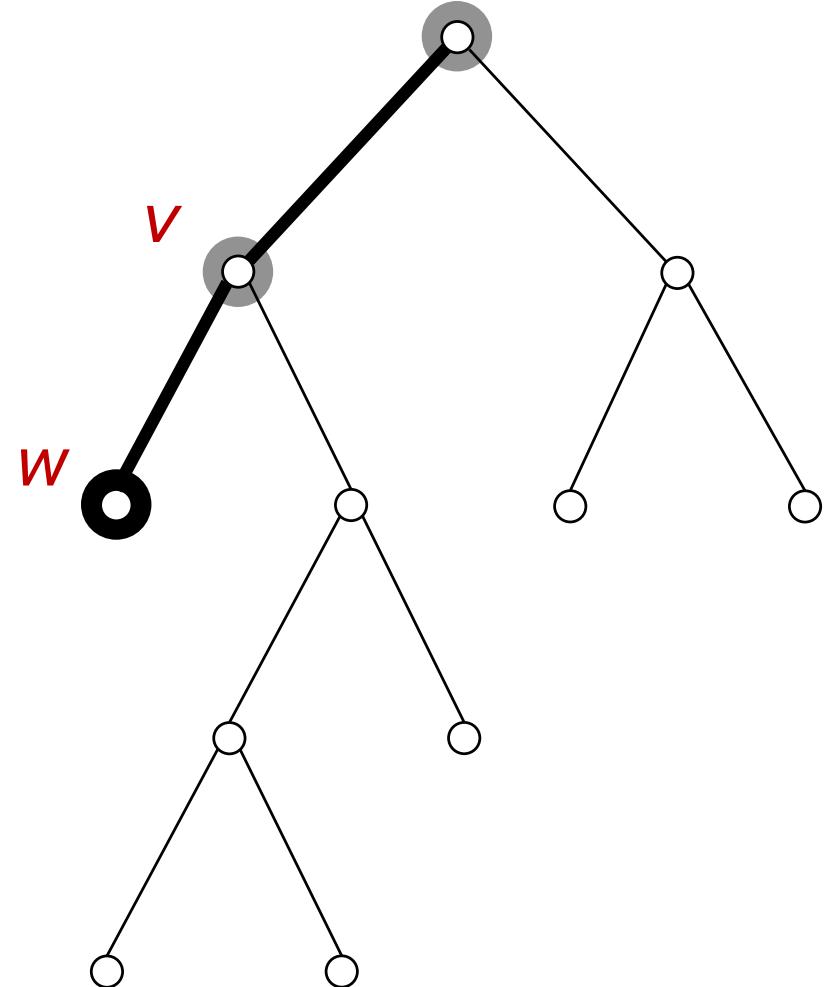


# Busca em profundidade

## Ideia em árvores

1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo**  $w$ .

2. Quando alcança, volta para o vértice anteriormente visitado  $v$  e segue **outro caminho** por outra sub-árvore de  $v$ , da mesma maneira.

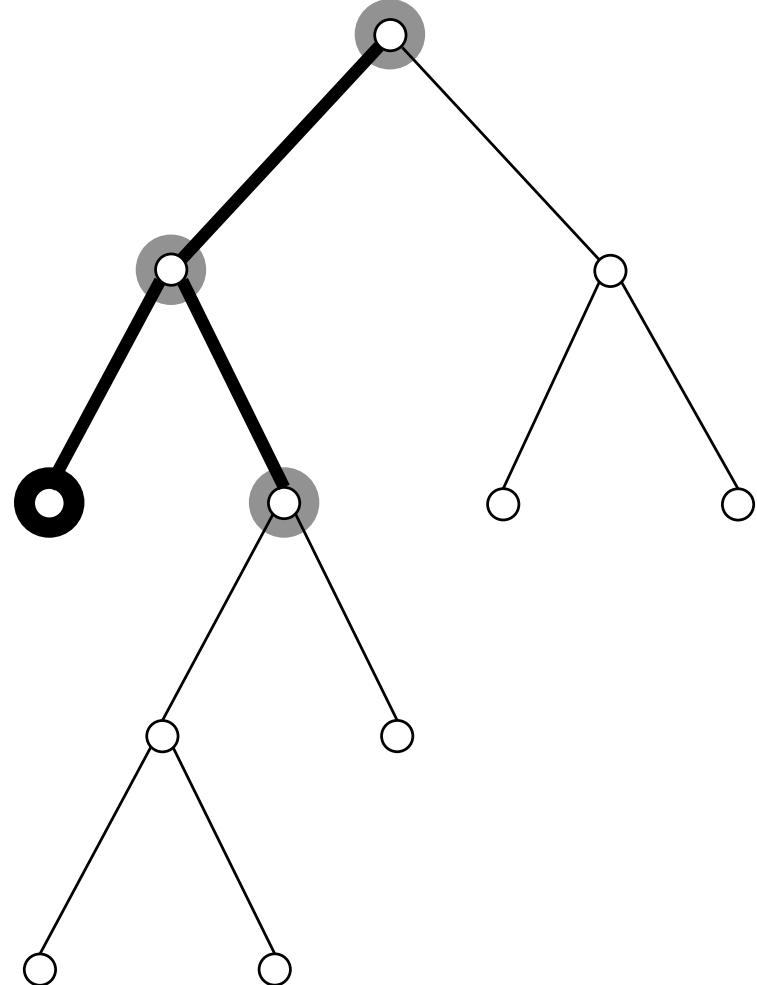


# Busca em profundidade

## Ideia em árvores

1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo**  $w$ .

2. Quando alcança, volta para o vértice anteriormente visitado  $v$  e segue **outro caminho** por outra sub-árvore de  $v$ , da mesma maneira.

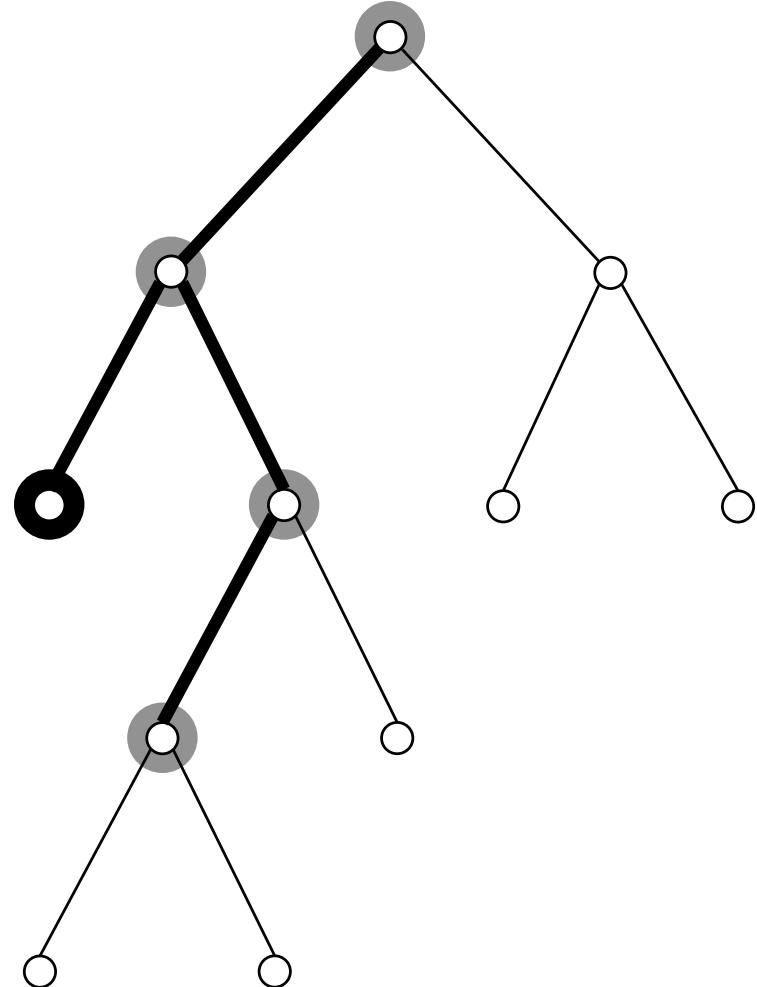


# Busca em profundidade

## Ideia em árvores

1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo**  $w$ .

2. Quando alcança, volta para o vértice anteriormente visitado  $v$  e segue **outro caminho** por outra sub-árvore de  $v$ , da mesma maneira.

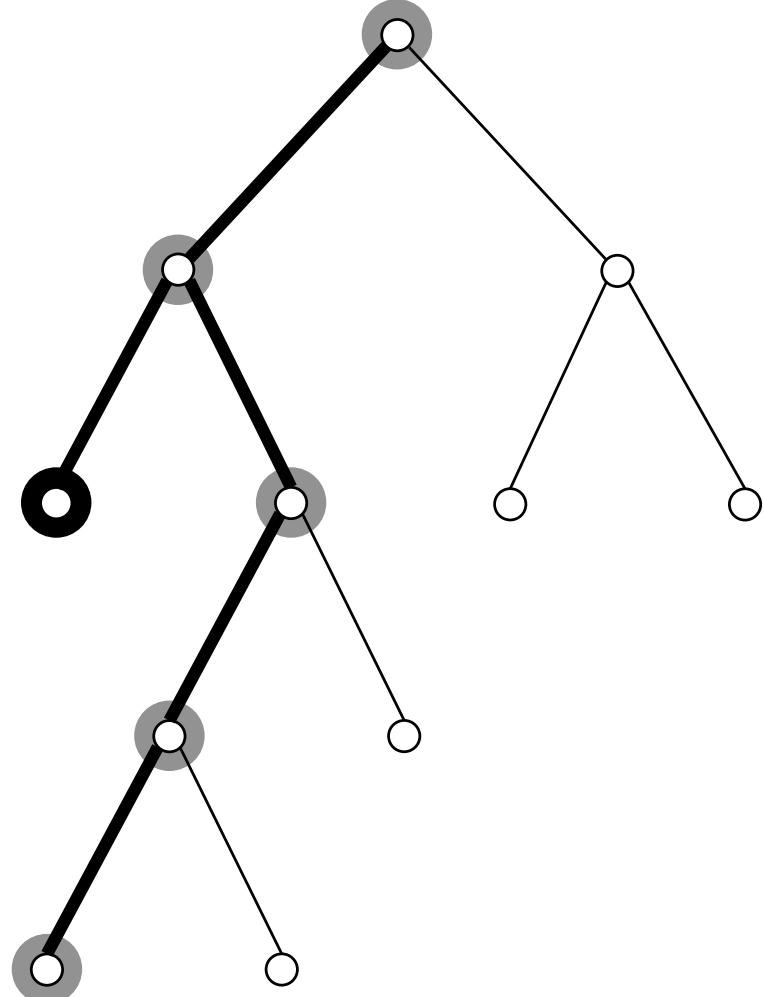


# Busca em profundidade

## Ideia em árvores

1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo**  $w$ .

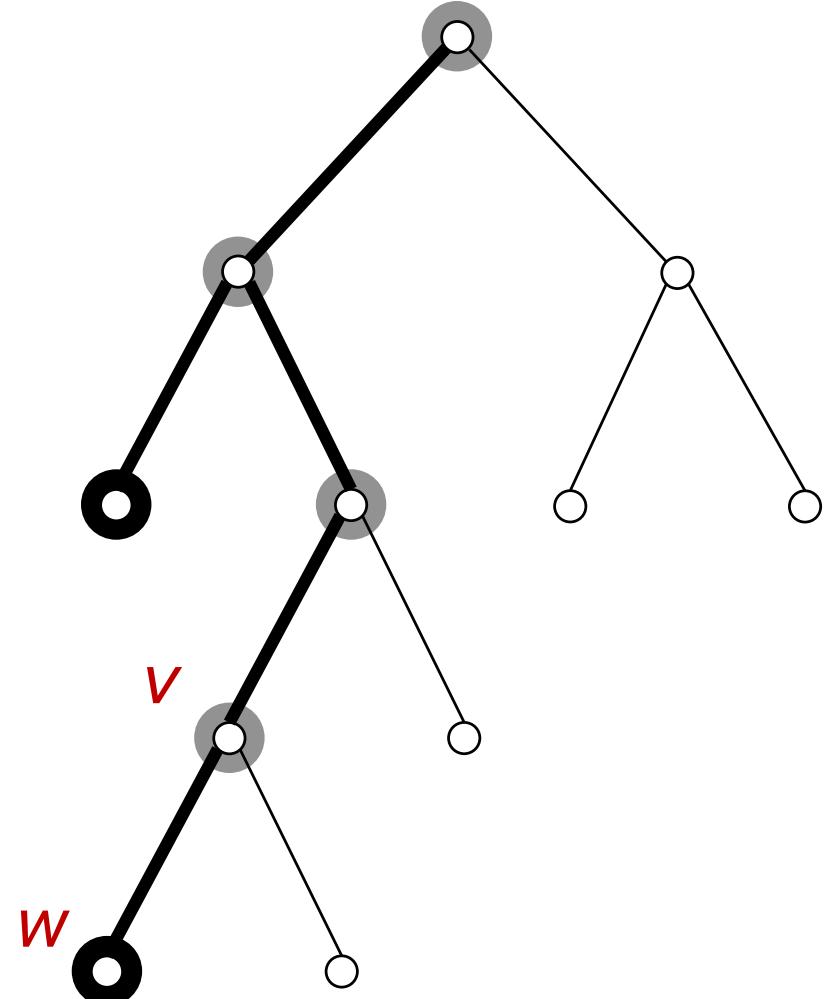
2. Quando alcança, volta para o vértice anteriormente visitado  $v$  e segue **outro caminho** por outra sub-árvore de  $v$ , da mesma maneira.



# Busca em profundidade

## Ideia em árvores

1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo**  $w$ .
2. Quando alcança, volta para o vértice anteriormente visitado  $v$  e segue **outro caminho** por outra sub-árvore de  $v$ , da mesma maneira.

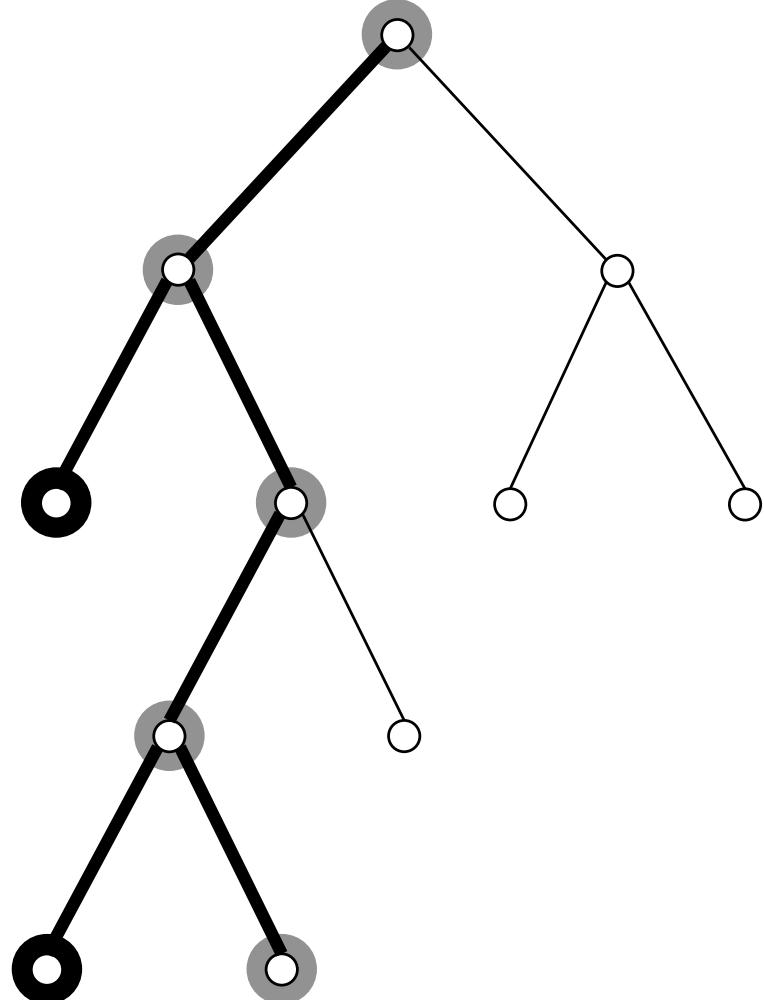


# Busca em profundidade

## Ideia em árvores

1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo**  $w$ .

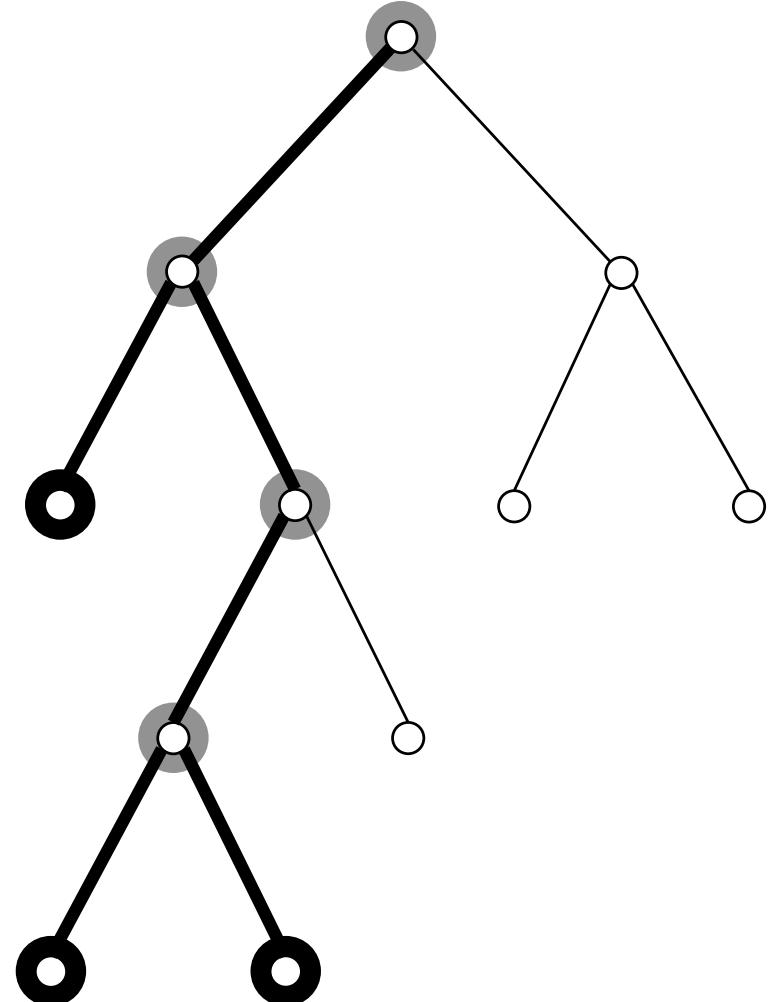
2. Quando alcança, volta para o vértice anteriormente visitado  $v$  e segue **outro caminho** por outra sub-árvore de  $v$ , da mesma maneira.



# Busca em profundidade

## Ideia em árvores

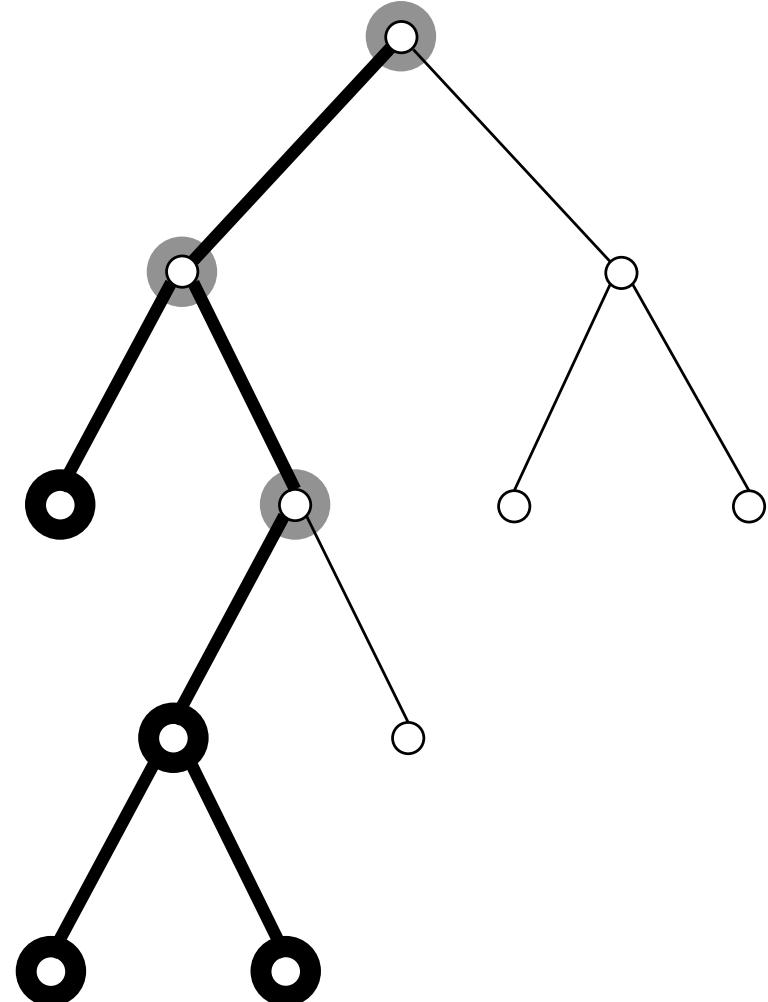
1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo**  $w$ .
2. Quando alcança, volta para o vértice anteriormente visitado  $v$  e segue **outro caminho** por outra sub-árvore de  $v$ , da mesma maneira.



# Busca em profundidade

## Ideia em árvores

1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo**  $w$ .
2. Quando alcança, volta para o vértice anteriormente visitado  $v$  e segue **outro caminho** por outra sub-árvore de  $v$ , da mesma maneira.

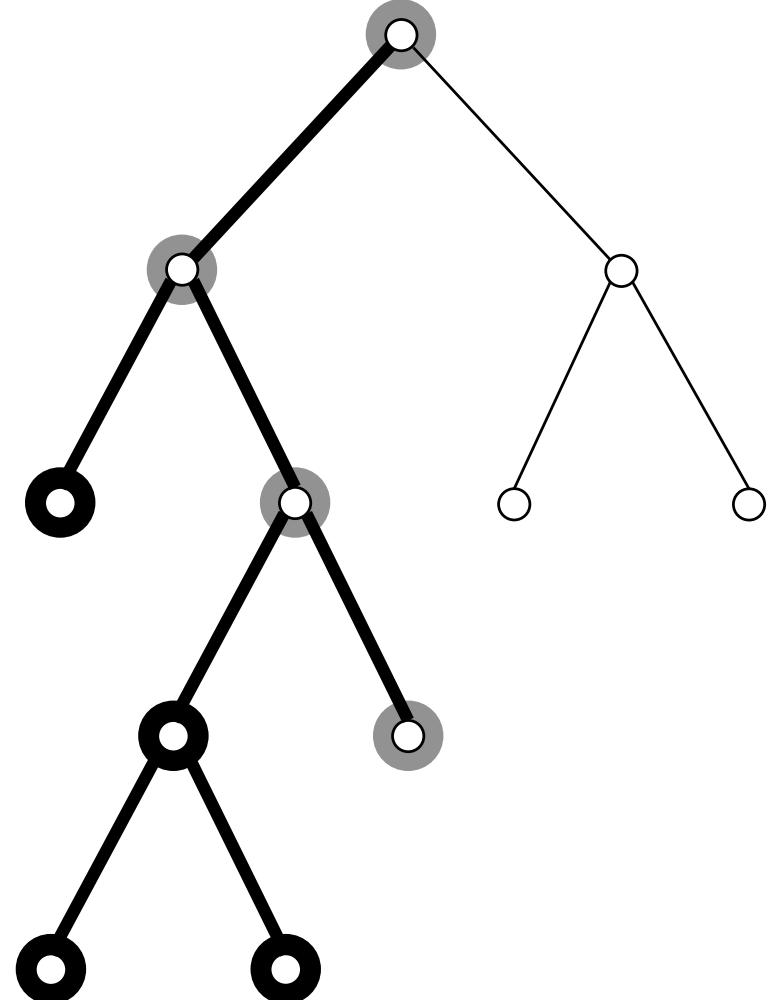


# Busca em profundidade

## Ideia em árvores

1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo**  $w$ .

2. Quando alcança, volta para o vértice anteriormente visitado  $v$  e segue **outro caminho** por outra sub-árvore de  $v$ , da mesma maneira.

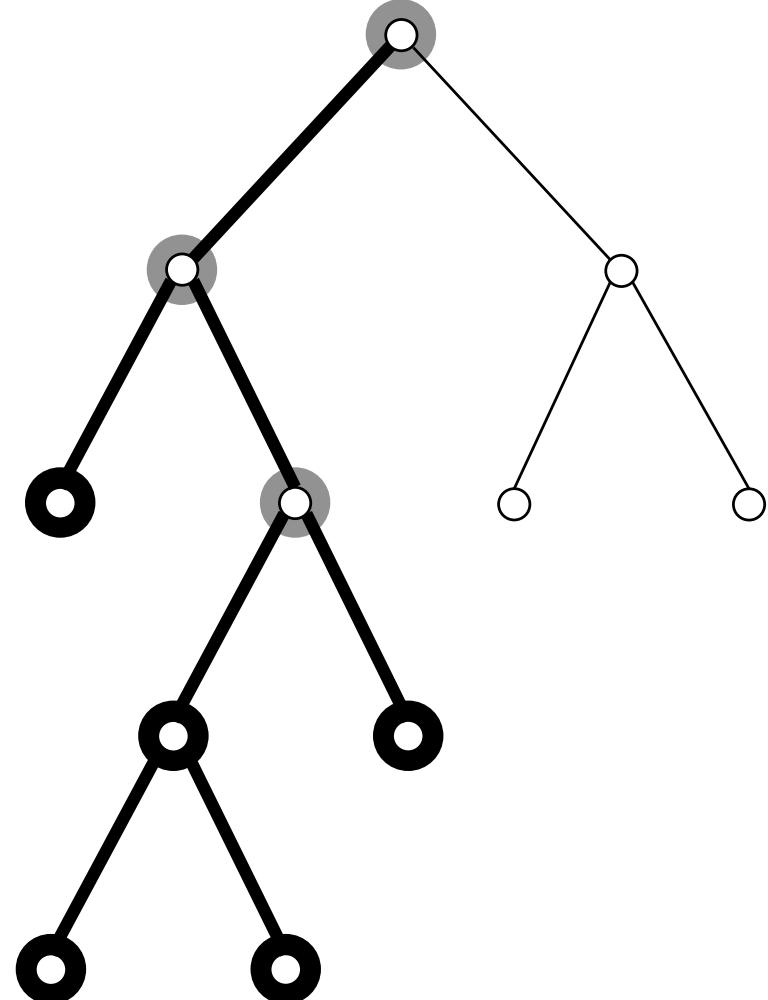


# Busca em profundidade

## Ideia em árvores

1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo**  $w$ .

2. Quando alcança, volta para o vértice anteriormente visitado  $v$  e segue **outro caminho** por outra sub-árvore de  $v$ , da mesma maneira.

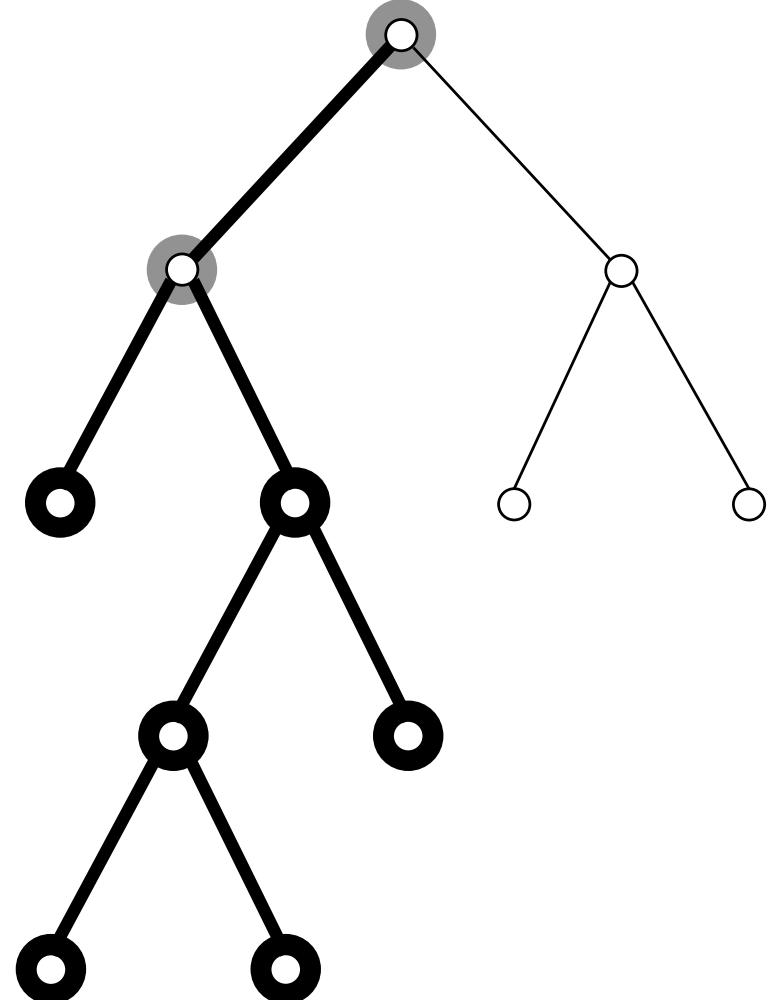


# Busca em profundidade

## Ideia em árvores

1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo**  $w$ .

2. Quando alcança, volta para o vértice anteriormente visitado  $v$  e segue **outro caminho** por outra sub-árvore de  $v$ , da mesma maneira.

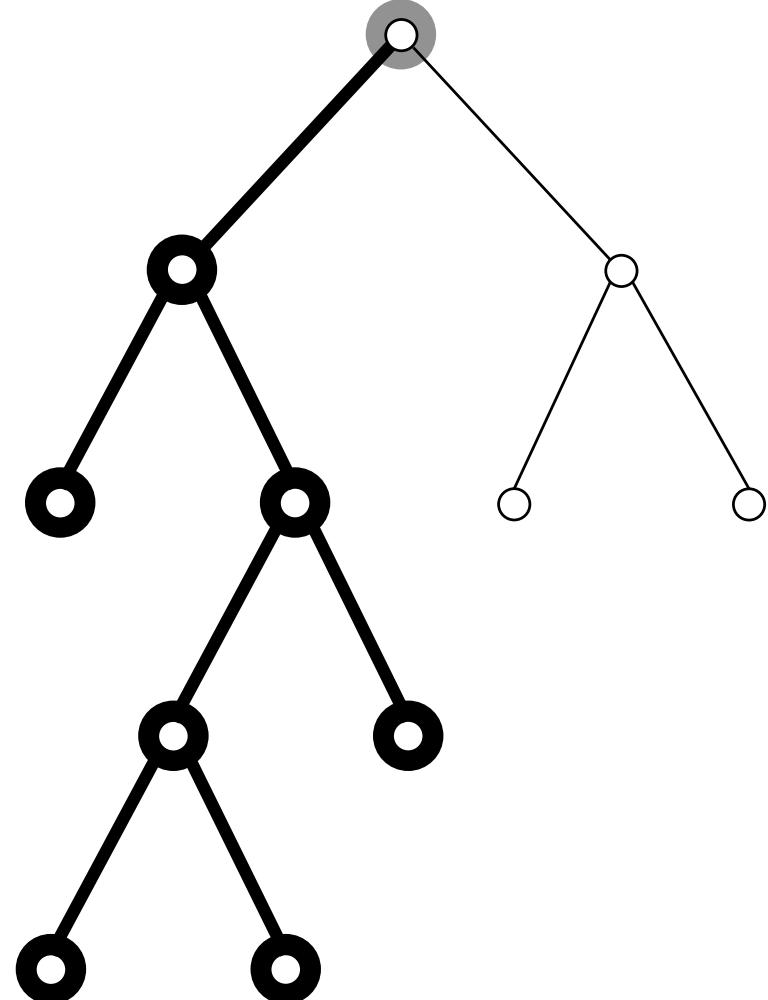


# Busca em profundidade

## Ideia em árvores

1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo**  $w$ .

2. Quando alcança, volta para o vértice anteriormente visitado  $v$  e segue **outro caminho** por outra sub-árvore de  $v$ , da mesma maneira.

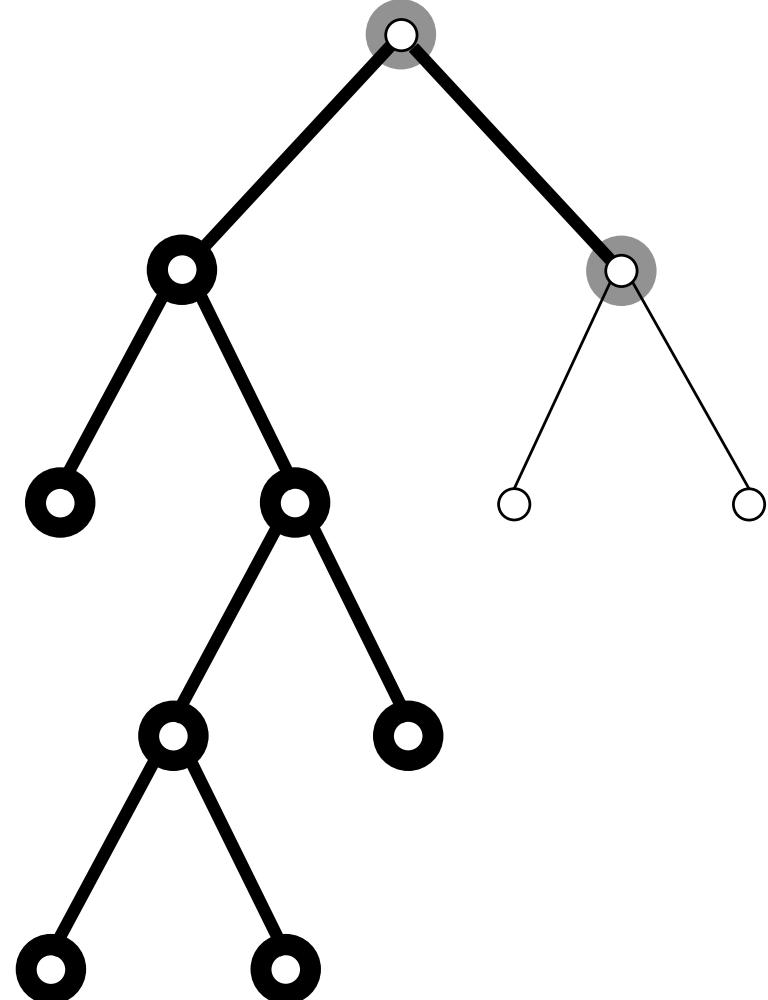


# Busca em profundidade

## Ideia em árvores

1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo**  $w$ .

2. Quando alcança, volta para o vértice anteriormente visitado  $v$  e segue **outro caminho** por outra sub-árvore de  $v$ , da mesma maneira.

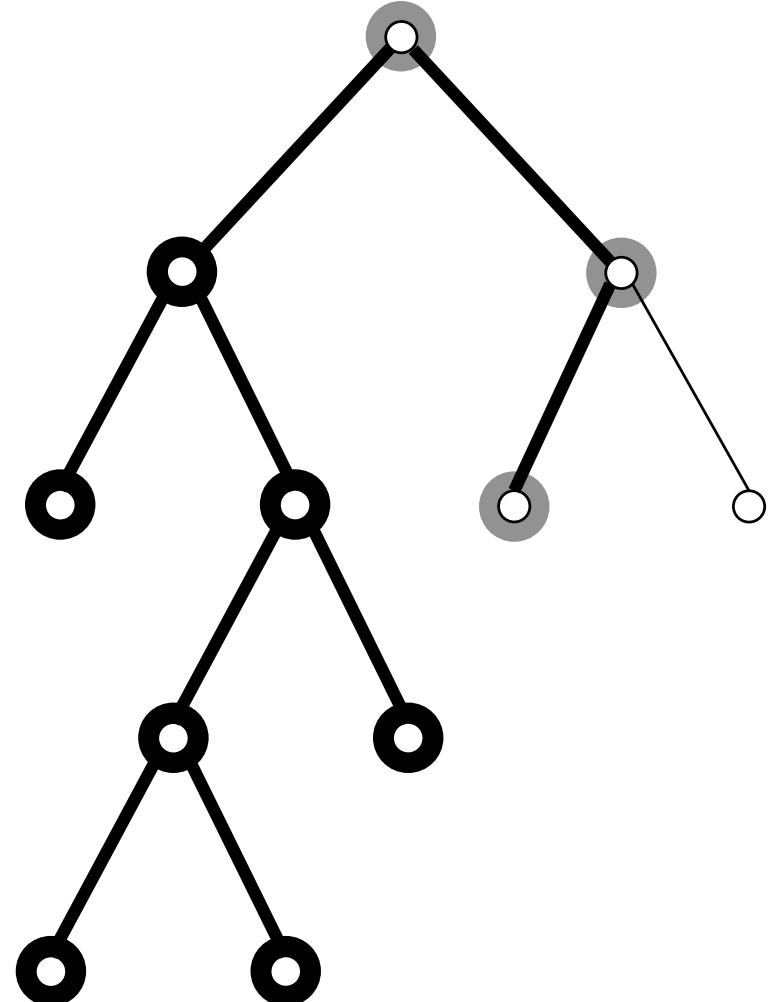


# Busca em profundidade

## Ideia em árvores

1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo**  $w$ .

2. Quando alcança, volta para o vértice anteriormente visitado  $v$  e segue **outro caminho** por outra sub-árvore de  $v$ , da mesma maneira.

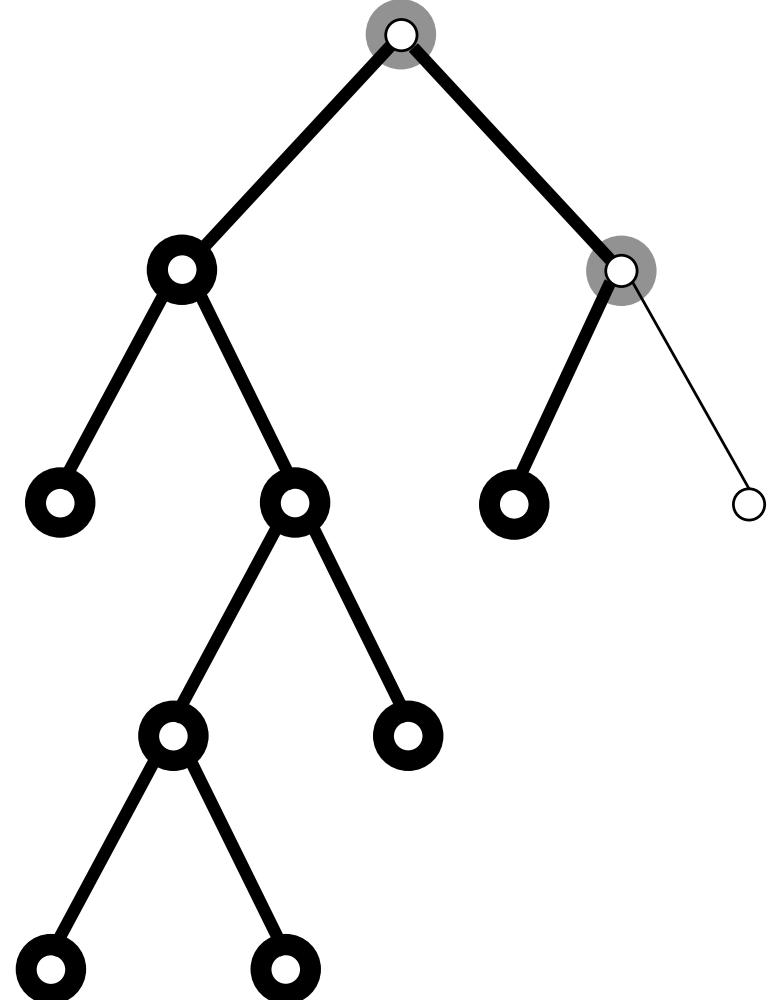


# Busca em profundidade

## Ideia em árvores

1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo**  $w$ .

2. Quando alcança, volta para o vértice anteriormente visitado  $v$  e segue **outro caminho** por outra sub-árvore de  $v$ , da mesma maneira.

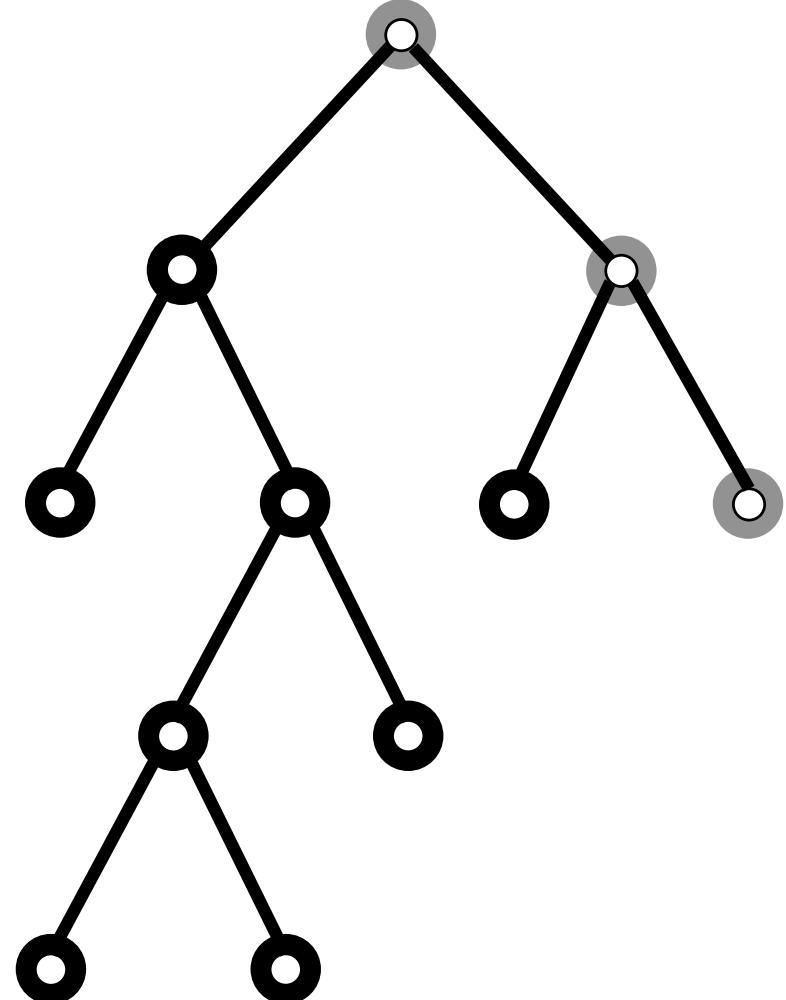


# Busca em profundidade

## Ideia em árvores

1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo**  $w$ .

2. Quando alcança, volta para o vértice anteriormente visitado  $v$  e segue **outro caminho** por outra sub-árvore de  $v$ , da mesma maneira.

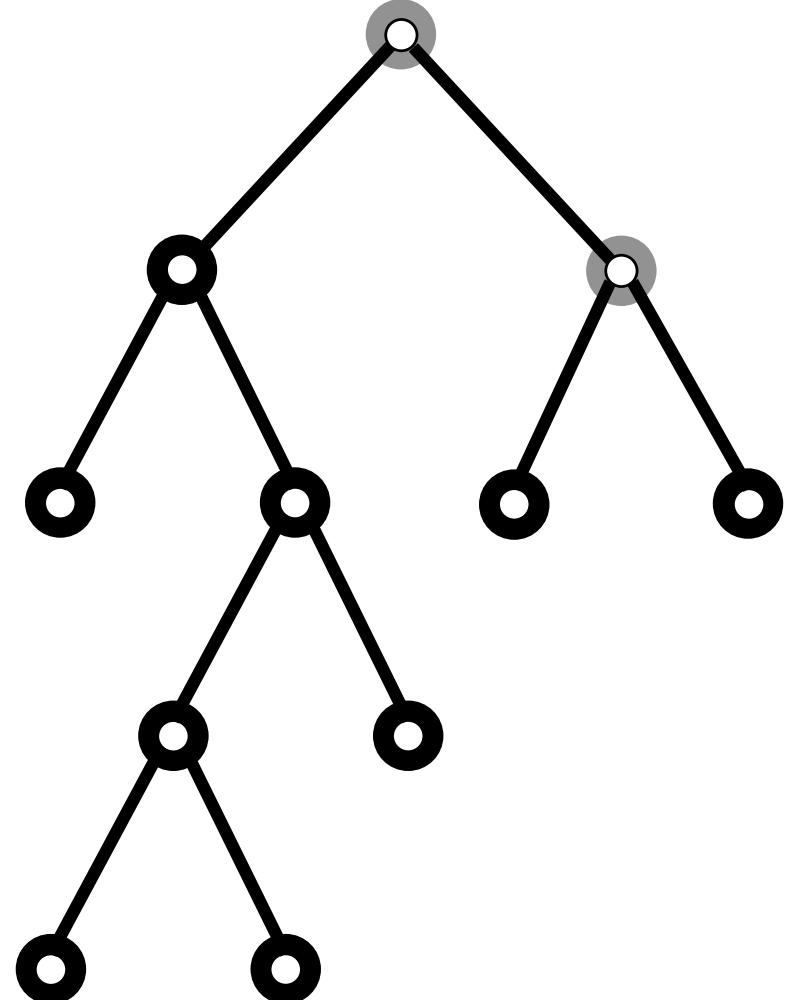


# Busca em profundidade

## Ideia em árvores

1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo**  $w$ .

2. Quando alcança, volta para o vértice anteriormente visitado  $v$  e segue **outro caminho** por outra sub-árvore de  $v$ , da mesma maneira.

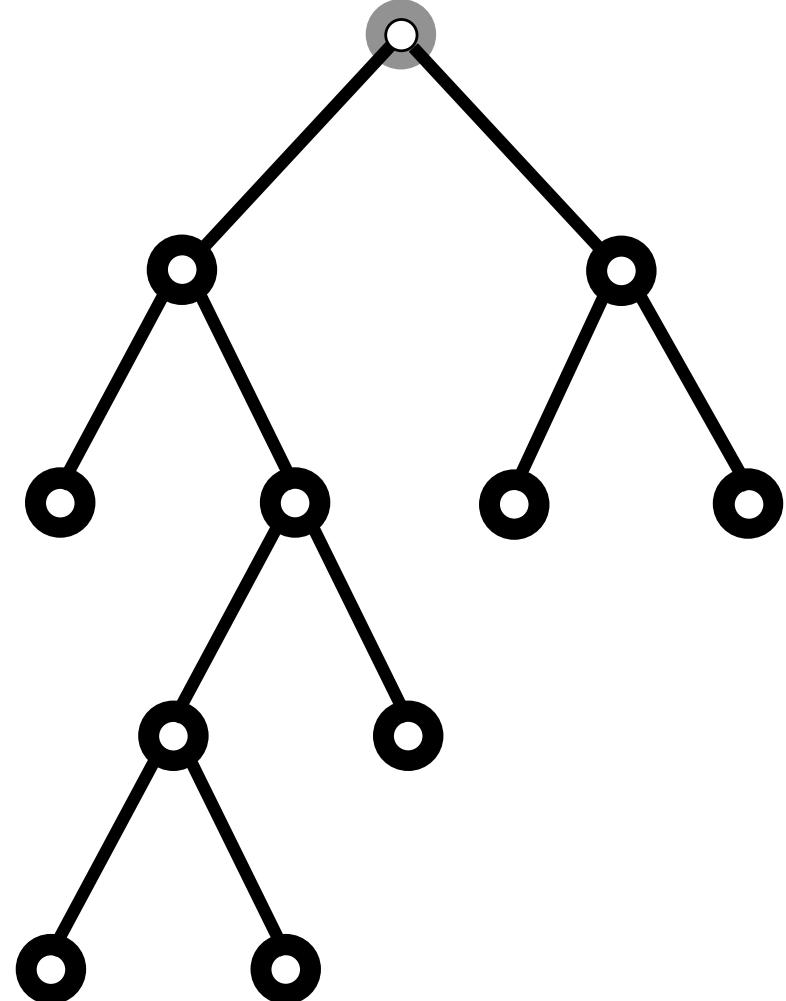


# Busca em profundidade

## Ideia em árvores

1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo**  $w$ .

2. Quando alcança, volta para o vértice anteriormente visitado  $v$  e segue **outro caminho** por outra sub-árvore de  $v$ , da mesma maneira.

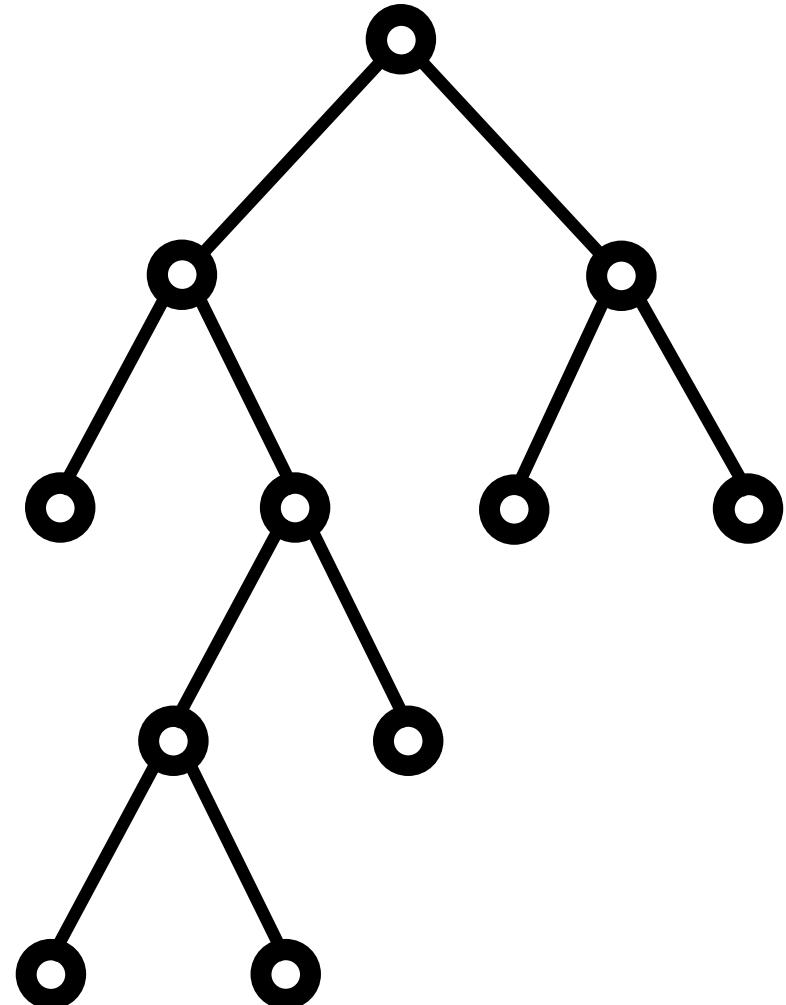


# Busca em profundidade

## Ideia em árvores

1. O algoritmo **caminha** na árvore buscando alcançar o seu vértice (nó) mais **profundo**  $w$ .

2. Quando alcança, volta para o vértice anteriormente visitado  $v$  e segue **outro caminho** por outra sub-árvore de  $v$ , da mesma maneira.

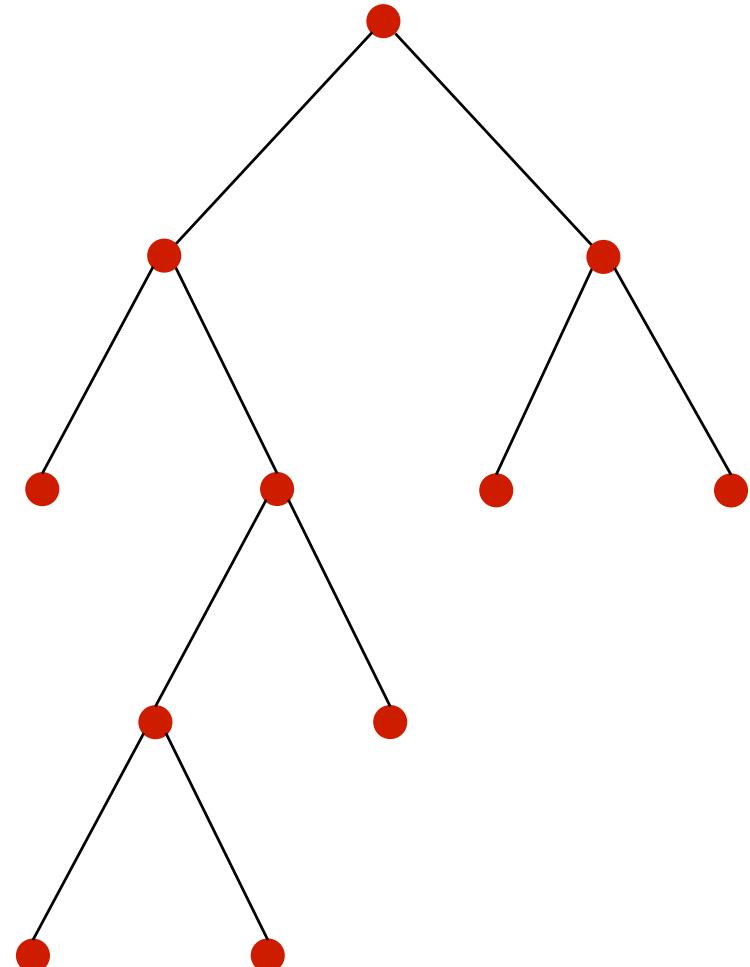


# Busca em profundidade

## Ideia em árvores

O algoritmo utiliza uma **pilha** para memorizar os vértices anteriores que deve visitar.

**Recursão** substitui uso direto da pilha.



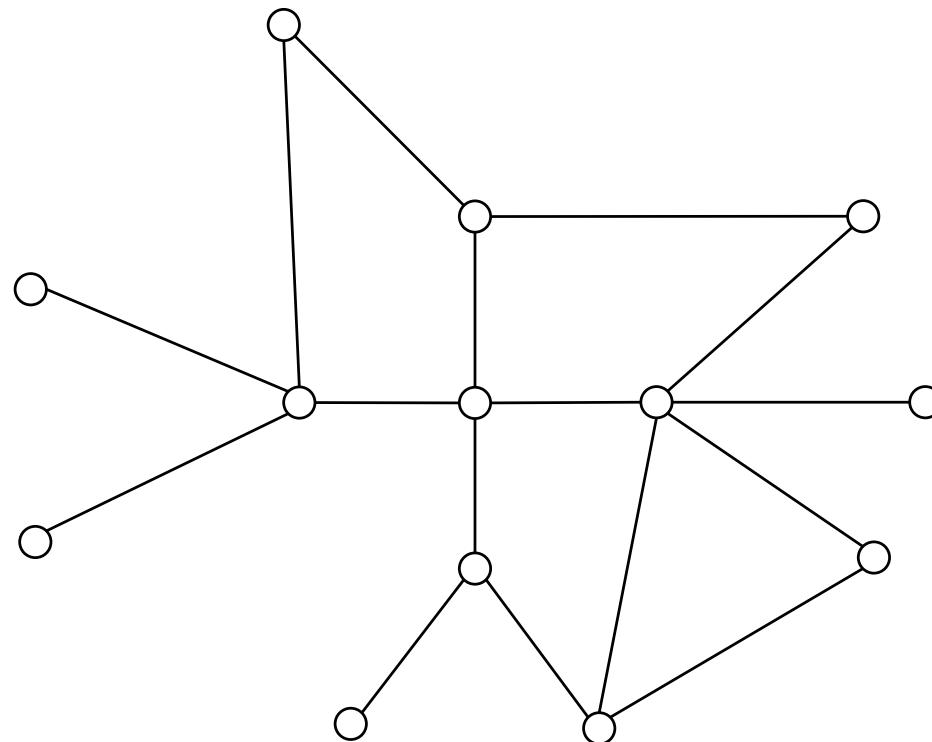
# Busca em profundidade

## Ideia em grafos

1. O algoritmo **caminha** no grafo buscando alcançar o seu **vértice mais distante** (“profundo”) **w**, sempre que possível.
2. Quando alcança, **volta** para o **vértice anteriormente visitado v** e segue **outro caminho** de algum adjacente de **v**, da mesma maneira.

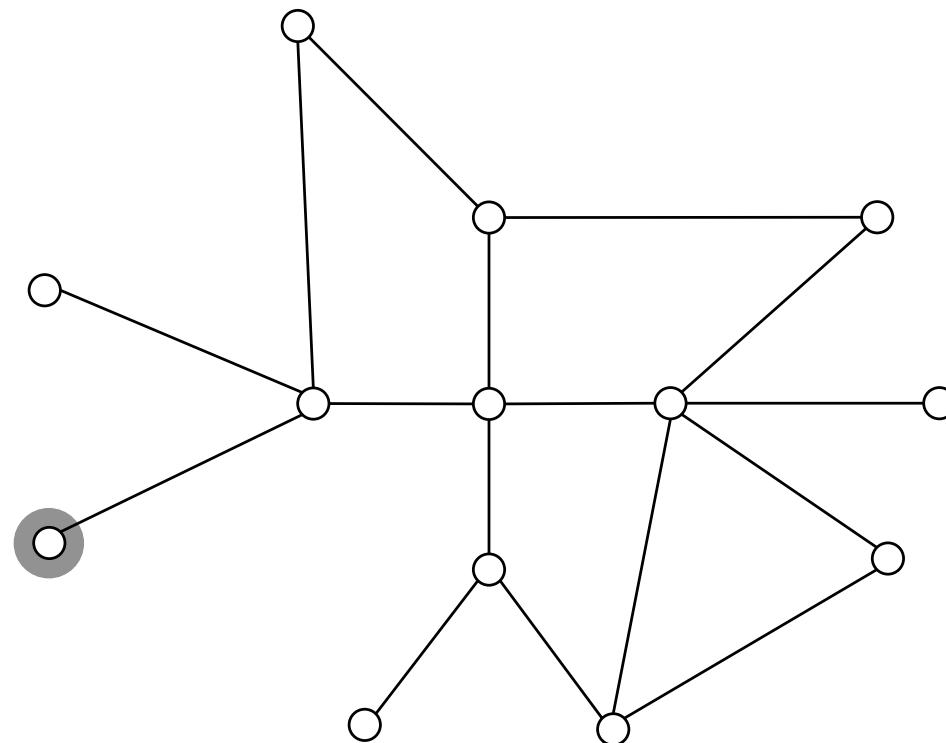
# Busca em profundidade

Cada vértice inicia **branco**.

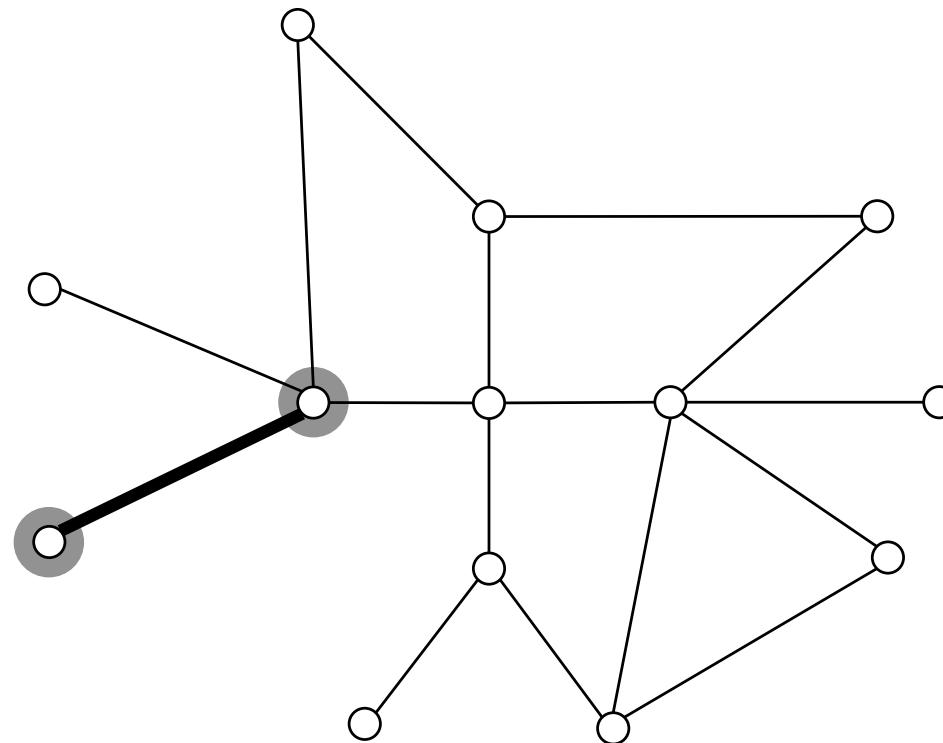


# Busca em profundidade

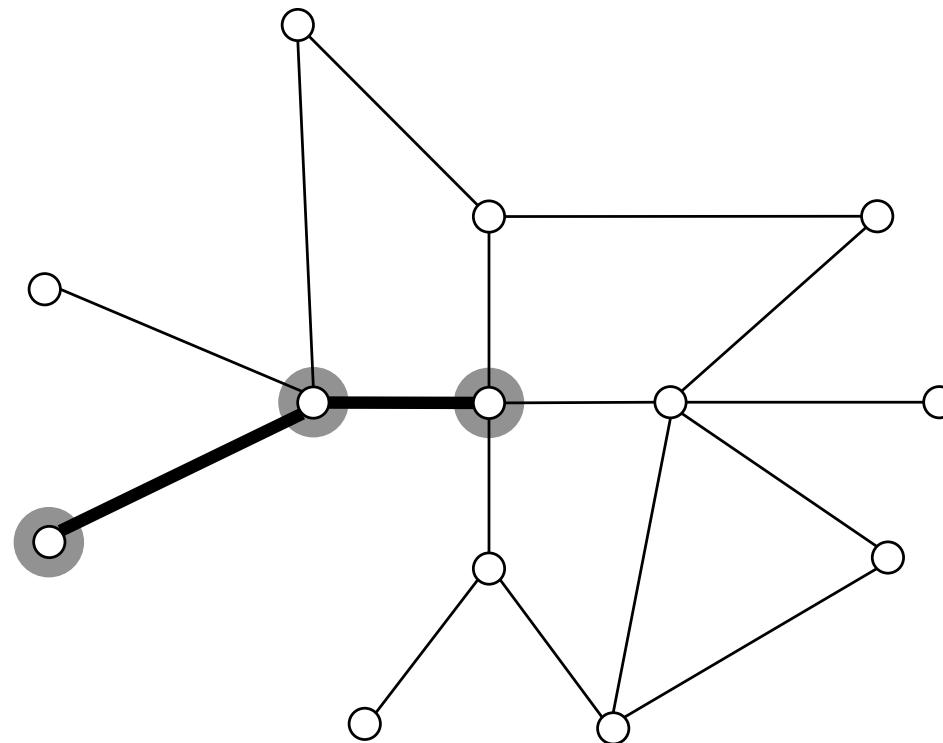
Cada vértice visitado (descoberto) fica **cinza**.



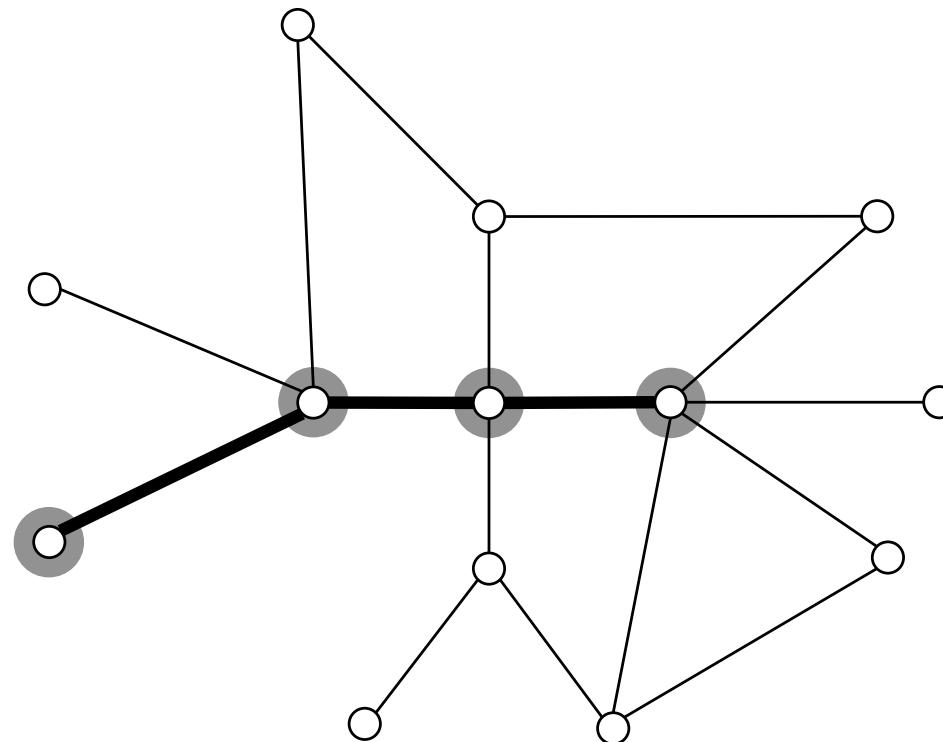
# Busca em profundidade



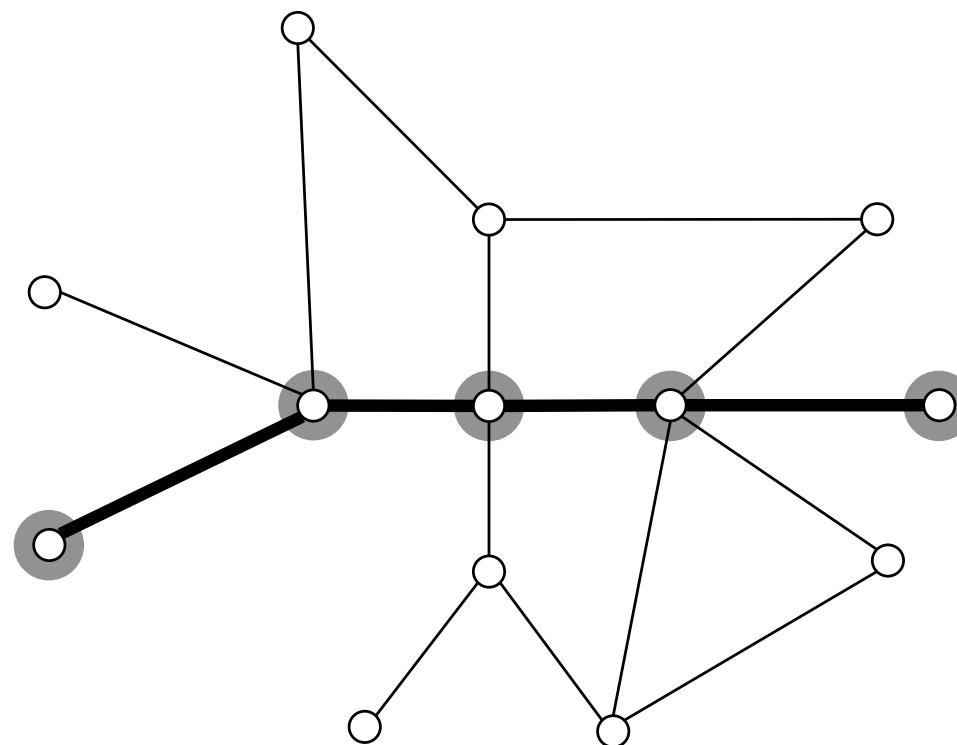
# Busca em profundidade



# Busca em profundidade

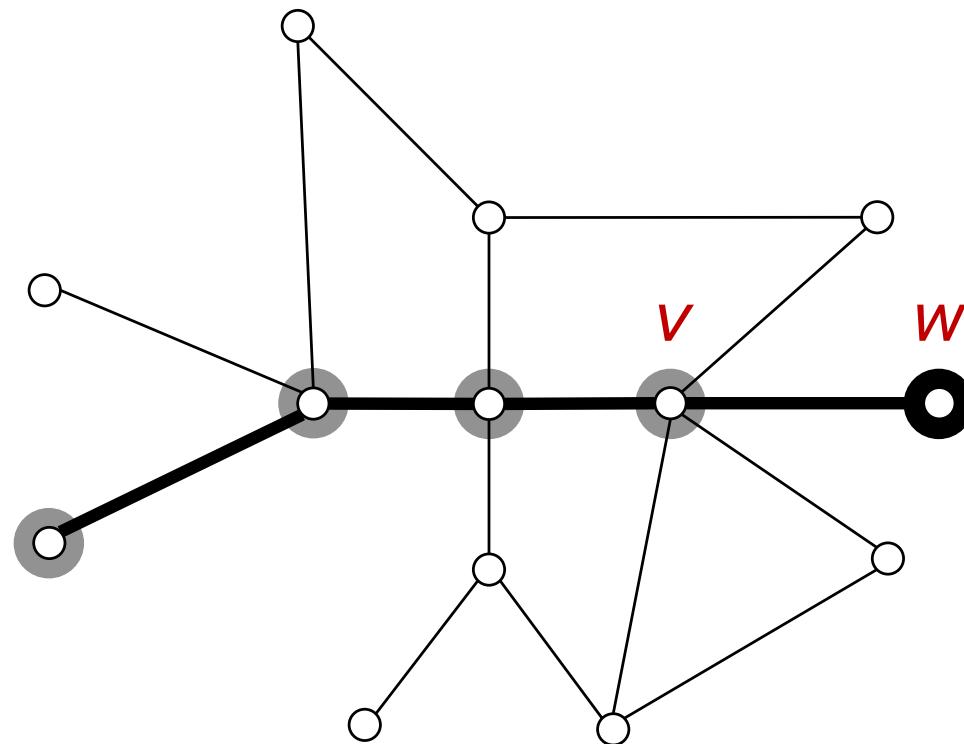


# Busca em profundidade

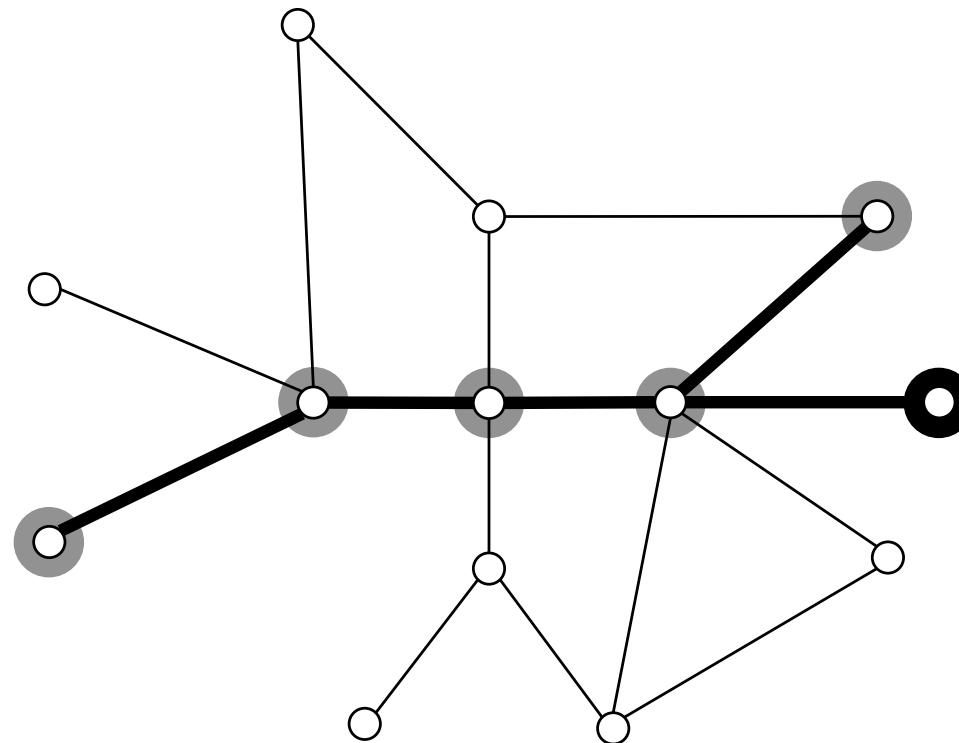


# Busca em profundidade

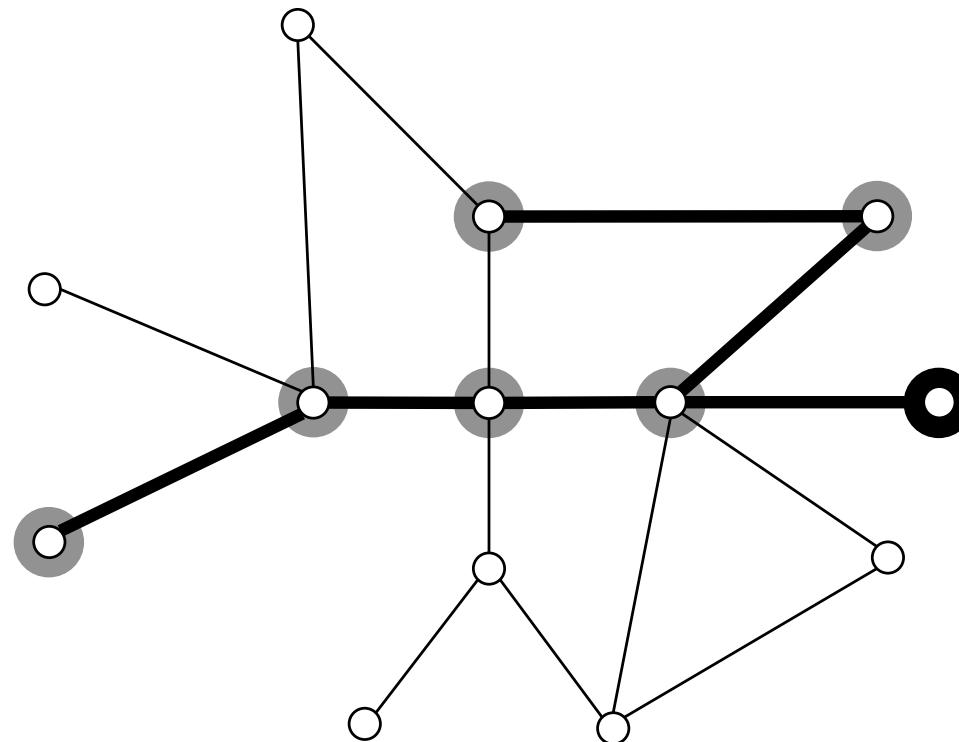
Vértice com adjacentes já visitados fica **preto**.



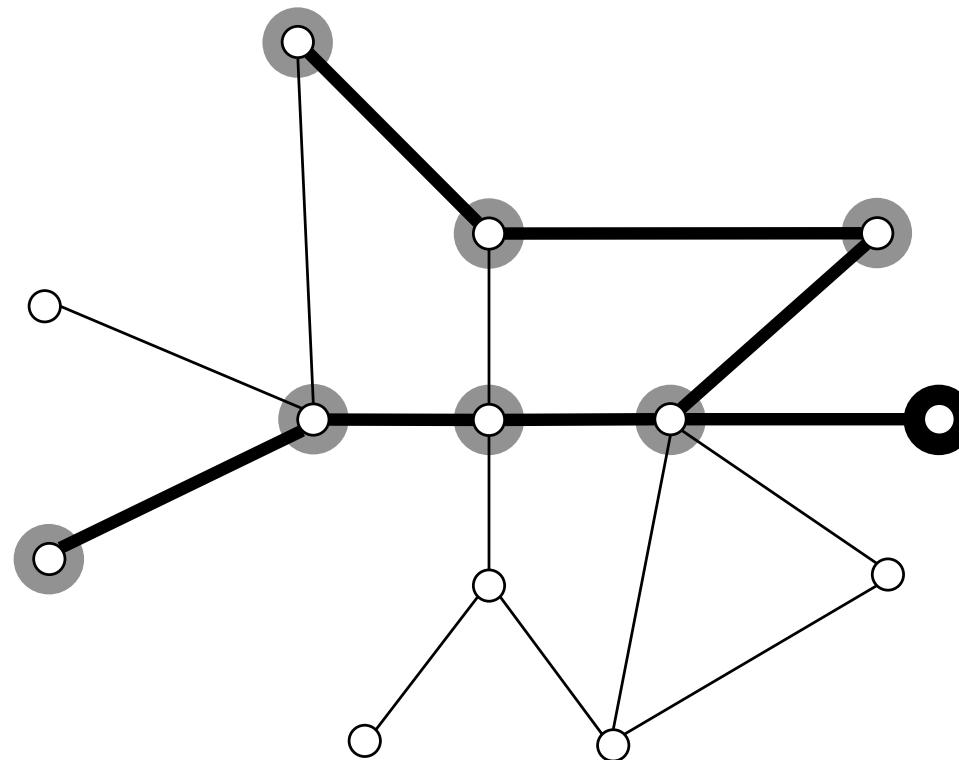
# Busca em profundidade



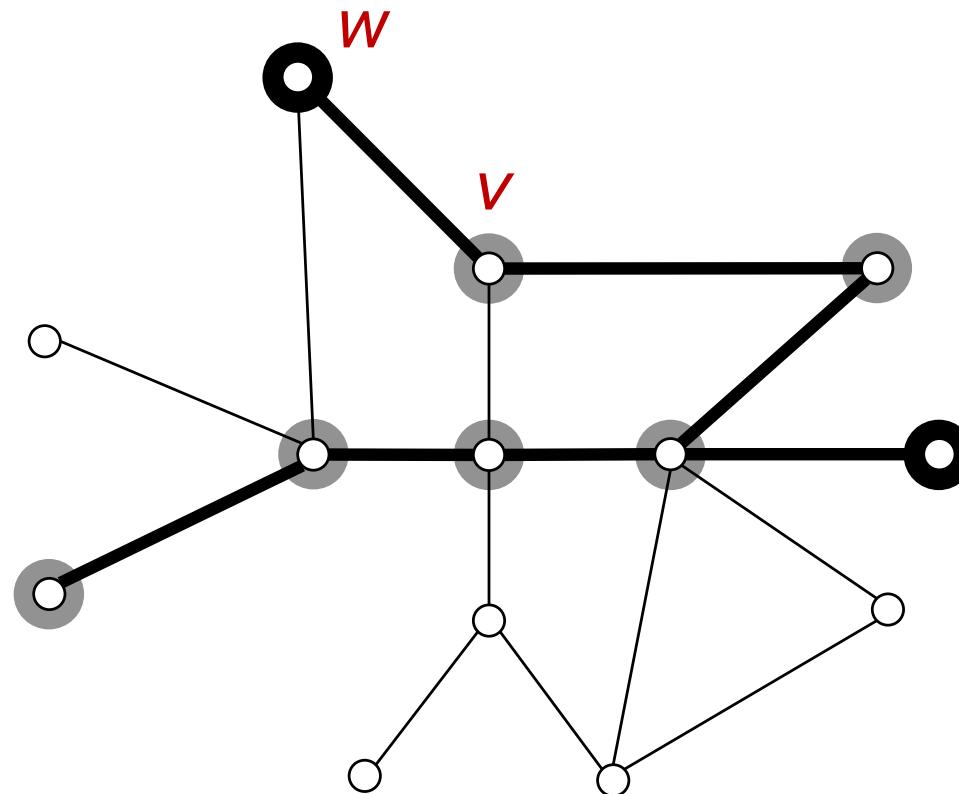
# Busca em profundidade



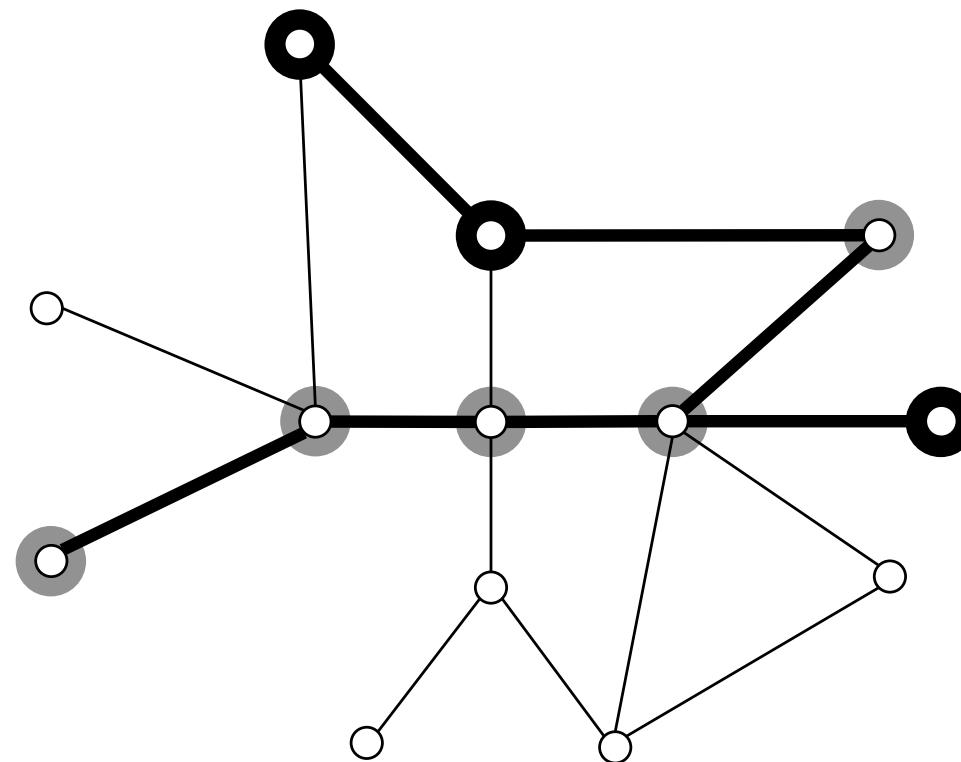
# Busca em profundidade



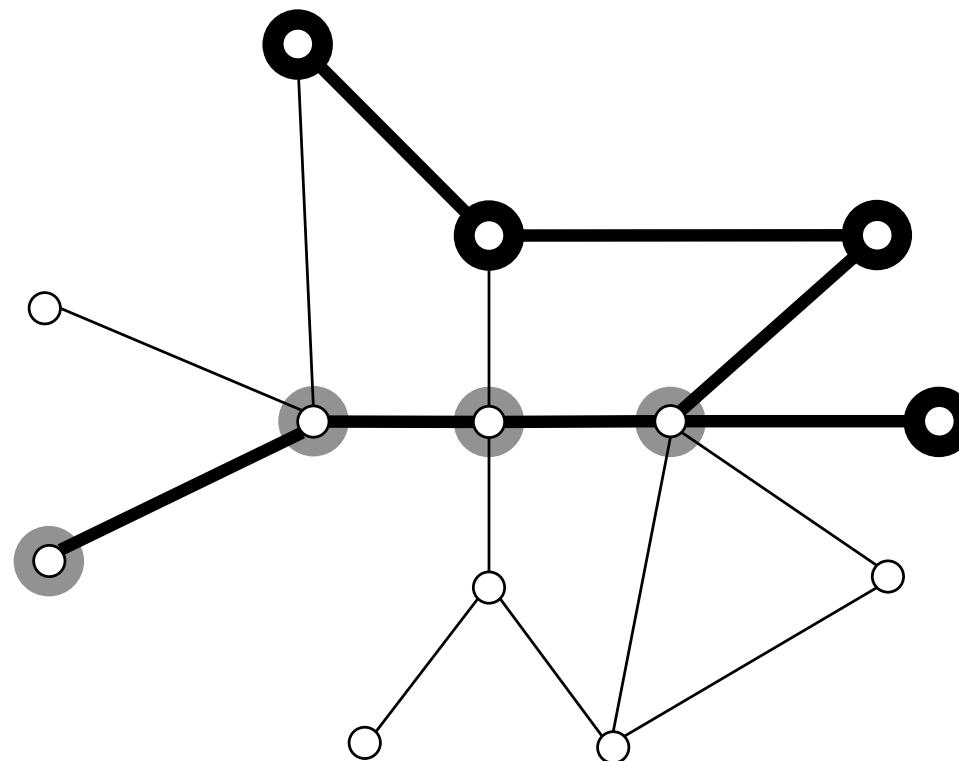
# Busca em profundidade



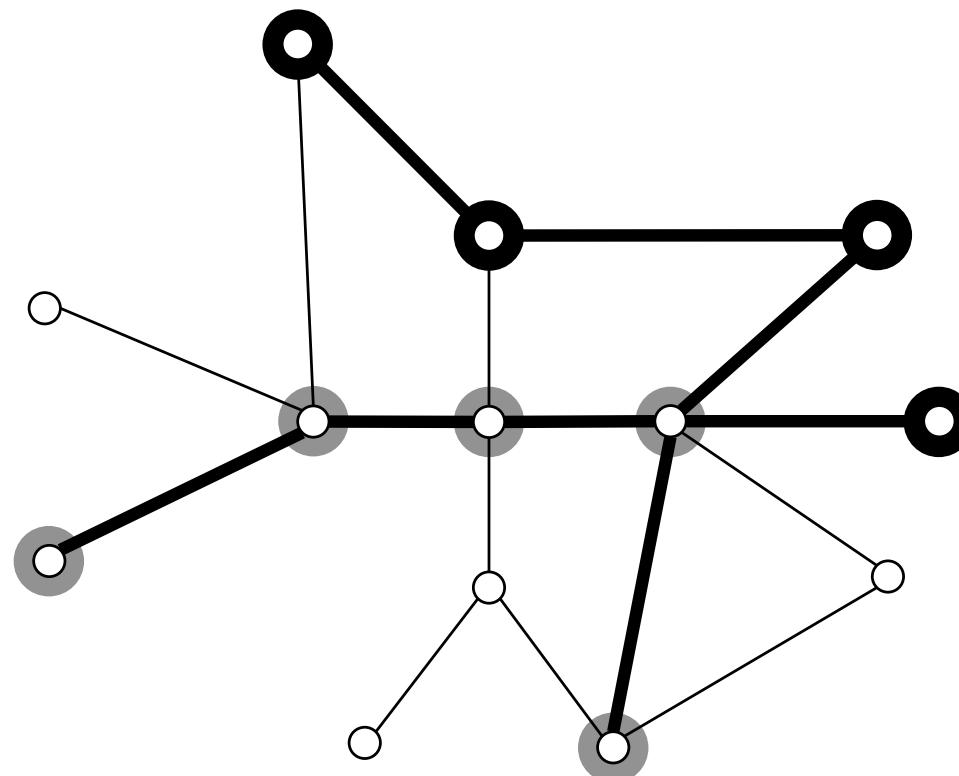
# Busca em profundidade



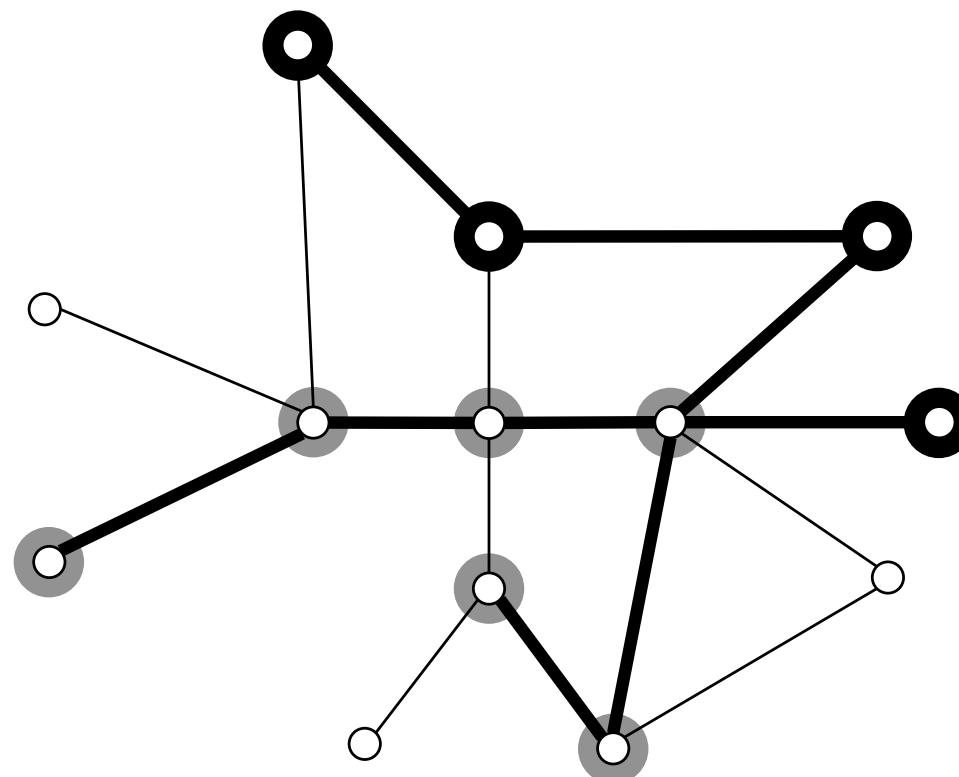
# Busca em profundidade



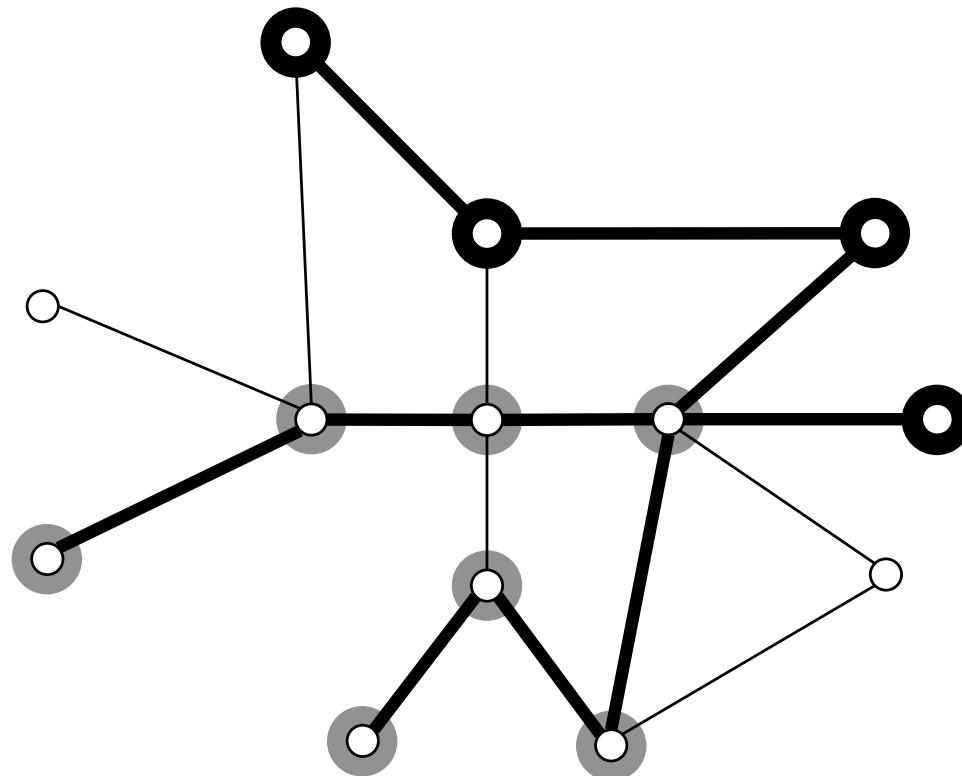
# Busca em profundidade



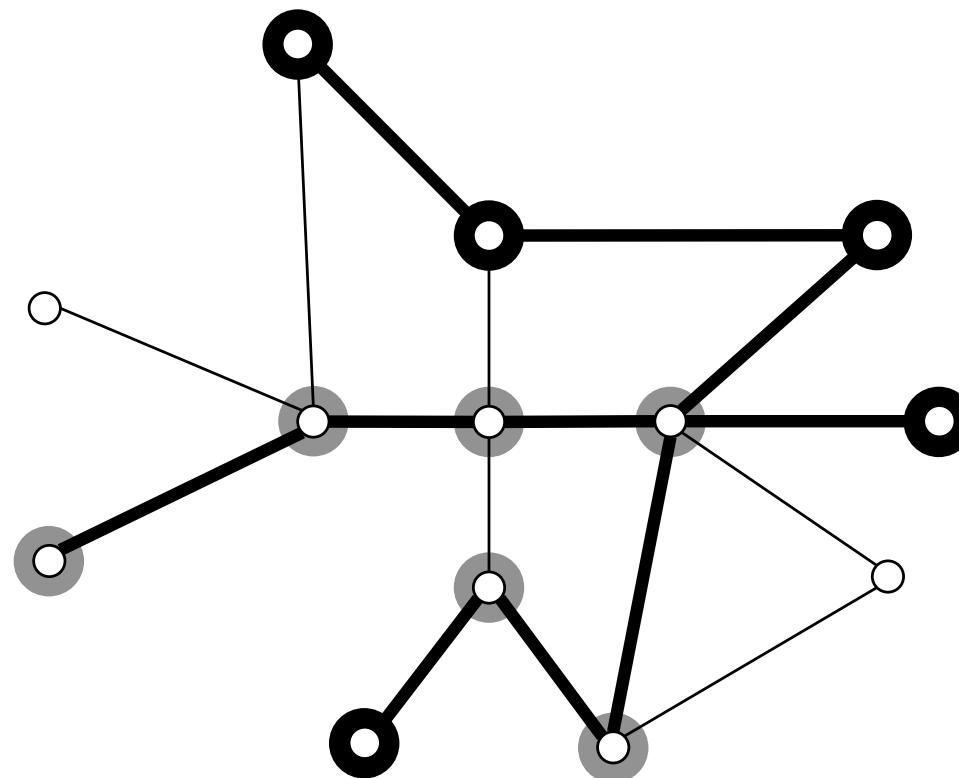
# Busca em profundidade



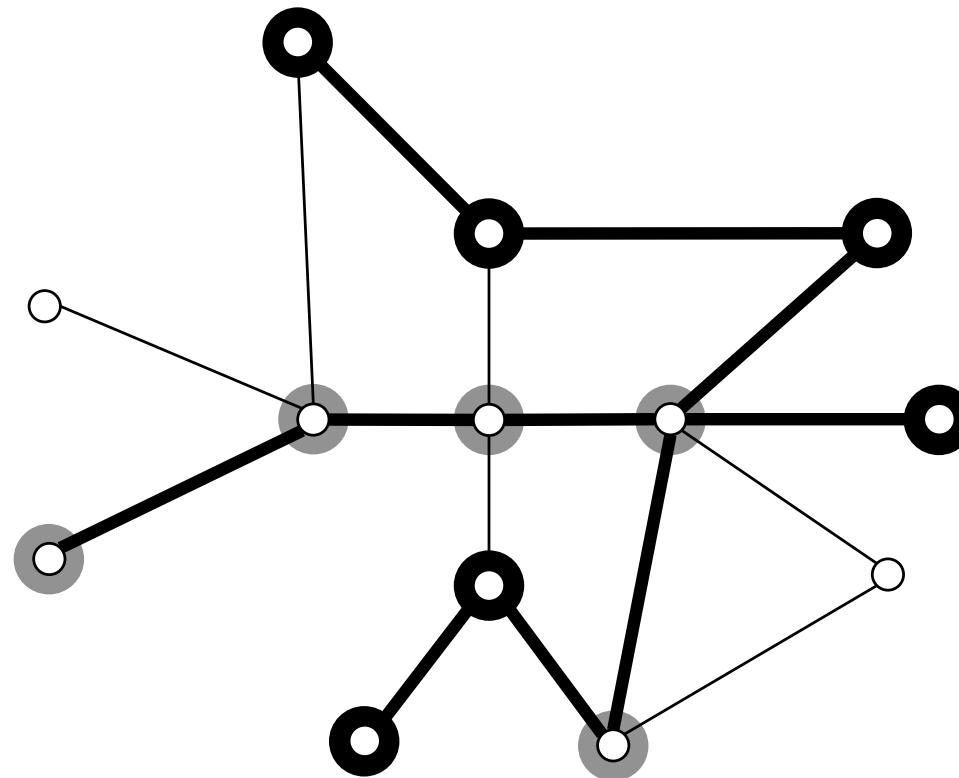
# Busca em profundidade



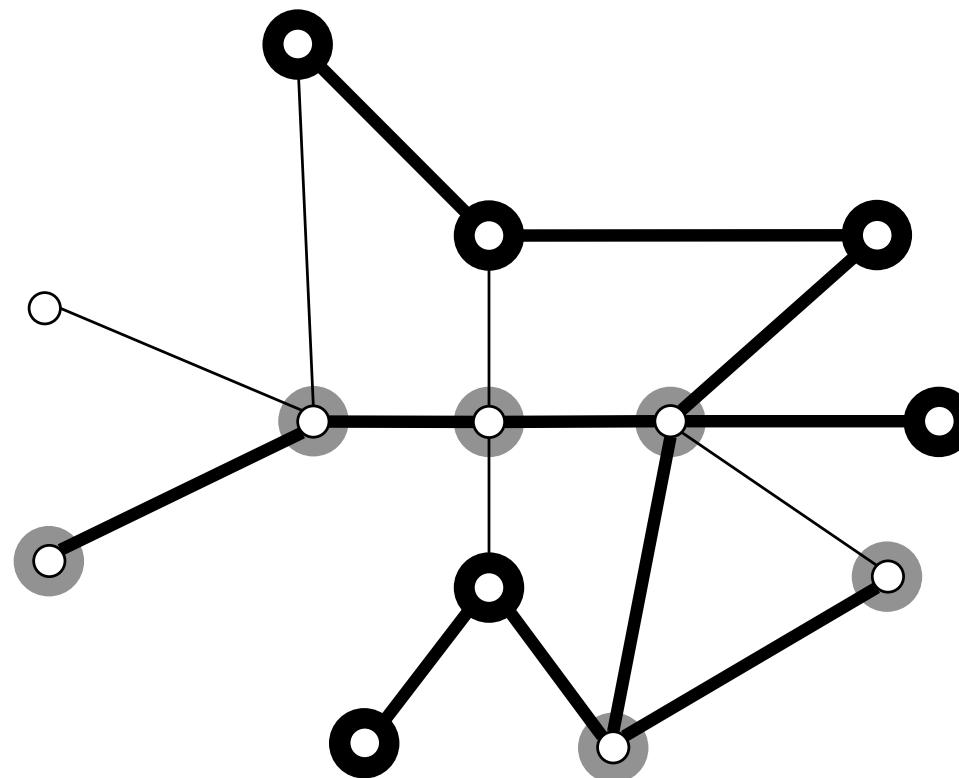
# Busca em profundidade



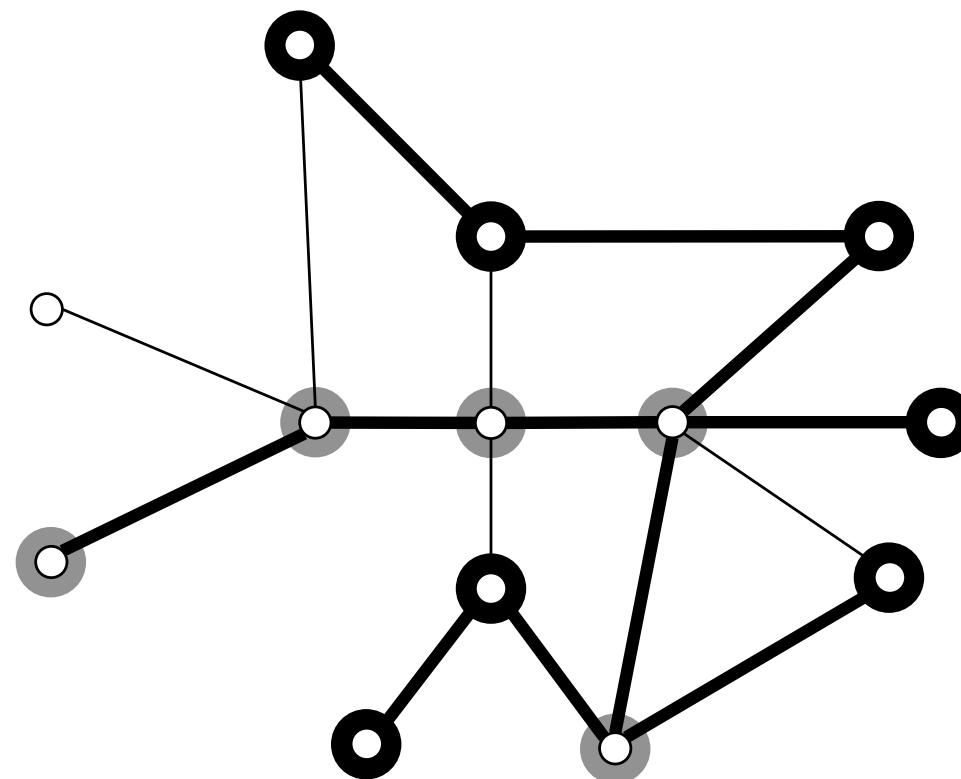
# Busca em profundidade



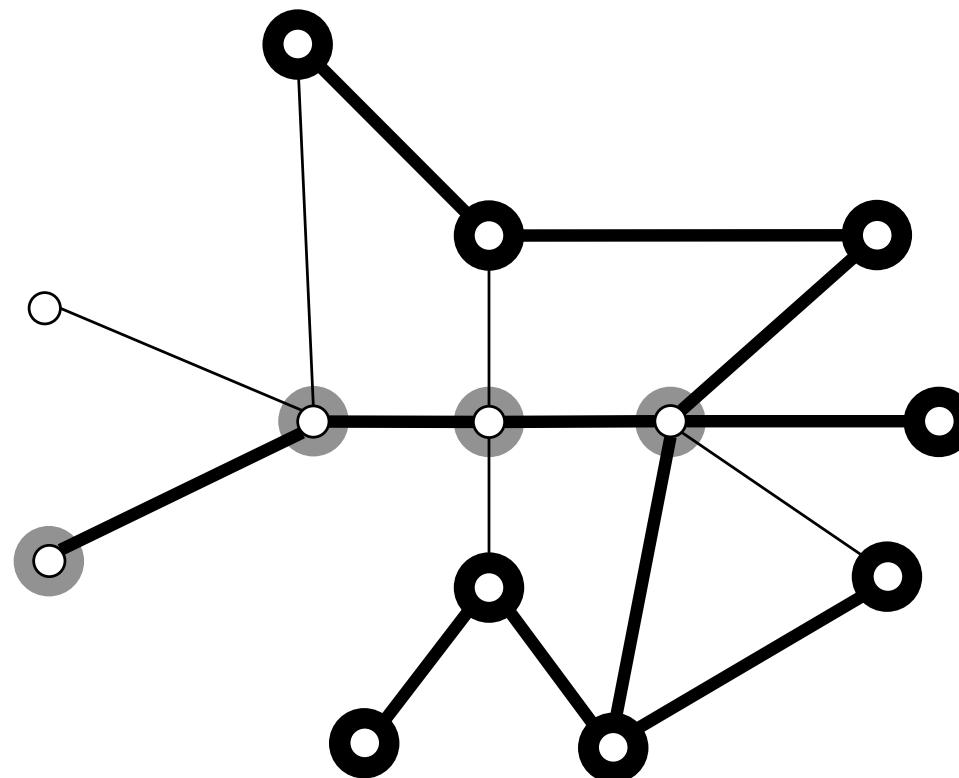
# Busca em profundidade



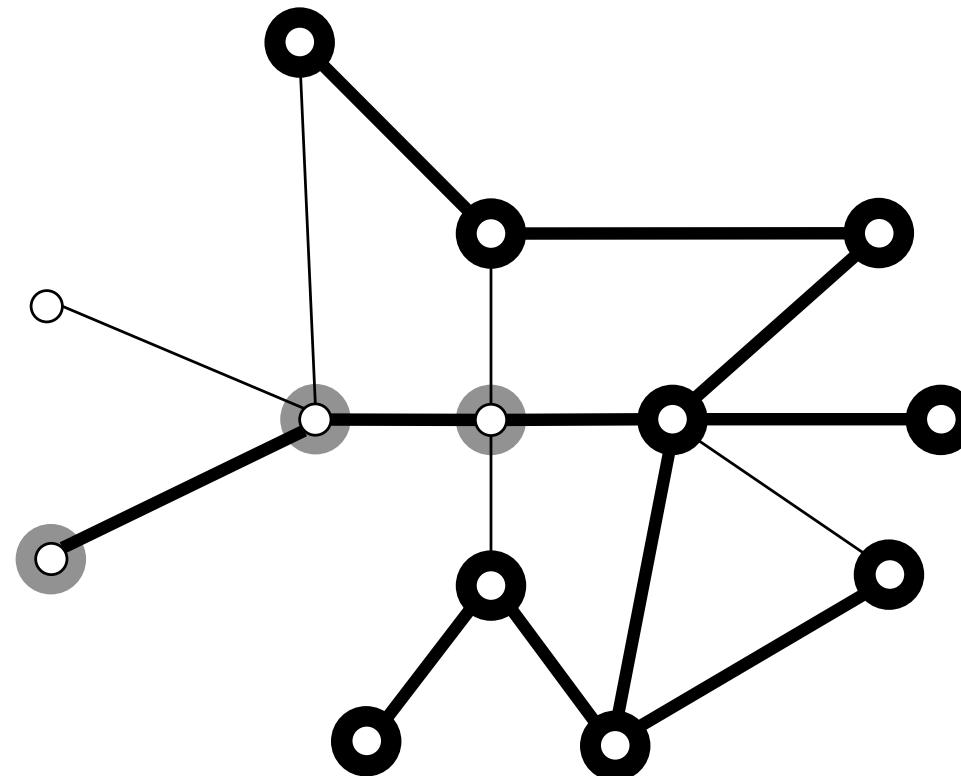
# Busca em profundidade



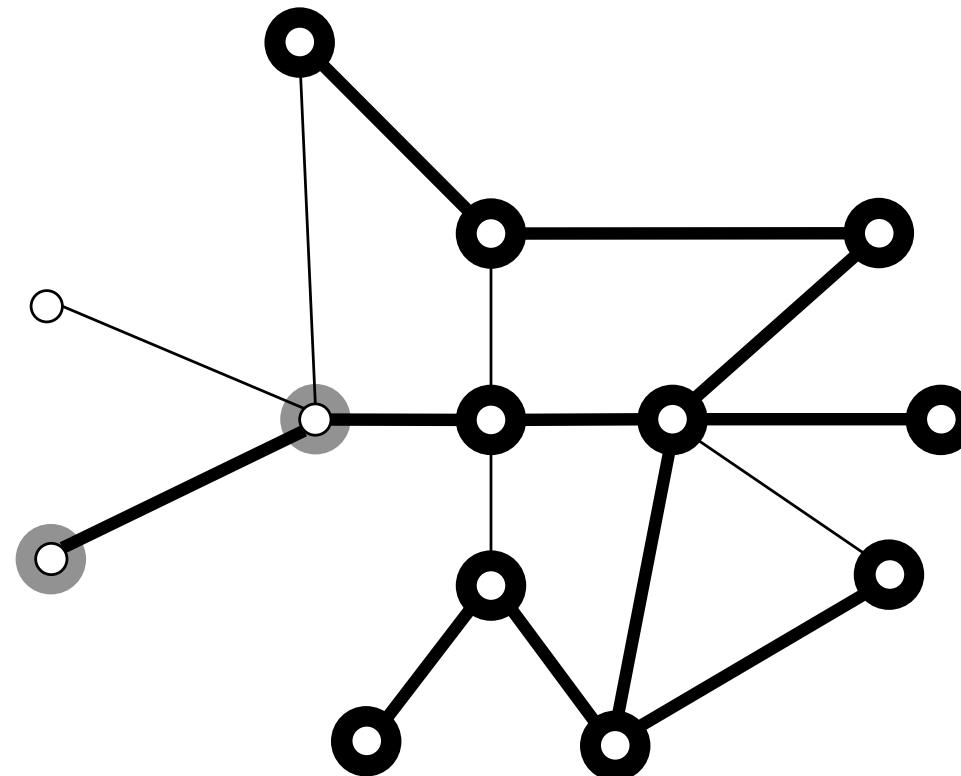
# Busca em profundidade



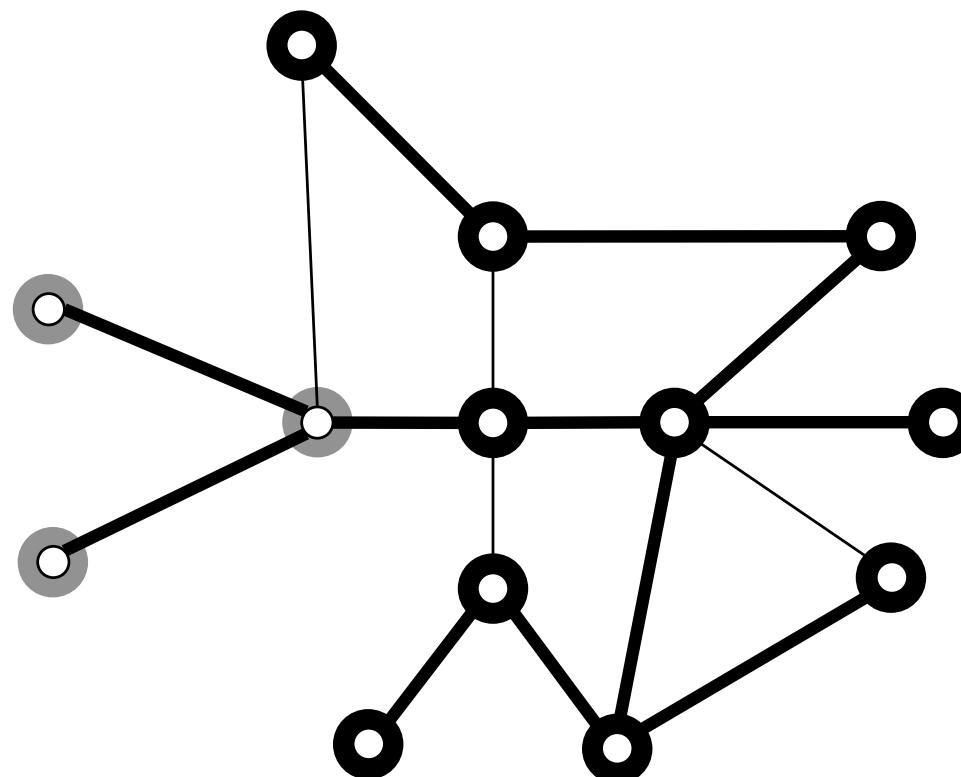
# Busca em profundidade



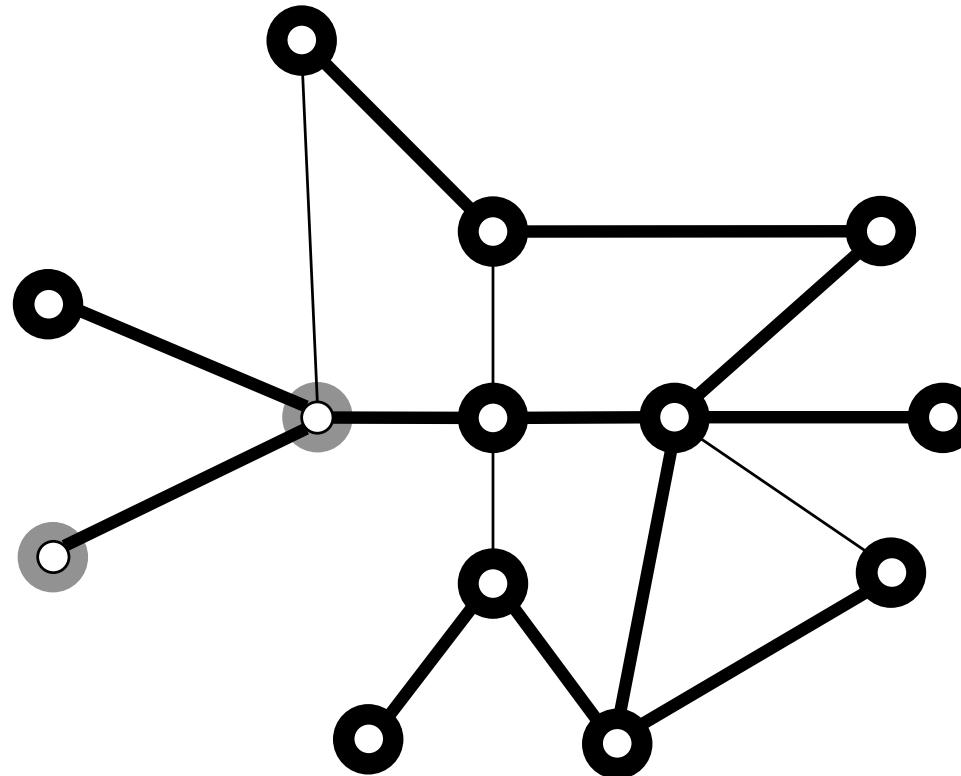
# Busca em profundidade



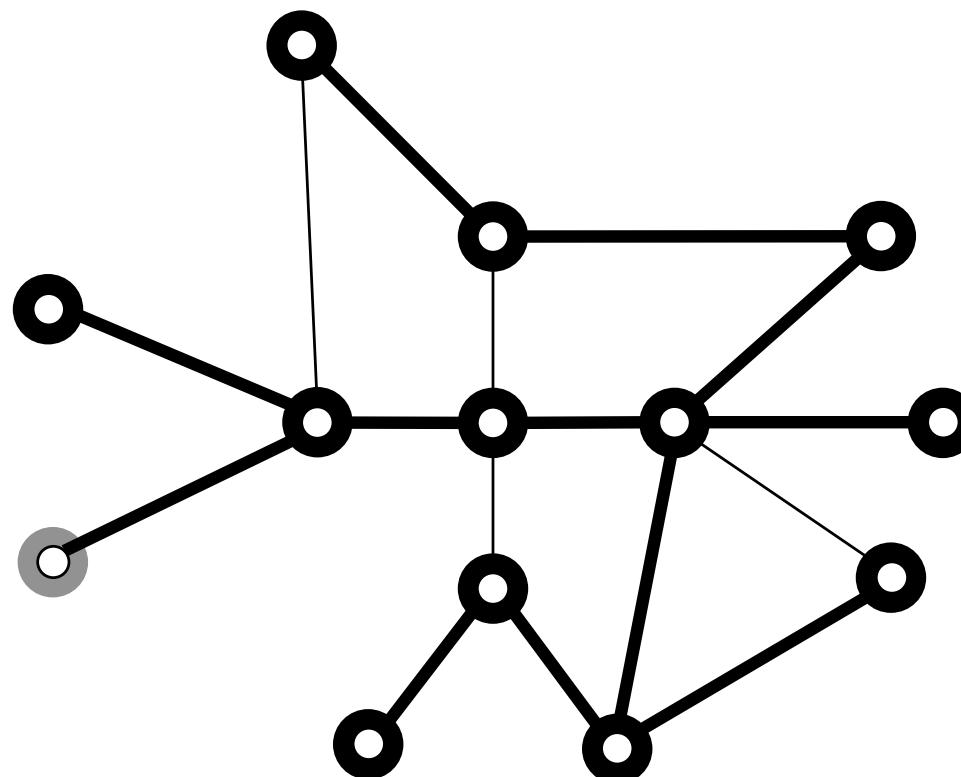
# Busca em profundidade



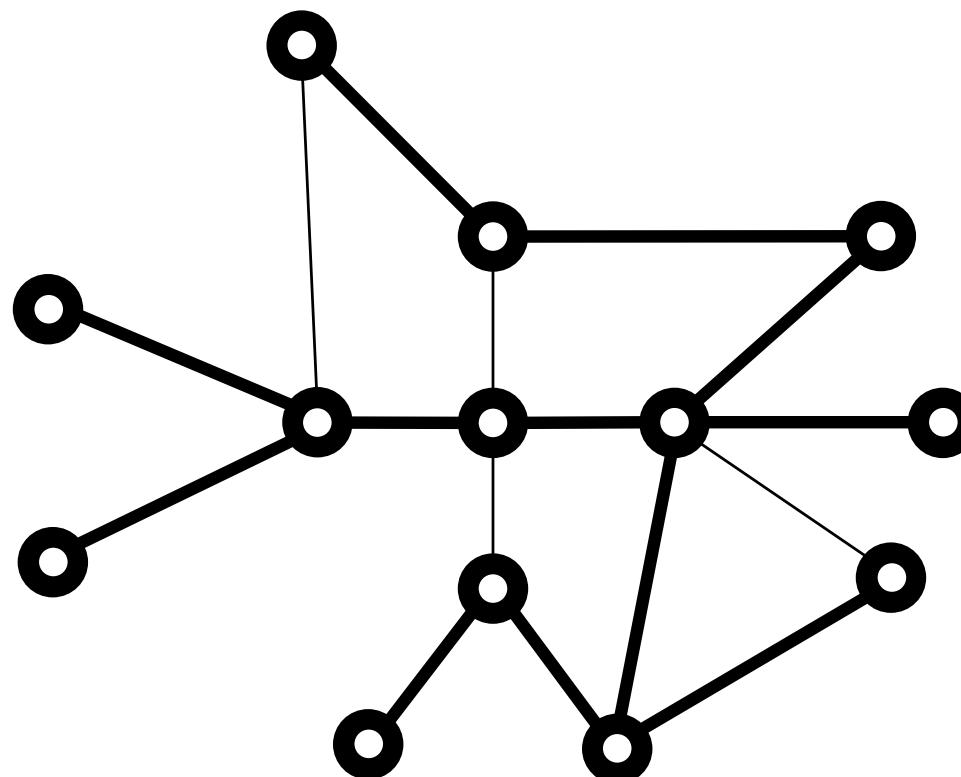
# Busca em profundidade



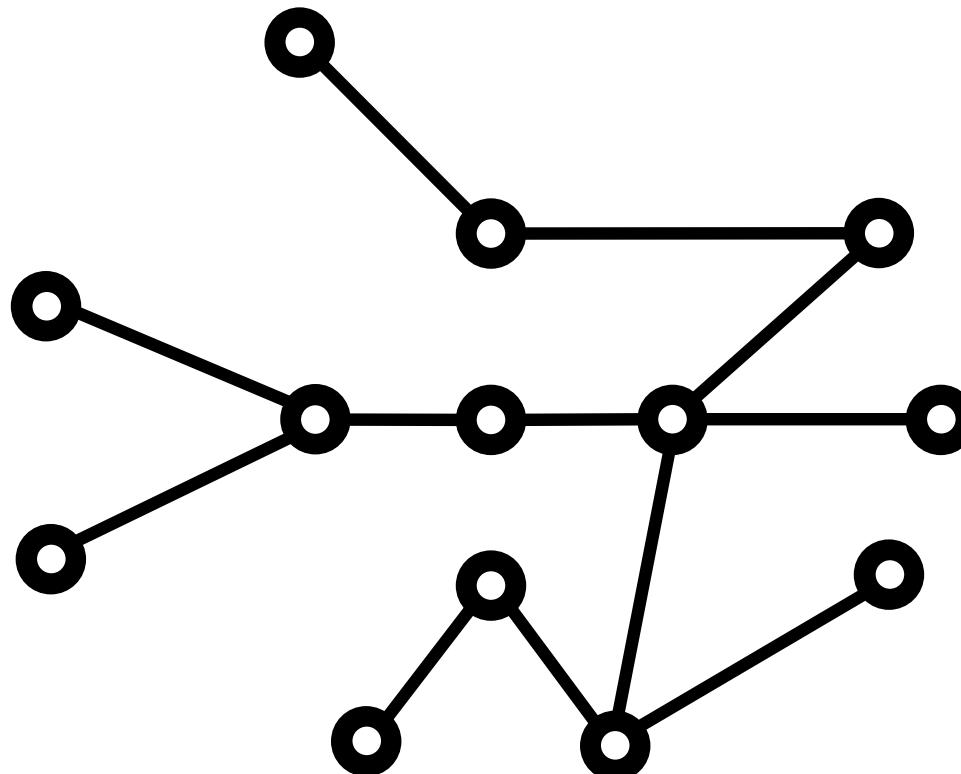
# Busca em profundidade



# Busca em profundidade

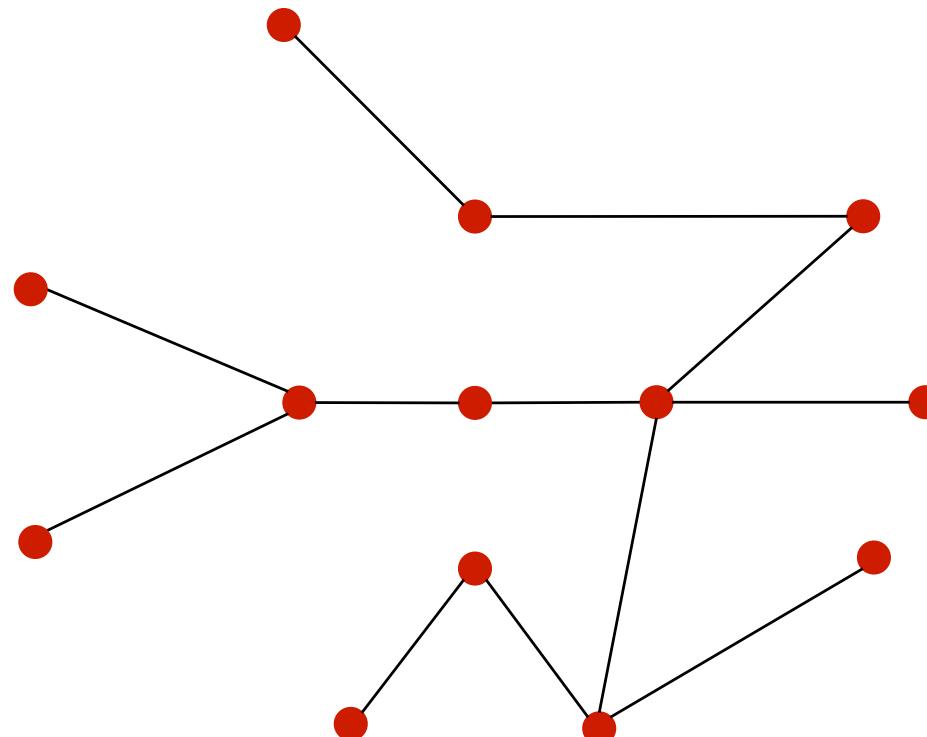


# Busca em profundidade



# Busca em profundidade

Descobre uma **árvore de busca em profundidade**.



# Busca em profundidade

## Elementos do algoritmo

Cada vértice:

1. Inicia **BRANCO**.
2. Muda para **CINZA** quando é descoberto.
3. Muda para **PRETO** quando todos seus vértices adjacentes já foram visitados.

# Busca em profundidade

## Elementos do algoritmo

A árvore é dada por:

$$G_\pi = (V, E_\pi), \text{ onde}$$

$$E_\pi = \{(v.\pi, v) | : v \in V \text{ e } v.\pi \neq \text{NULO}\} \text{ e}$$

$v.\pi$  armazena o vértice predecessor de  $v$ .

$u.d$  : guarda o “tempo” em que  $u$  é descoberto.

$u.f$  : guarda o “tempo” em que  $u$  foi “concluído”.

## ALGORITMO DFS( $G$ )

```
PARA CADA VÉRTICE  $u \in G.V$ 
     $u.COR \leftarrow \text{BRANCO}$ 
     $u.\pi \leftarrow \text{NULO}$ 
     $\text{TEMPO} \leftarrow 0$ 
    PARA CADA VÉRTICE  $u \in G.V$ 
        SE  $u.COR == \text{BRANCO}$  ENTÃO
            DFS_VISITA( $G, u$ )
```

## ALGORITMO DFS\_VISITA( $G, u$ )

```
     $\text{TEMPO} \leftarrow \text{TEMPO} + 1$                                 // VÉRTICE BRANCO FOI DESCOBERTO
     $u.D \leftarrow \text{TEMPO}$ 
     $u.COR \leftarrow \text{CINZA}$ 
    PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$  // EXPLORA ARESTA  $uv$ 
        SE  $v.COR == \text{BRANCO}$  ENTÃO
             $v.\pi \leftarrow u$ 
            DFS_VISITA( $G, v$ )
     $u.COR \leftarrow \text{PRETO}$                                 // PINTA  $u$  DE PRETO;  $u$  ESTÁ CONCLUÍDO
     $\text{TEMPO} \leftarrow \text{TEMPO} + 1$ 
     $u.F \leftarrow \text{TEMPO}$ 
```

## ALGORITMO DFS( $G$ )

PARA CADA VÉRTICE  $u \in G.V$

$u.COR \leftarrow \text{BRANCO}$

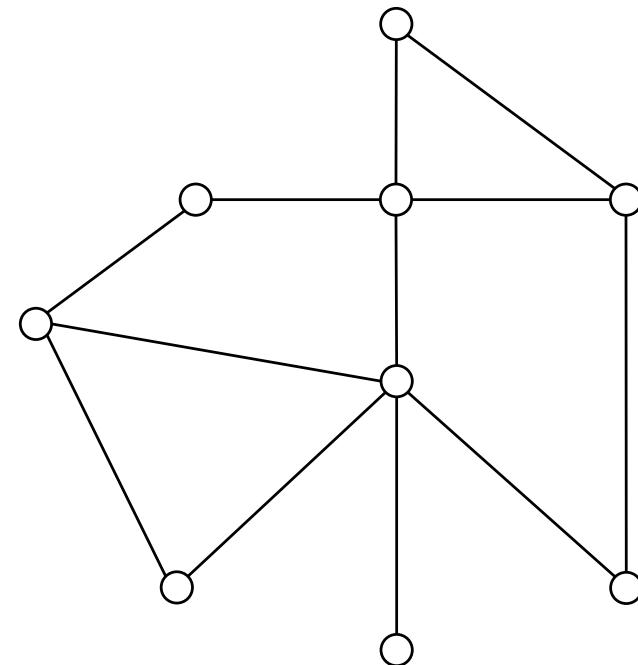
$u.\pi \leftarrow \text{NULO}$

$\text{TEMPO} \leftarrow 0$

PARA CADA VÉRTICE  $u \in G.V$

SE  $u.COR == \text{BRANCO}$  ENTÃO

DFS\_VISITA( $G, u$ )



## ALGORITMO DFS( $G$ )

PARA CADA VÉRTICE  $u \in G.V$

$u.COR \leftarrow \text{BRANCO}$

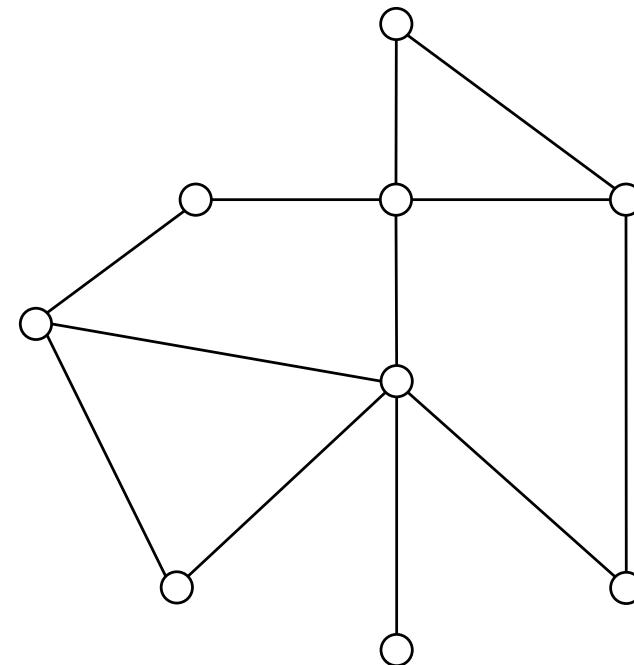
$u.\pi \leftarrow \text{NULO}$

$\text{TEMPO} \leftarrow 0$

PARA CADA VÉRTICE  $u \in G.V$

SE  $u.COR == \text{BRANCO}$  ENTÃO

DFS\_VISITA( $G, u$ )



## ALGORITMO DFS( $G$ )

PARA CADA VÉRTICE  $u \in G.V$

$u.COR \leftarrow \text{BRANCO}$

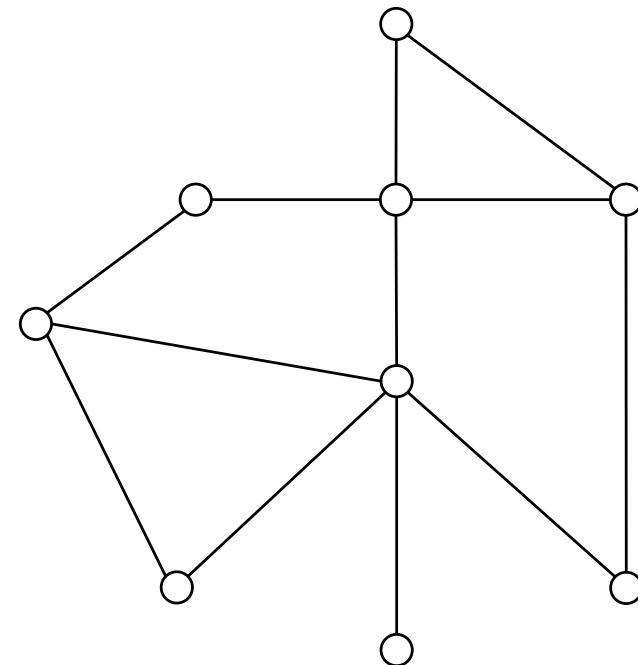
$u.\pi \leftarrow \text{NULO}$

$\text{TEMPO} \leftarrow 0$

PARA CADA VÉRTICE  $u \in G.V$

SE  $u.COR == \text{BRANCO}$  ENTÃO

DFS\_VISITA( $G, u$ )



## ALGORITMO DFS( $G$ )

PARA CADA VÉRTICE  $u \in G.V$

$u.COR \leftarrow \text{BRANCO}$

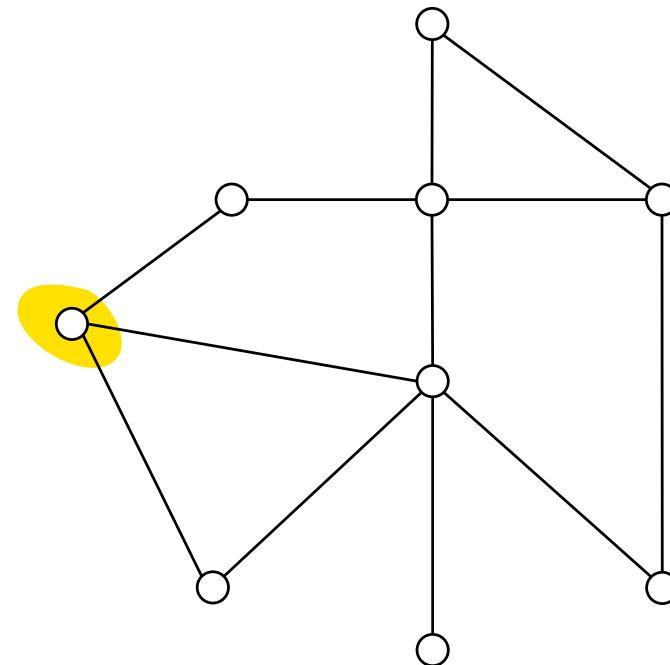
$u.\pi \leftarrow \text{NULO}$

$\text{TEMPO} \leftarrow 0$

PARA CADA VÉRTICE  $u \in G.V$

SE  $u.COR == \text{BRANCO}$  ENTÃO

DFS\_VISITA( $G, u$ )



## ALGORITMO DFS\_VISITA( $G, u$ )

$TEMPO \leftarrow TEMPO + 1$

$U.D \leftarrow TEMPO$

$U.COR \leftarrow CINZA$

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.COR == \text{BRANCO}$  ENTÃO

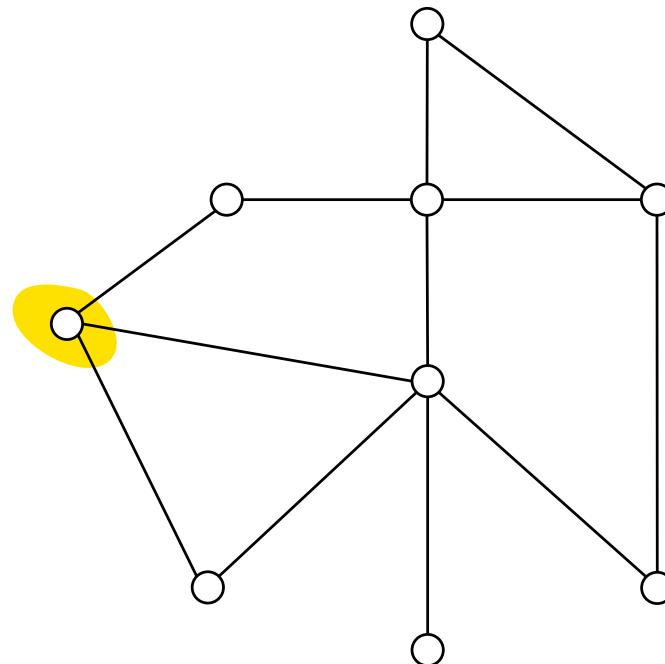
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$U.COR \leftarrow \text{PRETO}$

$TEMPO \leftarrow TEMPO + 1$

$U.F \leftarrow TEMPO$



## ALGORITMO DFS\_VISITA( $G, u$ )

$TEMPO \leftarrow TEMPO + 1$

$U.D \leftarrow TEMPO$

$U.COR \leftarrow CINZA$

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.COR == \text{BRANCO}$  ENTÃO

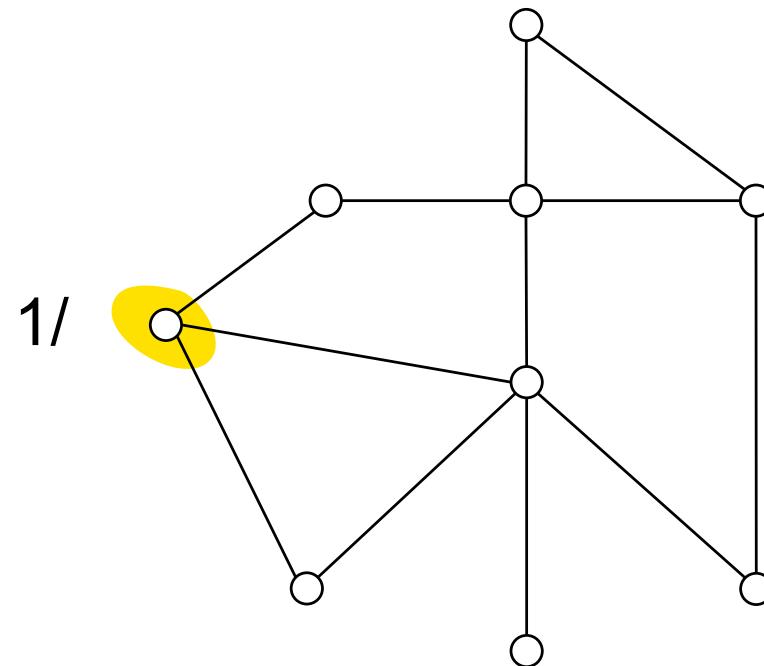
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$U.COR \leftarrow \text{PRETO}$

$TEMPO \leftarrow TEMPO + 1$

$U.F \leftarrow TEMPO$



## ALGORITMO DFS\_VISITA( $G, u$ )

$TEMPO \leftarrow TEMPO + 1$

$U.D \leftarrow TEMPO$

$U.COR \leftarrow CINZA$

PARA CADA VÉRTICE  $v \in G.ADJ[u]$

SE  $v.COR == BRANCO$  ENTÃO

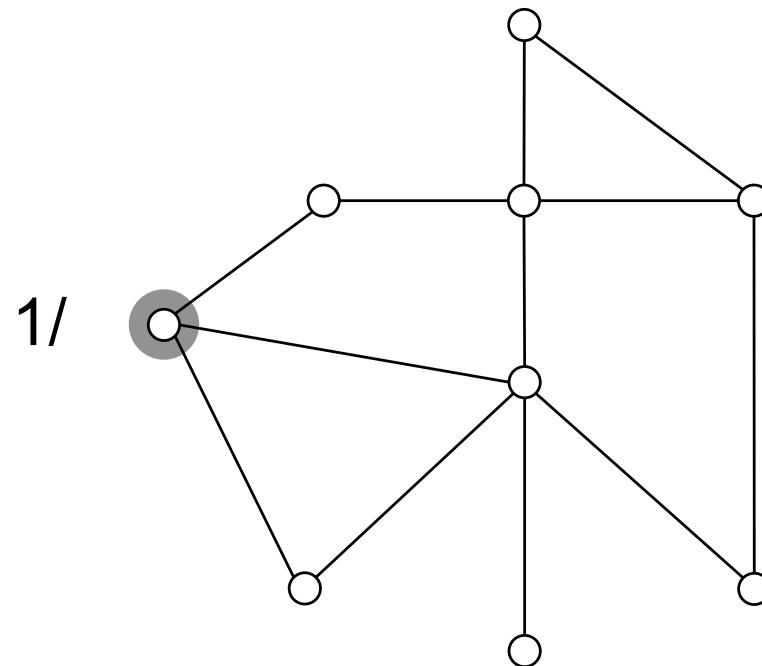
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$U.COR \leftarrow PRETO$

$TEMPO \leftarrow TEMPO + 1$

$U.F \leftarrow TEMPO$



## ALGORITMO DFS\_VISITA( $G, u$ )

$TEMPO \leftarrow TEMPO + 1$

$U.D \leftarrow TEMPO$

$U.COR \leftarrow CINZA$

**PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$**

**SE  $v.COR == BRANCO$  ENTÃO**

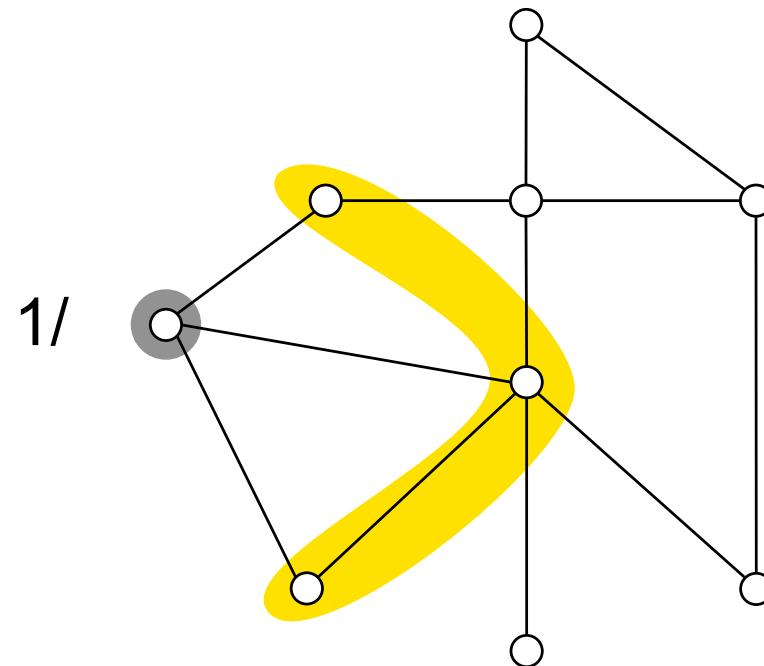
$v.\pi \leftarrow u$

$\text{DFS\_VISITA}(G, v)$

$U.COR \leftarrow PRETO$

$TEMPO \leftarrow TEMPO + 1$

$U.F \leftarrow TEMPO$



## ALGORITMO DFS\_VISITA( $G, u$ )

$TEMPO \leftarrow TEMPO + 1$

$U.D \leftarrow TEMPO$

$U.COR \leftarrow CINZA$

PARA CADA VÉRTICE  $v \in G.ADJ[u]$

SE  $v.COR == BRANCO$  ENTÃO

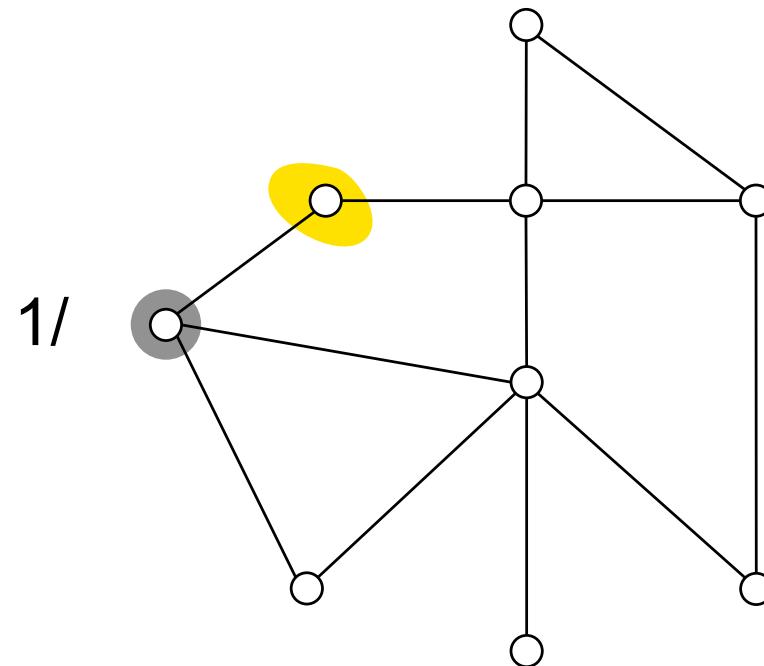
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$U.COR \leftarrow PRETO$

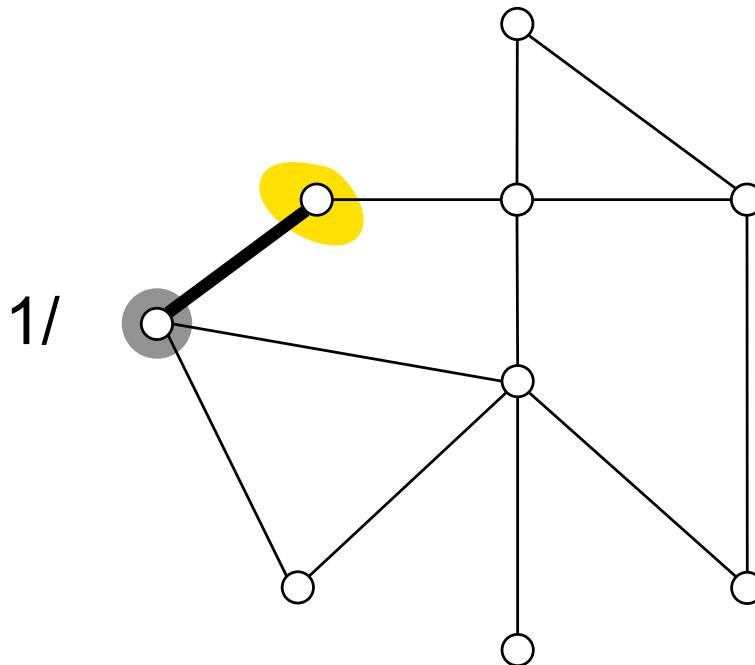
$TEMPO \leftarrow TEMPO + 1$

$U.F \leftarrow TEMPO$



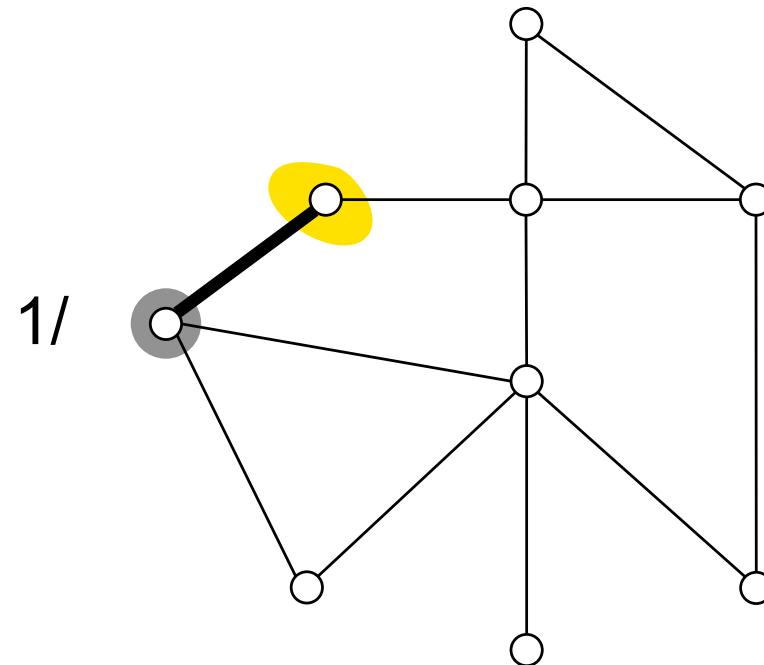
## ALGORITMO DFS\_VISITA( $G, u$ )

```
TEMPO ← TEMPO + 1  
U.D ← TEMPO  
U.COR ← CINZA  
PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$   
    SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO  
         $v.\pi \leftarrow u$   
        DFS_VISITA( $G, v$ )  
    U.COR ← PRETO  
    TEMPO ← TEMPO + 1  
    U.F ← TEMPO
```



## ALGORITMO DFS\_VISITA( $G, u$ )

```
TEMPO ← TEMPO + 1  
U.D ← TEMPO  
U.COR ← CINZA  
PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$   
    SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO  
         $v.\pi \leftarrow u$   
        DFS_VISITA( $G, v$ )  
    U.COR ← PRETO  
    TEMPO ← TEMPO + 1  
    U.F ← TEMPO
```



ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

$TEMPO \leftarrow TEMPO + 1$

$U.D \leftarrow TEMPO$

$U.COR \leftarrow CINZA$

PARA CADA VÉRTICE  $v \in G.ADJ[u]$

SE  $v.COR == BRANCO$  ENTÃO

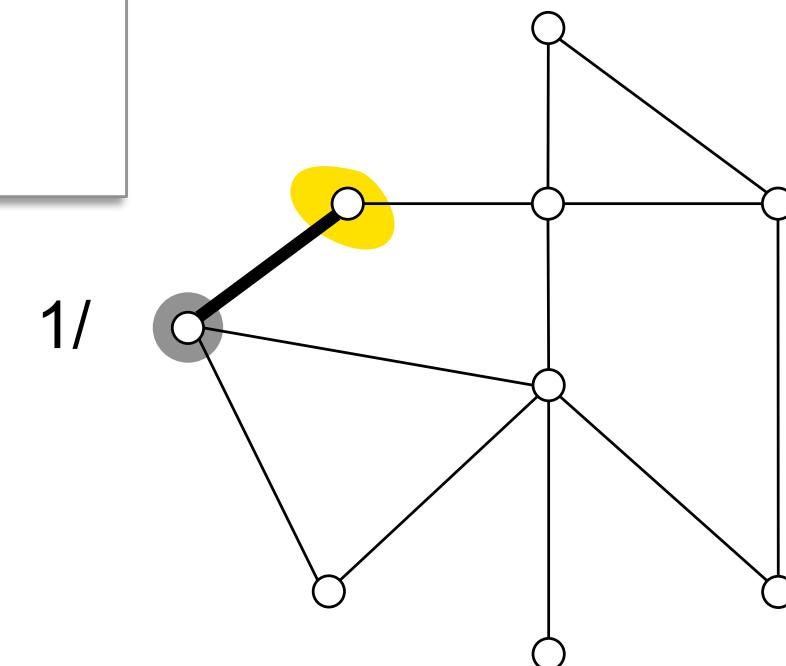
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$U.COR \leftarrow PRETO$

$TEMPO \leftarrow TEMPO + 1$

$U.F \leftarrow TEMPO$



ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

$TEMPO \leftarrow TEMPO + 1$

$U.D \leftarrow TEMPO$

$U.COR \leftarrow CINZA$

PARA CADA VÉRTICE  $v \in G.ADJ[u]$

SE  $v.COR == BRANCO$  ENTÃO

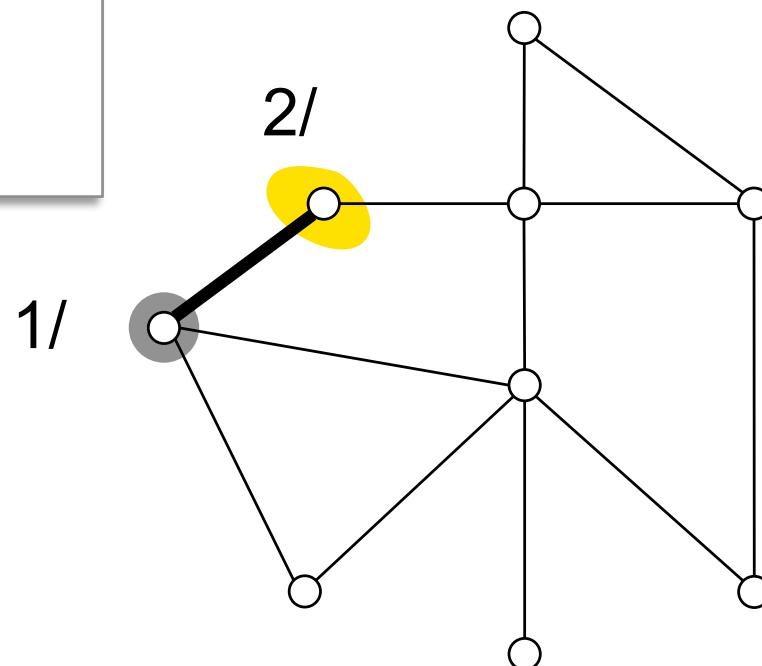
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$U.COR \leftarrow PRETO$

$TEMPO \leftarrow TEMPO + 1$

$U.F \leftarrow TEMPO$



ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

$TEMPO \leftarrow TEMPO + 1$

$U.D \leftarrow TEMPO$

$U.COR \leftarrow CINZA$

PARA CADA VÉRTICE  $v \in G.ADJ[u]$

SE  $v.COR == BRANCO$  ENTÃO

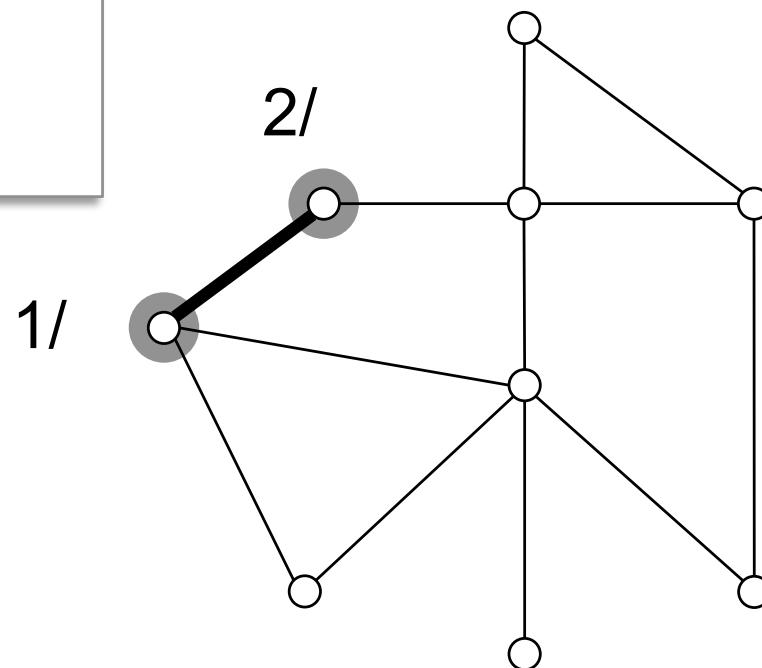
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$U.COR \leftarrow PRETO$

$TEMPO \leftarrow TEMPO + 1$

$U.F \leftarrow TEMPO$



ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

$TEMPO \leftarrow TEMPO + 1$

$U.D \leftarrow TEMPO$

$U.COR \leftarrow CINZA$

PARA CADA VÉRTICE  $v \in G.ADJ[u]$

SE  $v.COR == BRANCO$  ENTÃO

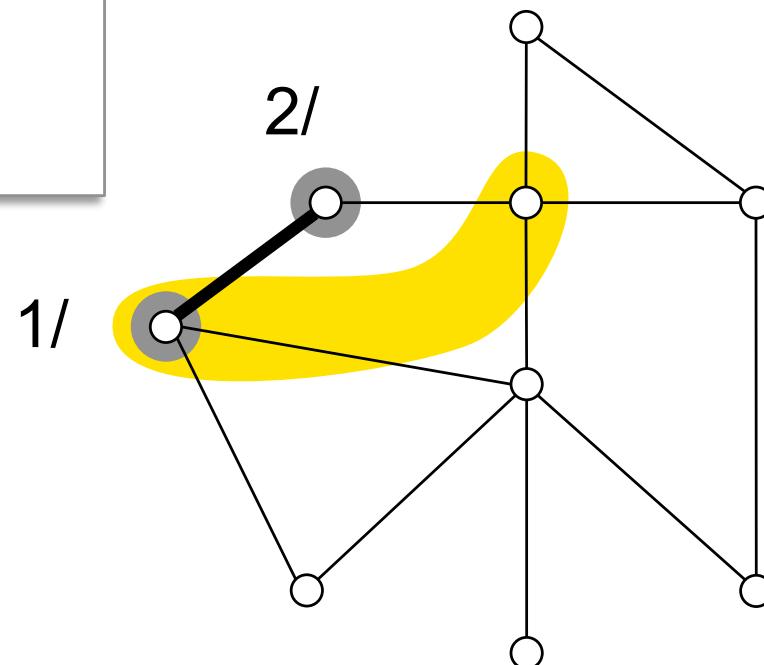
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$U.COR \leftarrow PRETO$

$TEMPO \leftarrow TEMPO + 1$

$U.F \leftarrow TEMPO$



ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

$TEMPO \leftarrow TEMPO + 1$

$U.D \leftarrow TEMPO$

$U.COR \leftarrow CINZA$

PARA CADA VÉRTICE  $v \in G.ADJ[u]$

SE  $v.COR == BRANCO$  ENTÃO

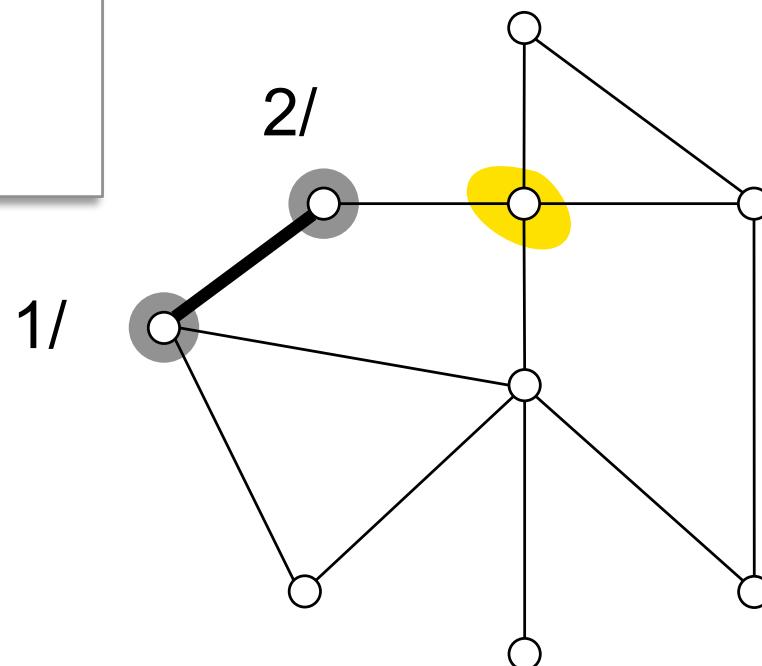
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$U.COR \leftarrow PRETO$

$TEMPO \leftarrow TEMPO + 1$

$U.F \leftarrow TEMPO$



ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

$TEMPO \leftarrow TEMPO + 1$

$U.D \leftarrow TEMPO$

$U.COR \leftarrow CINZA$

PARA CADA VÉRTICE  $v \in G.ADJ[u]$

SE  $v.COR == BRANCO$  ENTÃO

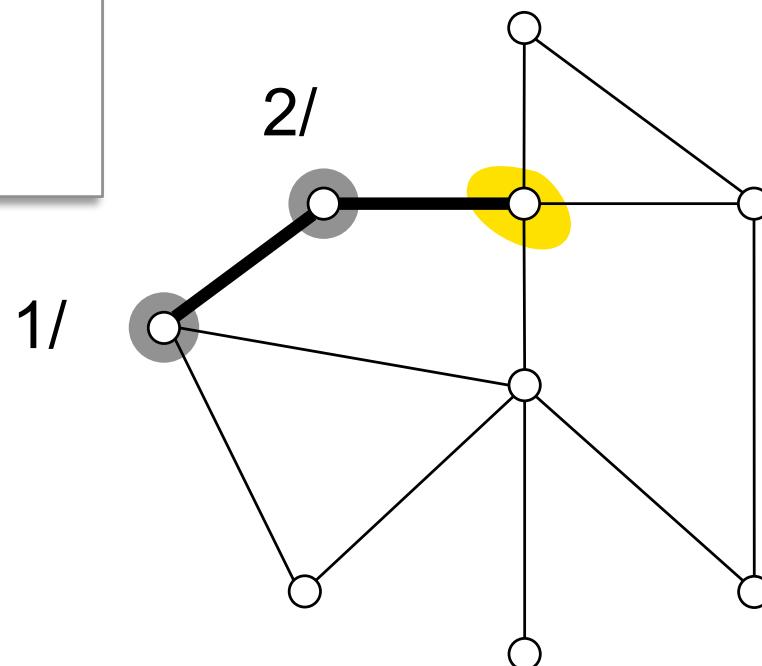
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$U.COR \leftarrow PRETO$

$TEMPO \leftarrow TEMPO + 1$

$U.F \leftarrow TEMPO$



**ALGORITMO DFS\_VISITA( $G, u$ )**

## ALGORITMO DFS\_VISITA( $G, u$ )

$\text{TEMPO} \leftarrow \text{TEMPO} + 1$

$U.D \leftarrow TEMPO$

*U.COR* ← CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE V.COR == BRANCO ENTÃO

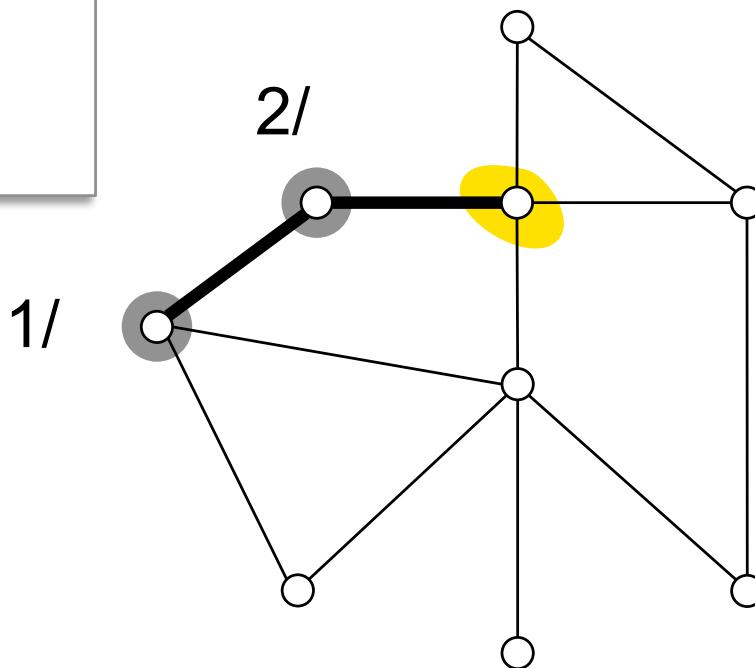
$V.\pi \leftarrow U$

DFS\_VISITA( $G, v$ )

*U.COR* ← PRETO

$TEMPO \leftarrow TEMPO + 1$

*U.F ← TEMPO*



**ALGORITMO DFS\_VISITA( $G, u$ )**

## ALGORITMO DFS VISITA( $G, u$ )

## ALGORITMO DFS\_VISITA( $G, u$ )

**TEMPO**  $\leftarrow$  **TEMPO** + 1

$U.D \leftarrow TEMPO$

*U.COR* ← CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

**SE V.COR == BRANCO ENTÃO**

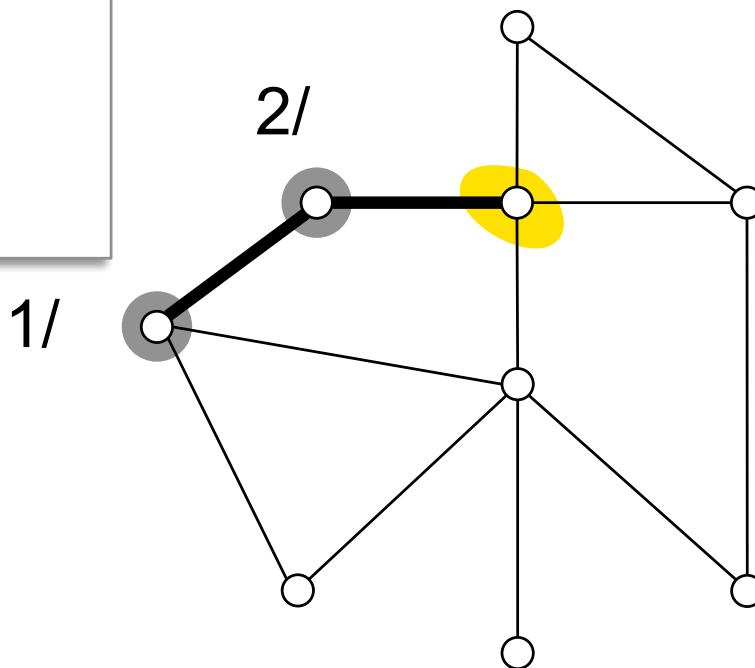
$$V.\pi \leftarrow U$$

DFS\_VISITA( $G, v$ )

*U.COR* ← PRETO

$\text{TEMPO} \leftarrow \text{TEMPO} + 1$

U.F  $\leftarrow$  TEMPO



## ALGORITMO DFS\_VISITA( $G, u$ )

## ALGORITMO DFS VISITA( $G, u$ )

## ALGORITMO DFS\_VISITA( $G, u$ )

$\text{TEMPO} \leftarrow \text{TEMPO} + 1$

$U.D \leftarrow TEMPO$

*U.COR* ← CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE V.COR == BRANCO ENTÃO

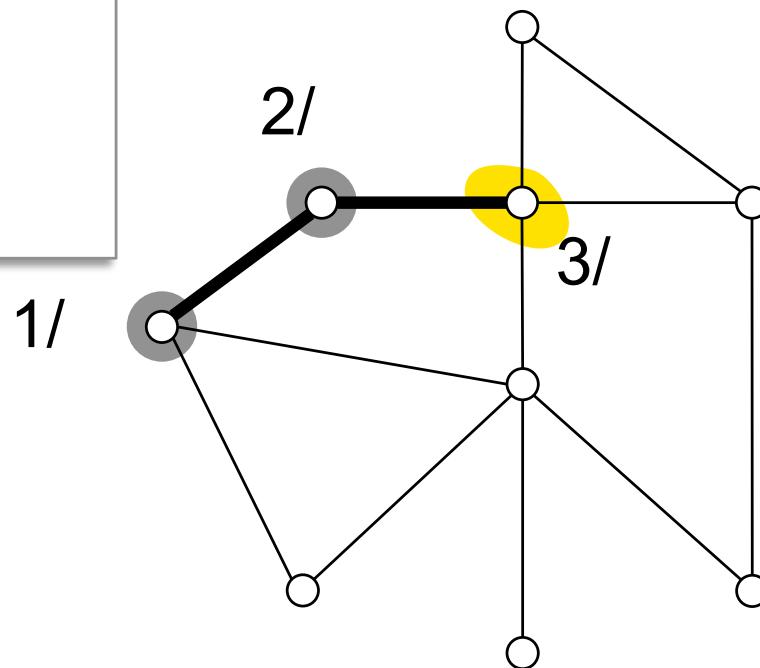
$V.\pi \leftarrow U$

DFS\_VISITA( $G, v$ )

*U.COR* ← PRETO

$\text{TEMPO} \leftarrow \text{TEMPO} + 1$

U.F ← TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

$U.D \leftarrow$  TEMPO

$U.COR \leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.COR ==$  BRANCO ENTÃO

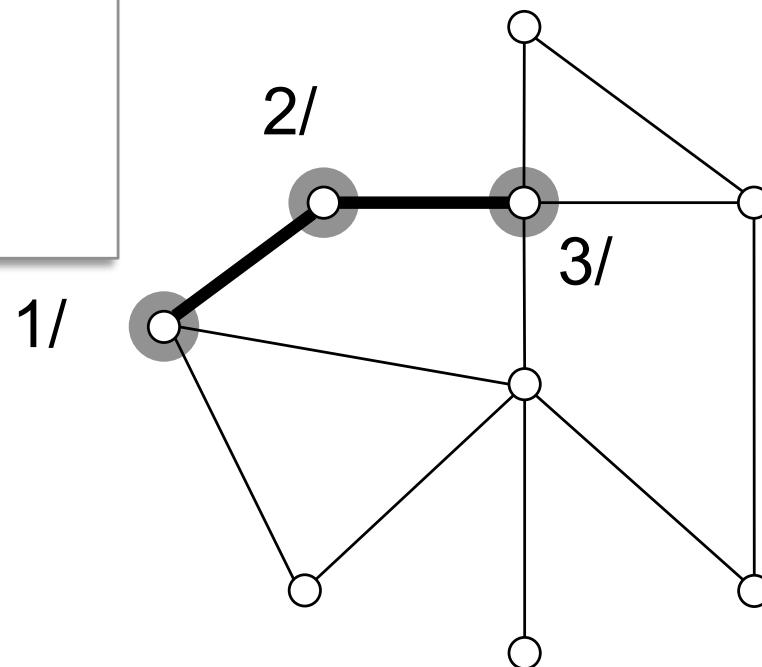
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$U.COR \leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

$U.F \leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

$U.D \leftarrow$  TEMPO

$U.COR \leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.COR ==$  BRANCO ENTÃO

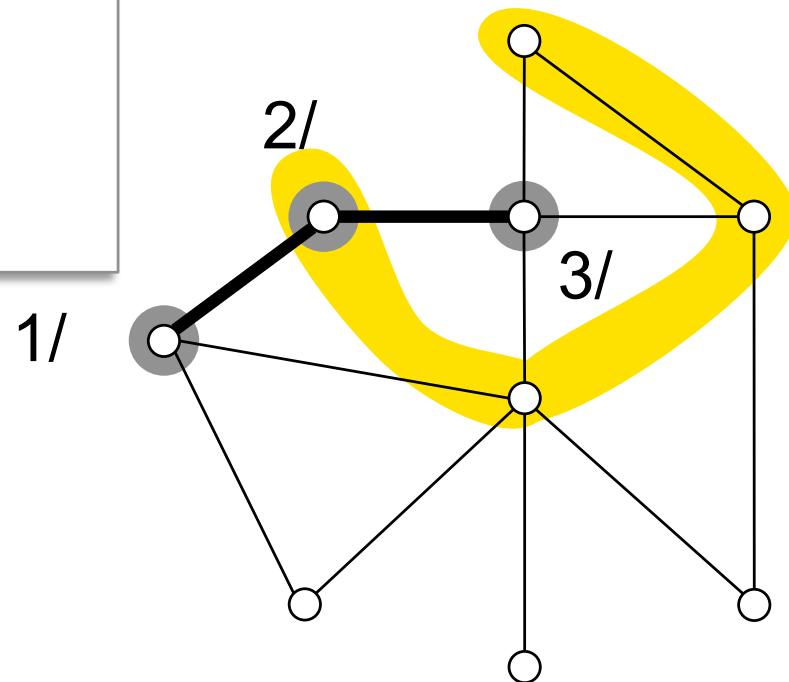
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$U.COR \leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

$U.F \leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

$u.D \leftarrow$  TEMPO

$u.COR \leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.COR ==$  BRANCO ENTÃO

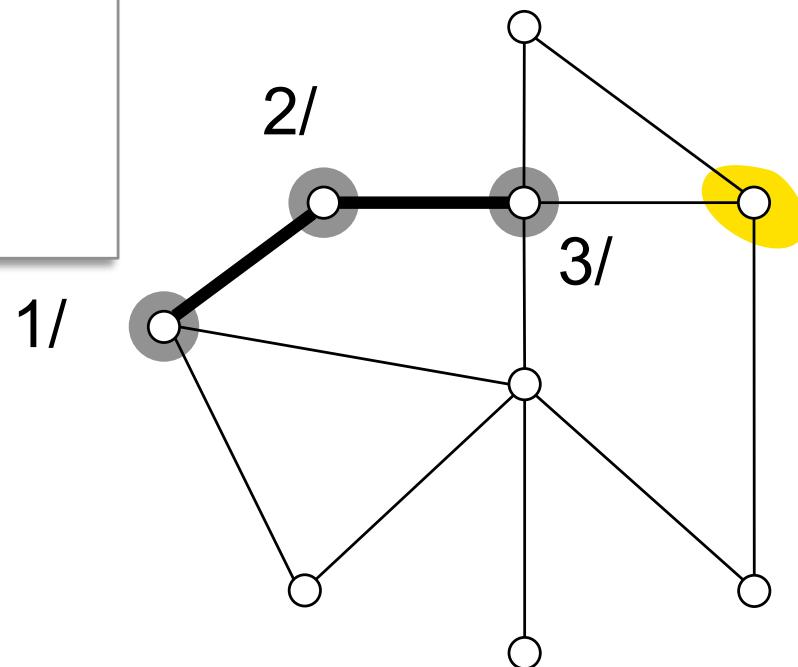
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$u.COR \leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

$u.F \leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

$u.D \leftarrow$  TEMPO

$u.COR \leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.COR ==$  BRANCO ENTÃO

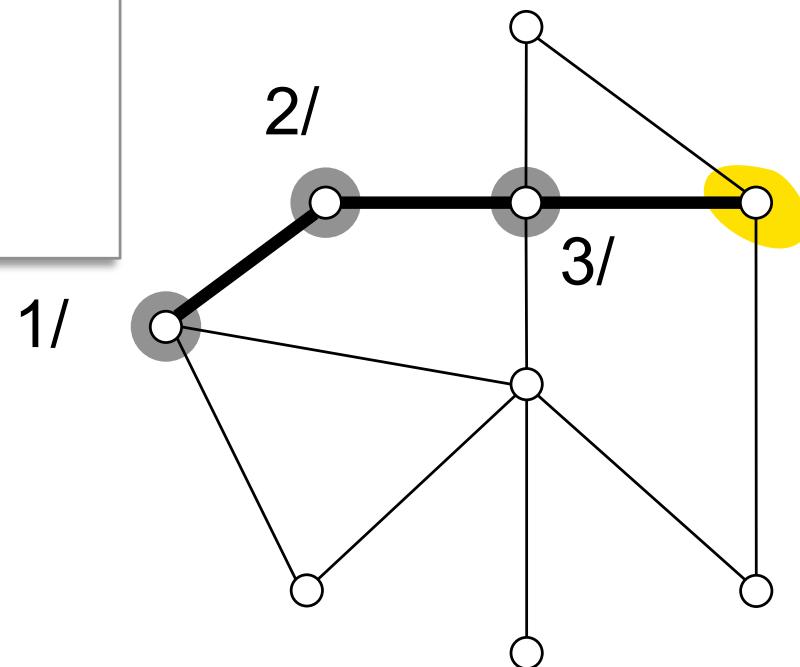
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$u.COR \leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

$u.F \leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

$U.D \leftarrow$  TEMPO

$U.COR \leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.COR ==$  BRANCO ENTÃO

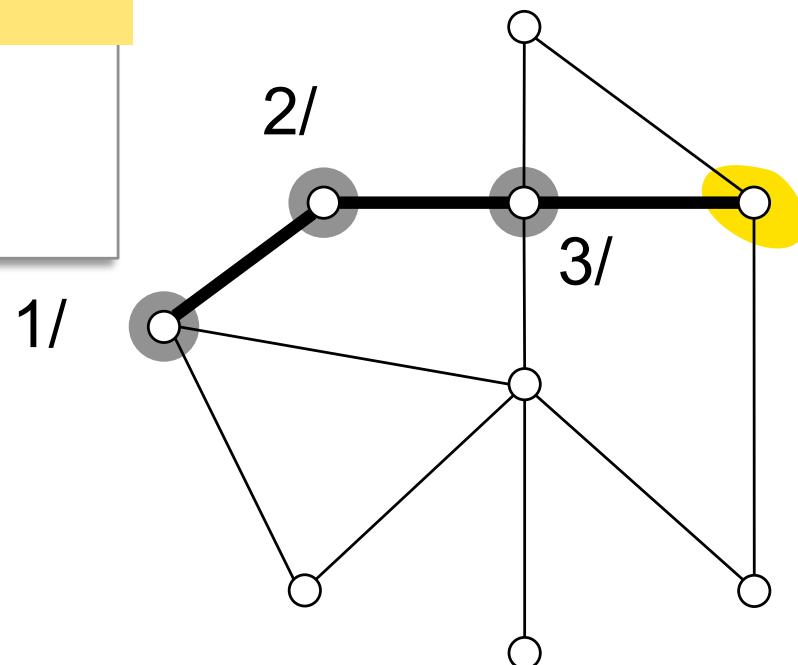
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$U.COR \leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

$U.F \leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

$U.D \leftarrow$  TEMPO

$U.COR \leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.ADJ[u]$

SE  $v.COR ==$  BRANCO ENTÃO

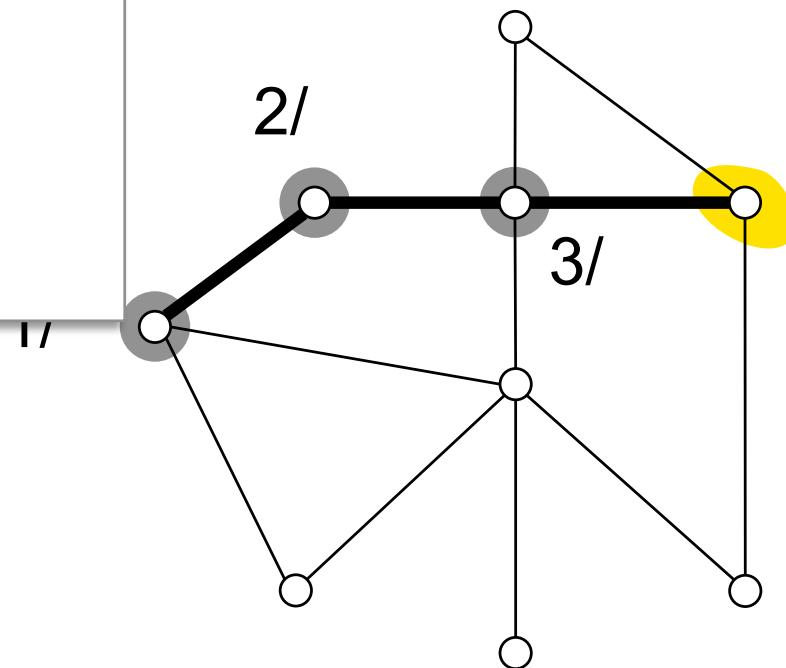
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$U.COR \leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

$U.F \leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

  ALGORITMO DFS\_VISITA( $G, u$ )

    ALGORITMO DFS\_VISITA( $G, u$ )

      ALGORITMO DFS\_VISITA( $G, u$ )

       TEMPO  $\leftarrow$  TEMPO + 1

$U.D \leftarrow$  TEMPO

$U.COR \leftarrow$  CINZA

       PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

          SE  $v.COR ==$  BRANCO ENTÃO

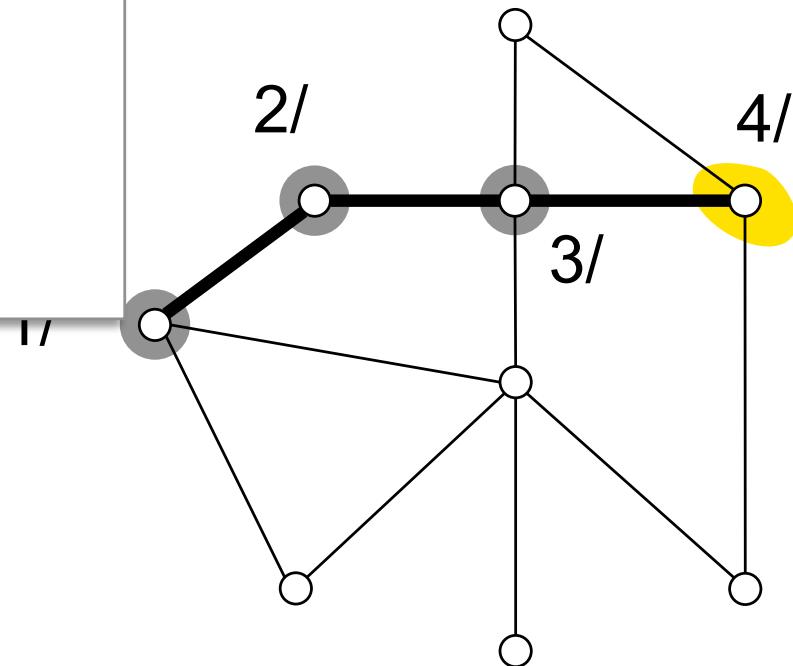
$V.\pi \leftarrow u$

            DFS\_VISITA( $G, v$ )

$U.COR \leftarrow$  PRETO

       TEMPO  $\leftarrow$  TEMPO + 1

$U.F \leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

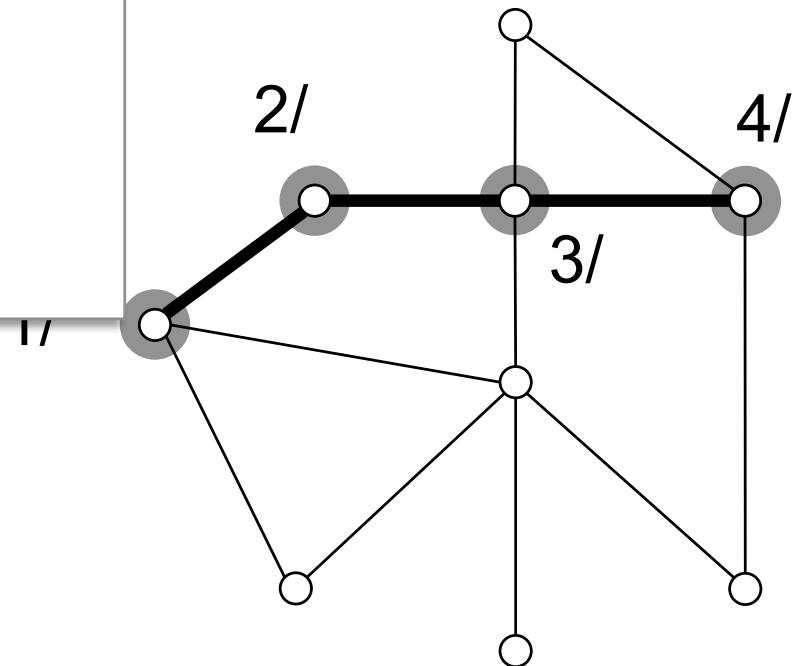
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

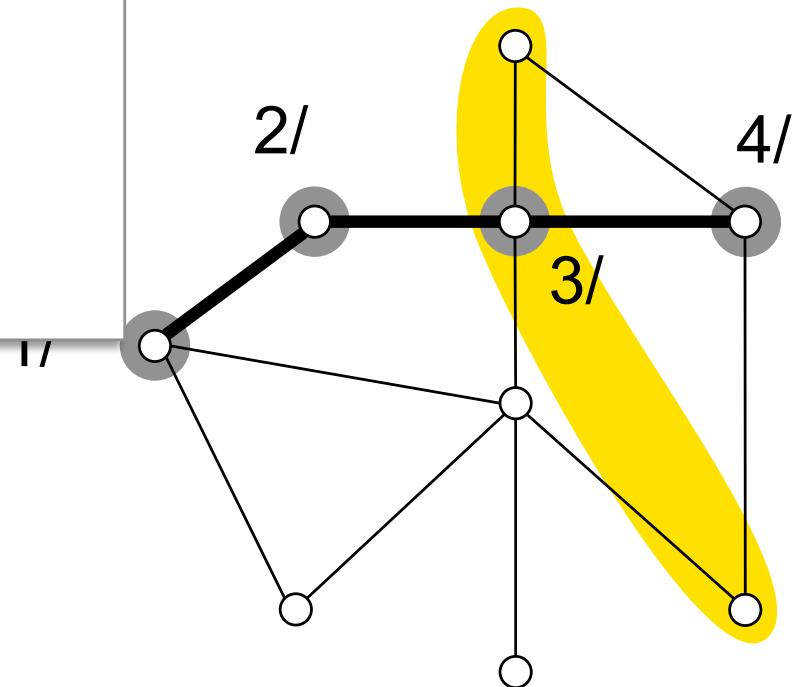
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

  ALGORITMO DFS\_VISITA( $G, u$ )

    ALGORITMO DFS\_VISITA( $G, u$ )

      ALGORITMO DFS\_VISITA( $G, u$ )

       TEMPO  $\leftarrow$  TEMPO + 1

.D  $\leftarrow$  TEMPO

.COR  $\leftarrow$  CINZA

       PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

          SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

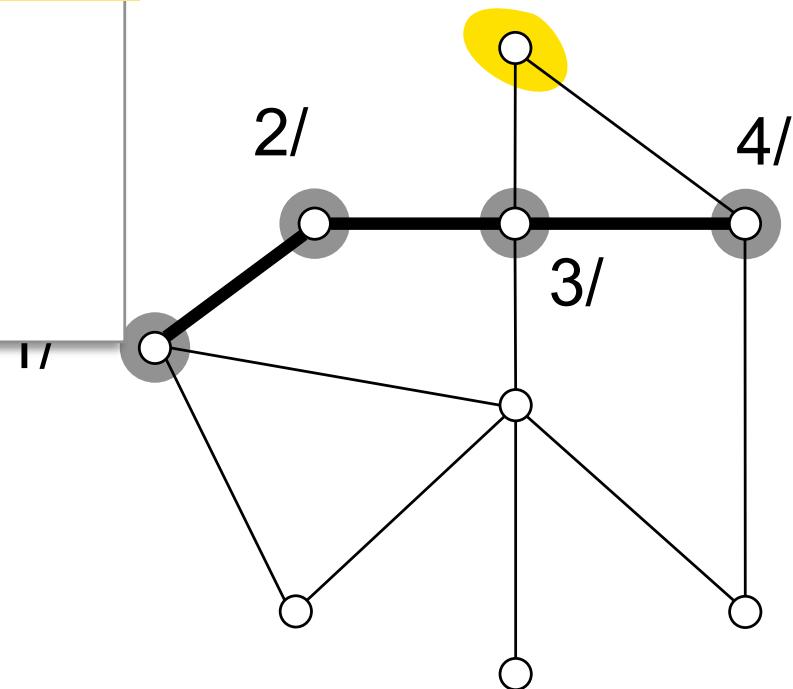
$v.\pi \leftarrow u$

            DFS\_VISITA( $G, v$ )

.COR  $\leftarrow$  PRETO

          TEMPO  $\leftarrow$  TEMPO + 1

.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

  ALGORITMO DFS\_VISITA( $G, u$ )

    ALGORITMO DFS\_VISITA( $G, u$ )

      ALGORITMO DFS\_VISITA( $G, u$ )

       TEMPO  $\leftarrow$  TEMPO + 1

.D  $\leftarrow$  TEMPO

.COR  $\leftarrow$  CINZA

       PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

          SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

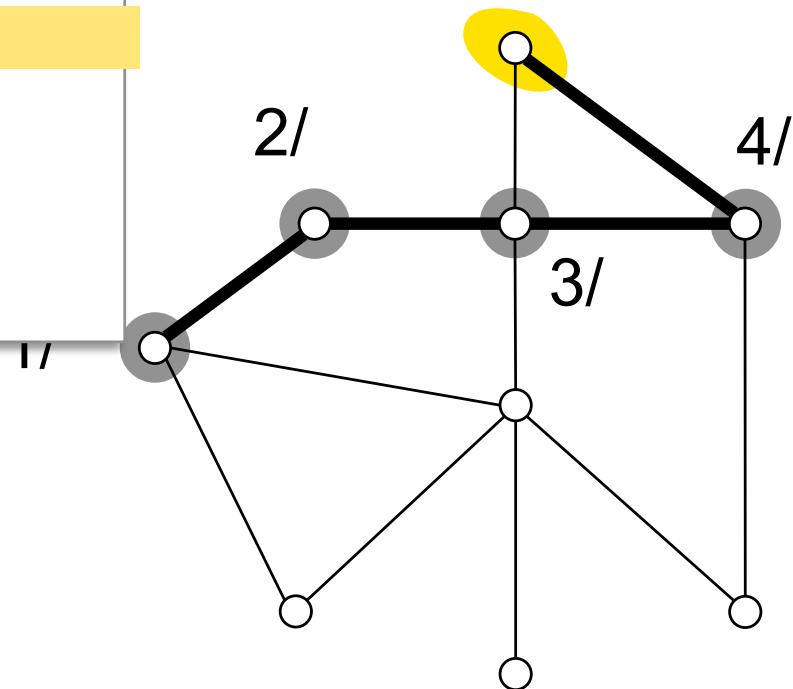
$v.\pi \leftarrow u$

            DFS\_VISITA( $G, v$ )

.COR  $\leftarrow$  PRETO

       TEMPO  $\leftarrow$  TEMPO + 1

.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

  ALGORITMO DFS\_VISITA( $G, u$ )

    ALGORITMO DFS\_VISITA( $G, u$ )

      ALGORITMO DFS\_VISITA( $G, u$ )

       TEMPO  $\leftarrow$  TEMPO + 1

.D  $\leftarrow$  TEMPO

.COR  $\leftarrow$  CINZA

       PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

          SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

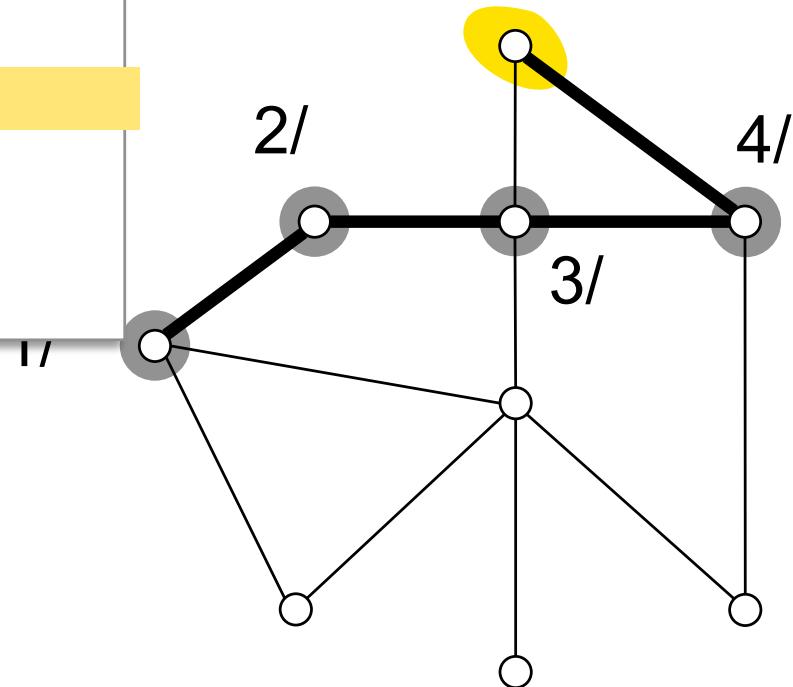
$v.\pi \leftarrow u$

           DFS\_VISITA( $G, v$ )

.COR  $\leftarrow$  PRETO

       TEMPO  $\leftarrow$  TEMPO + 1

.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

  ALGORITMO DFS\_VISITA( $G, u$ )

    ALGORITMO DFS\_VISITA( $G, u$ )

      ALGORITMO DFS\_VISITA( $G, u$ )

      ALGORITMO DFS\_VISITA( $G, u$ )

        TEMPO  $\leftarrow$  TEMPO + 1

.D  $\leftarrow$  TEMPO

.COR  $\leftarrow$  CINZA

        PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

          SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

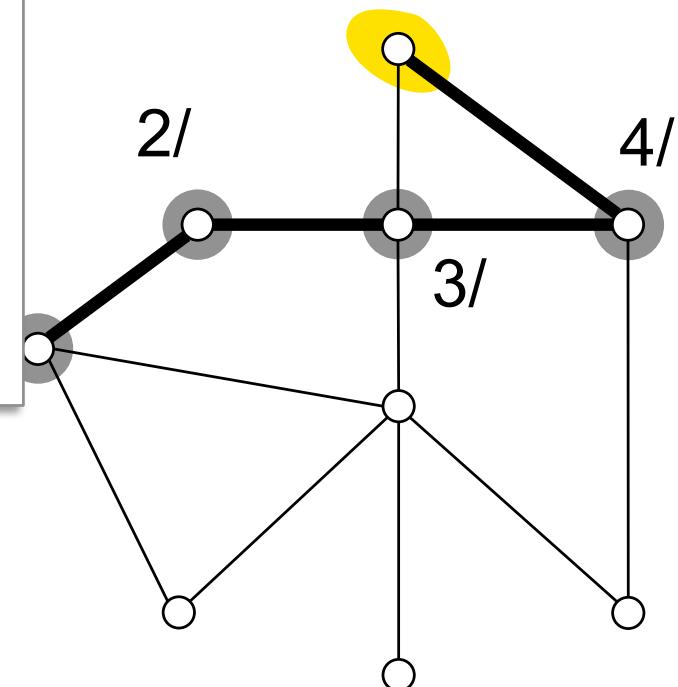
$v.\pi \leftarrow u$

            DFS\_VISITA( $G, v$ )

.COR  $\leftarrow$  PRETO

        TEMPO  $\leftarrow$  TEMPO + 1

.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

  ALGORITMO DFS\_VISITA( $G, u$ )

    ALGORITMO DFS\_VISITA( $G, u$ )

      ALGORITMO DFS\_VISITA( $G, u$ )

      ALGORITMO DFS\_VISITA( $G, u$ )

        TEMPO  $\leftarrow$  TEMPO + 1

$u.D \leftarrow \text{TEMPO}$

$u.COR \leftarrow \text{CINZA}$

        PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

          SE  $v.COR == \text{BRANCO}$  ENTÃO

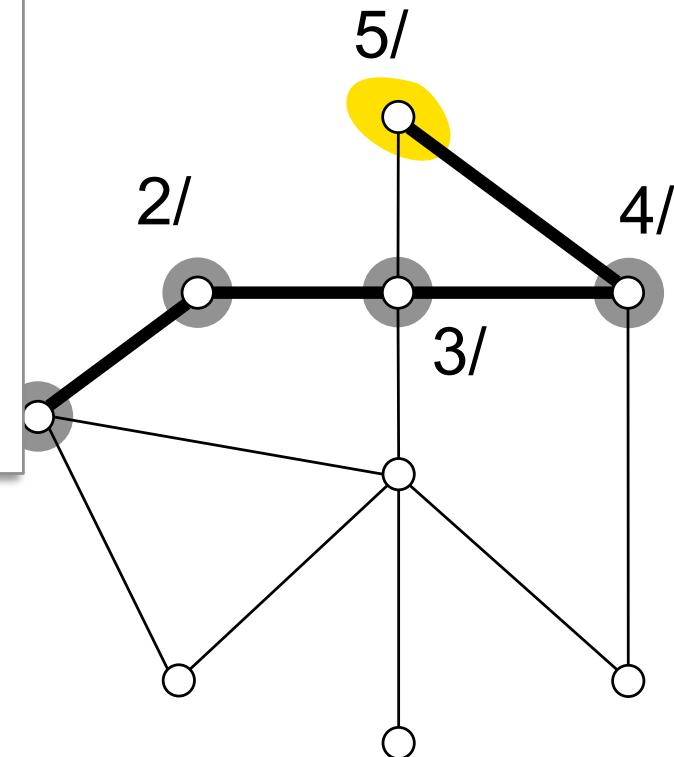
$v.\pi \leftarrow u$

            DFS\_VISITA( $G, v$ )

$u.COR \leftarrow \text{PRETO}$

        TEMPO  $\leftarrow$  TEMPO + 1

$u.F \leftarrow \text{TEMPO}$



ALGORITMO DFS\_VISITA( $G, u$ )

  ALGORITMO DFS\_VISITA( $G, u$ )

    ALGORITMO DFS\_VISITA( $G, u$ )

      ALGORITMO DFS\_VISITA( $G, u$ )

      ALGORITMO DFS\_VISITA( $G, u$ )

        TEMPO  $\leftarrow$  TEMPO + 1

$u.D \leftarrow$  TEMPO

$u.COR \leftarrow$  CINZA

        PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

          SE  $v.COR ==$  BRANCO ENTÃO

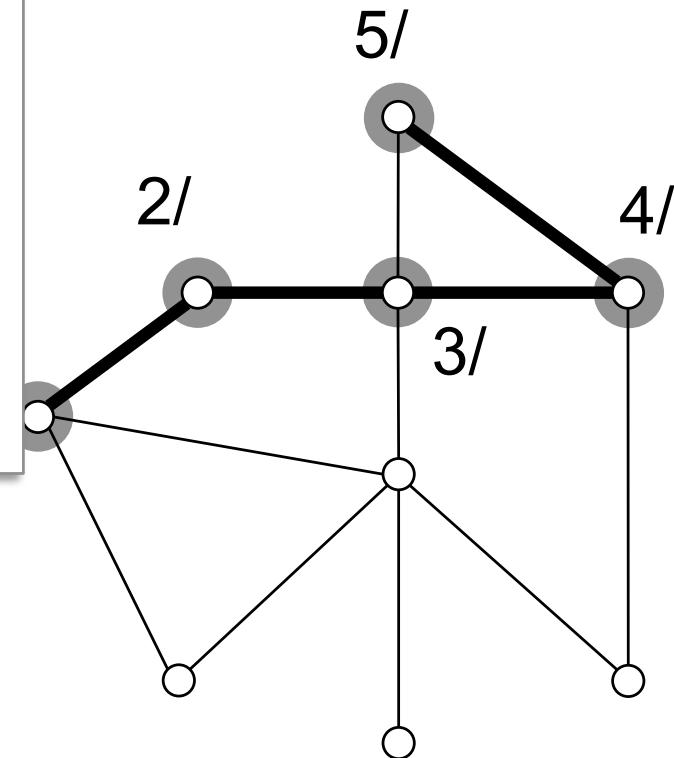
$v.\pi \leftarrow u$

            DFS\_VISITA( $G, v$ )

$u.COR \leftarrow$  PRETO

        TEMPO  $\leftarrow$  TEMPO + 1

$u.F \leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

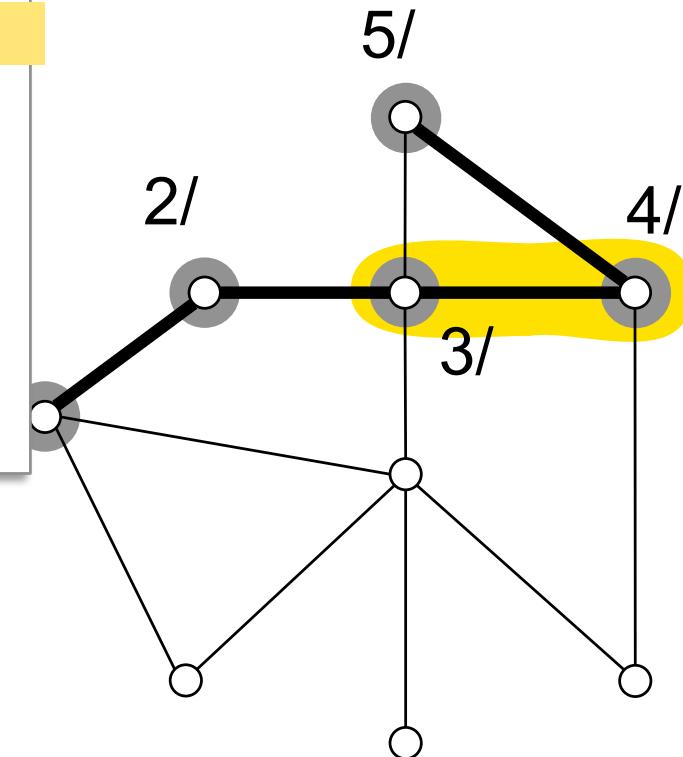
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

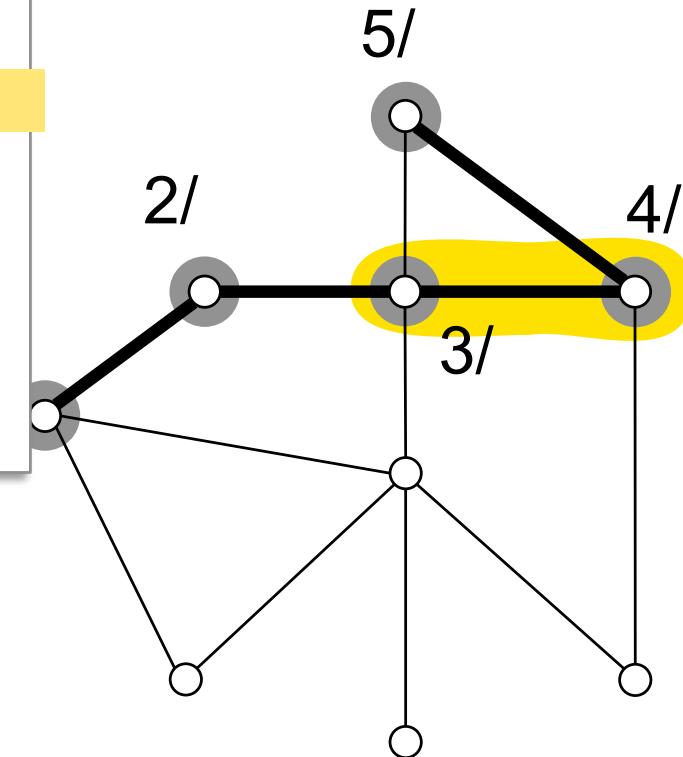
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

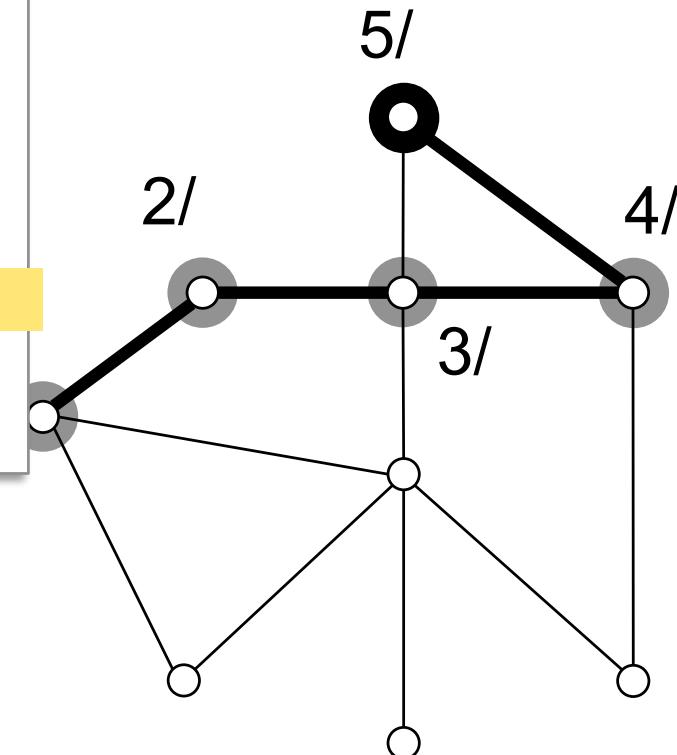
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

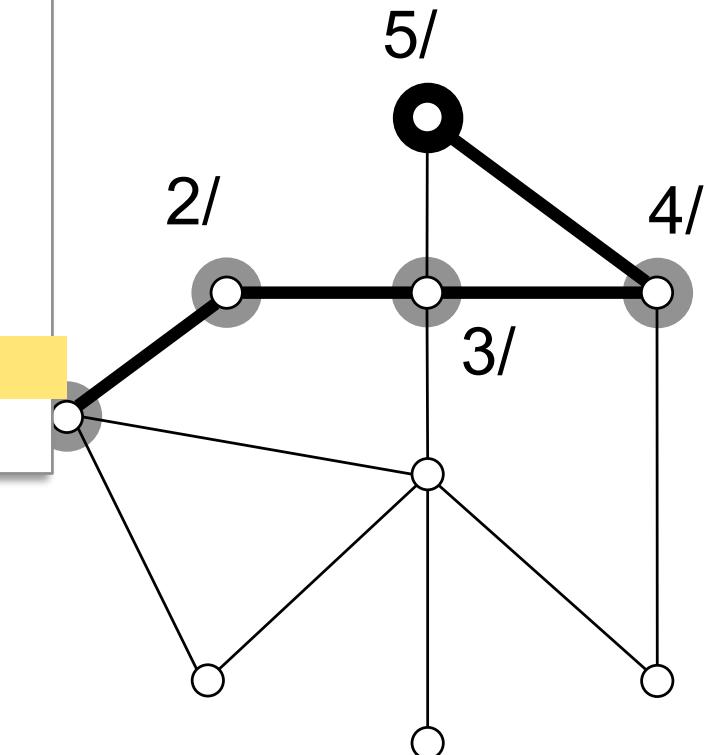
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

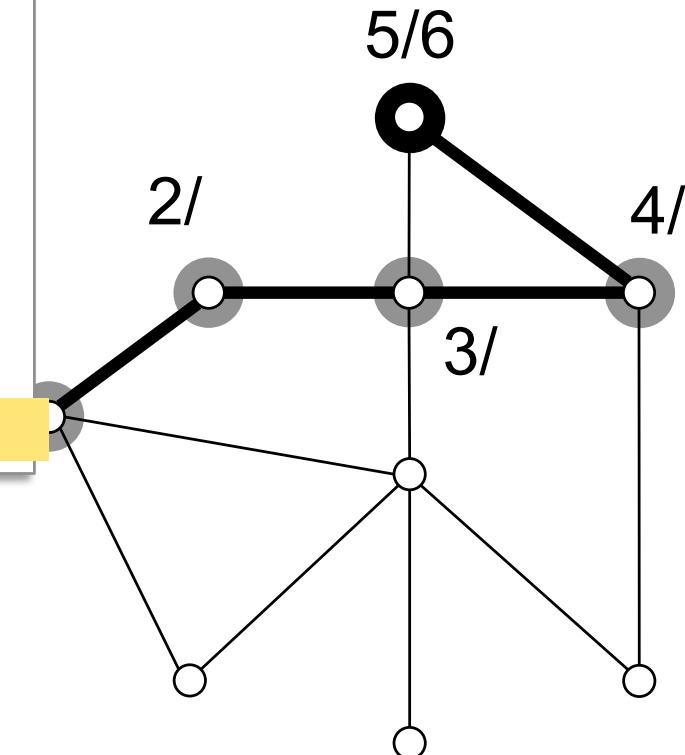
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

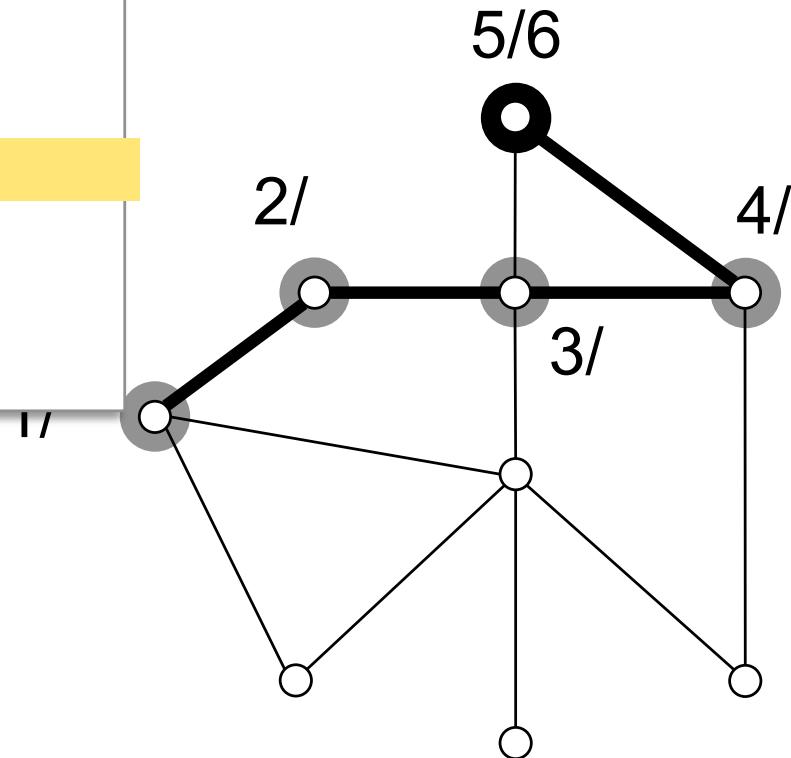
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

  ALGORITMO DFS\_VISITA( $G, u$ )

    ALGORITMO DFS\_VISITA( $G, u$ )

      ALGORITMO DFS\_VISITA( $G, u$ )

       TEMPO  $\leftarrow$  TEMPO + 1

.D  $\leftarrow$  TEMPO

.COR  $\leftarrow$  CINZA

       PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

          SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

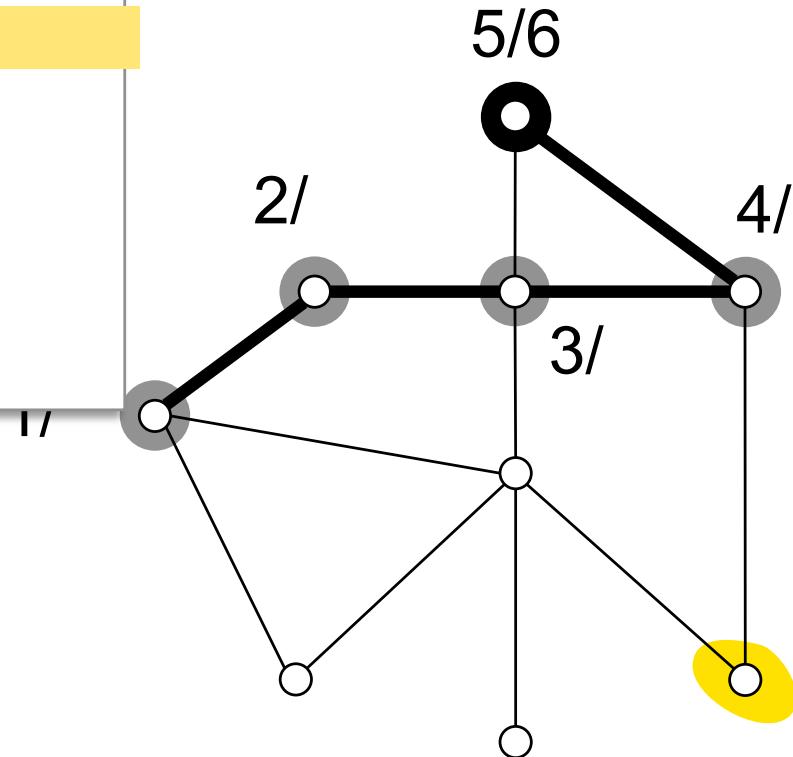
$v.\pi \leftarrow u$

           DFS\_VISITA( $G, v$ )

.COR  $\leftarrow$  PRETO

       TEMPO  $\leftarrow$  TEMPO + 1

.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

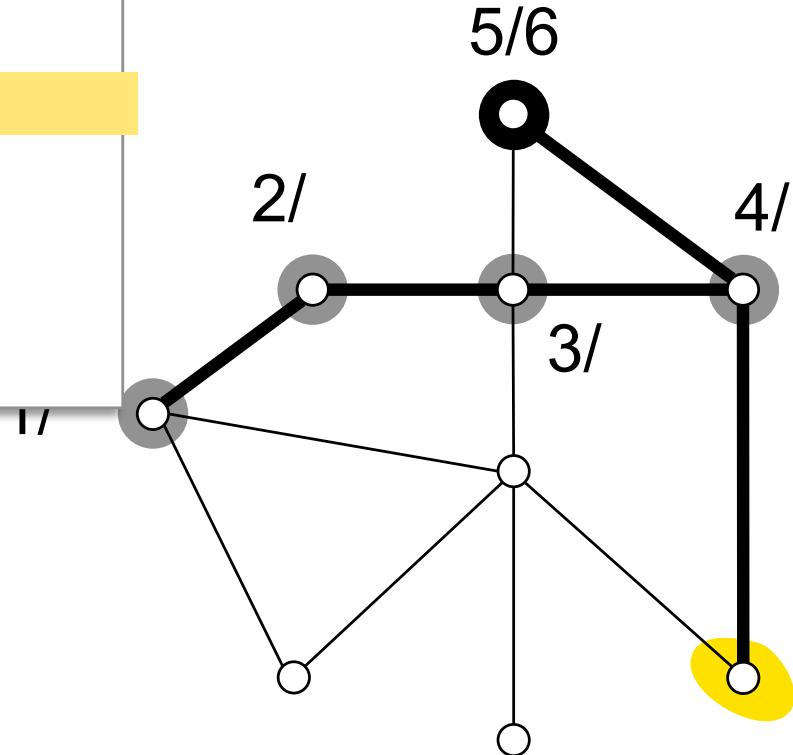
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

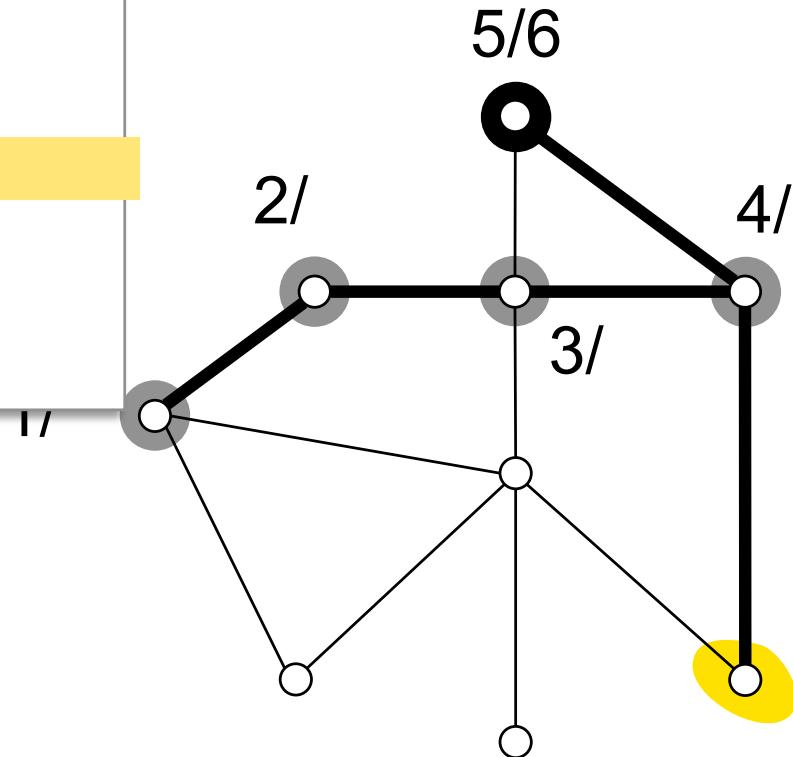
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

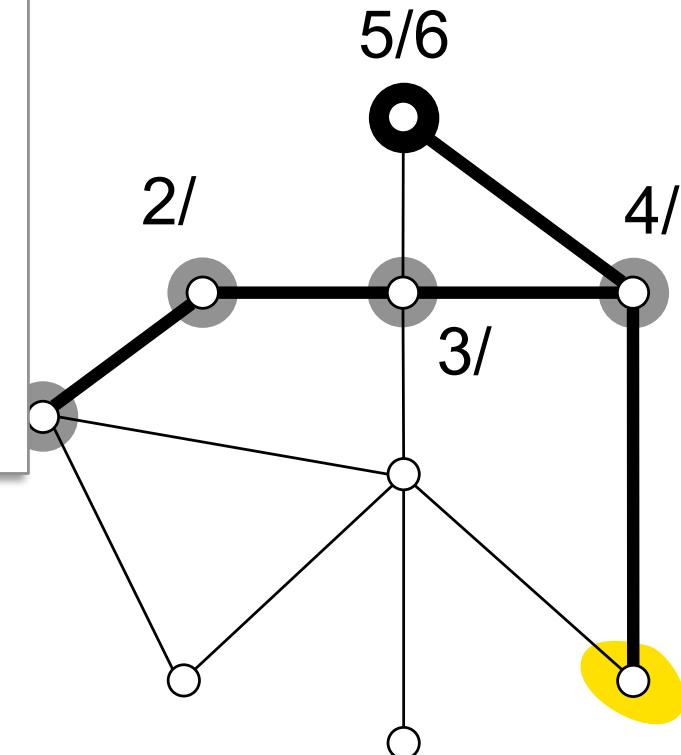
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

$u.D \leftarrow$  TEMPO

$u.COR \leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.ADJ[u]$

SE  $v.COR ==$  BRANCO ENTÃO

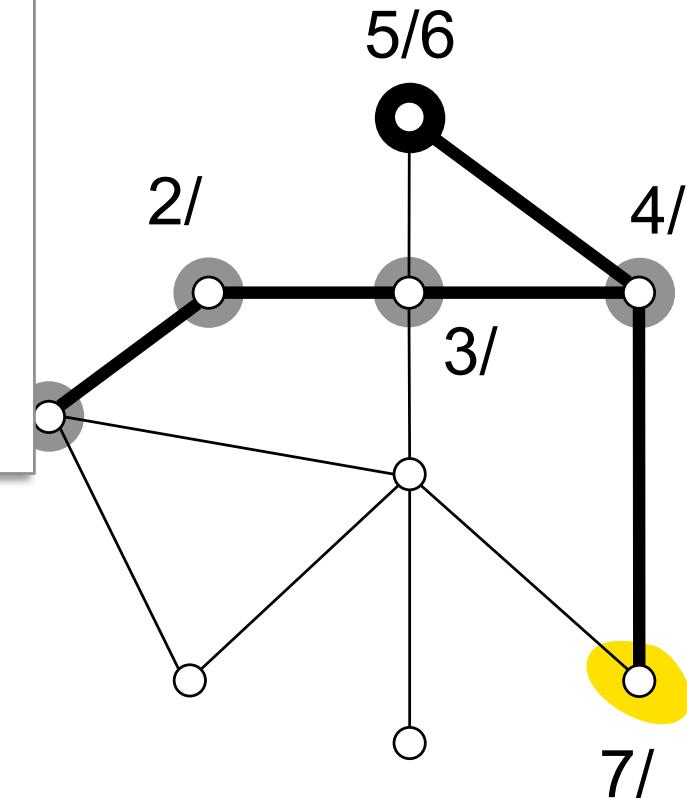
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$u.COR \leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

$u.F \leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

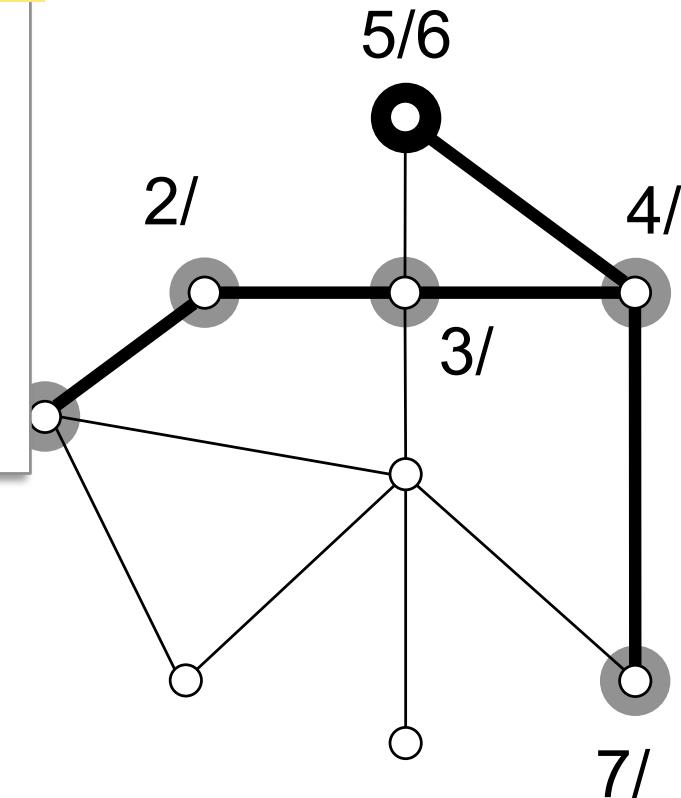
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

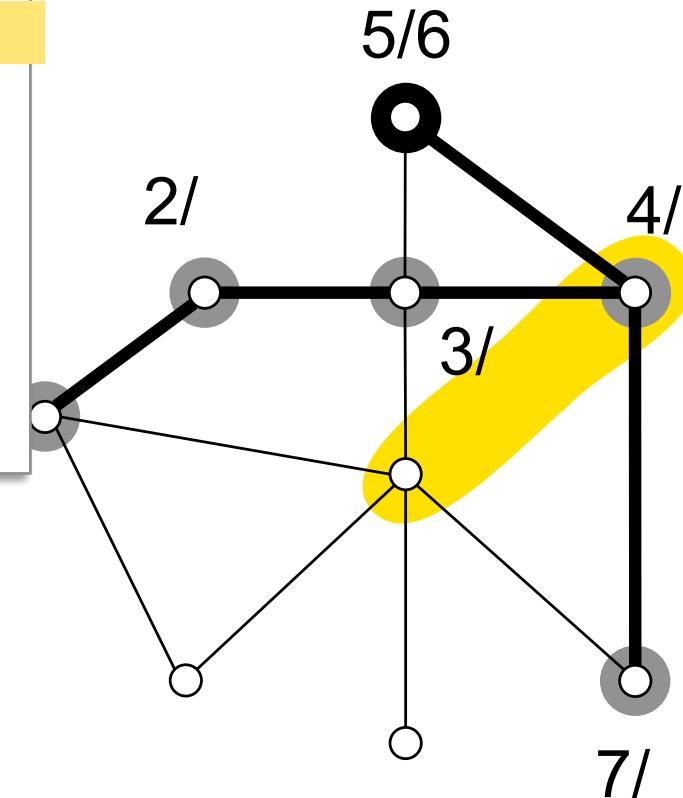
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

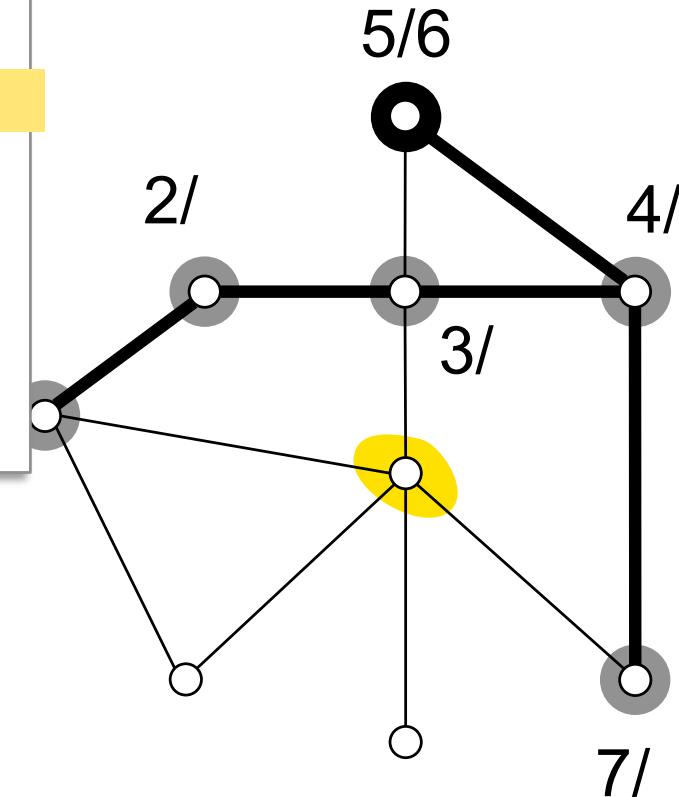
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

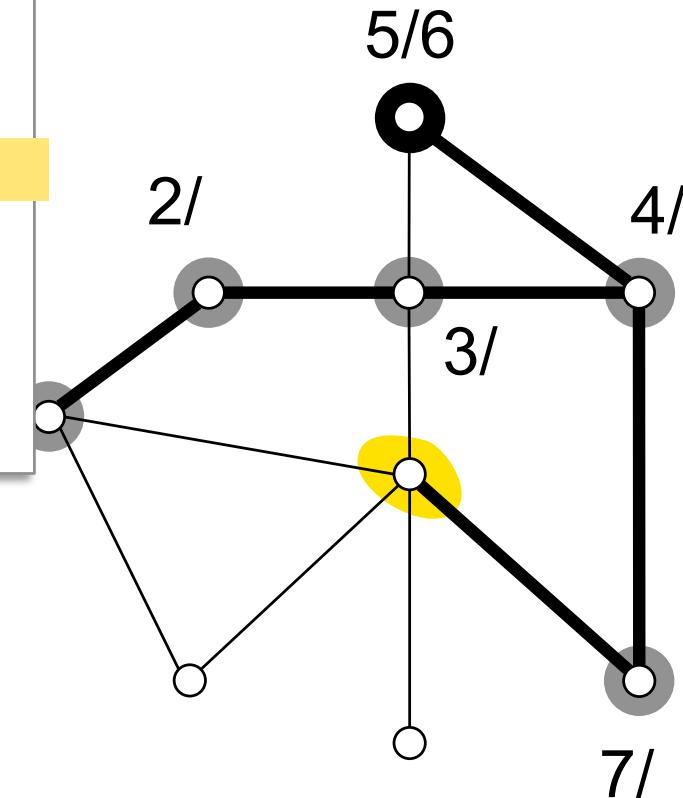
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

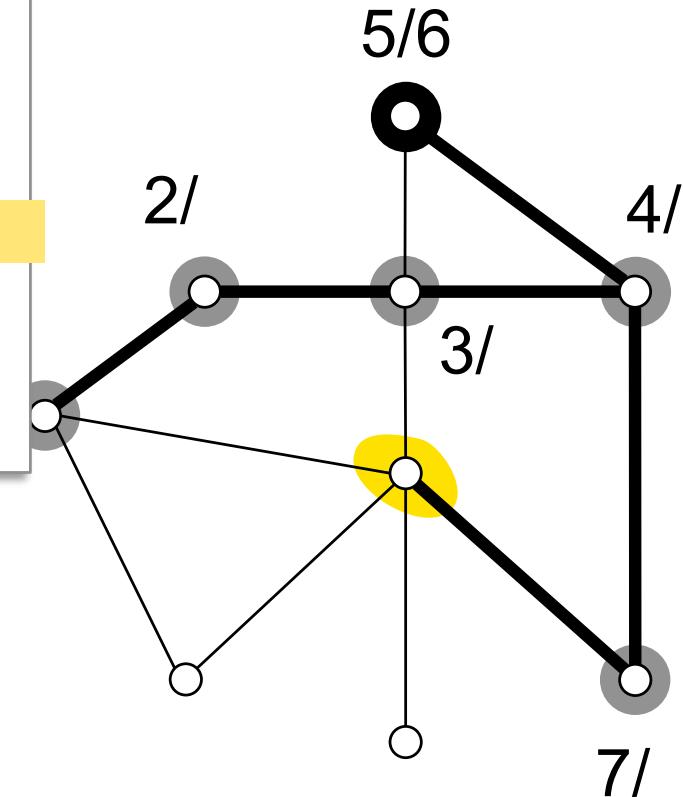
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



### ALGORITMO DFS\_VISITA( $G, u$ )

## ALGORITMO DFS VISITA(G, u)

## ALGORITMO DFS\_VISITA( $G, u$ )

$\text{TEMPO} \leftarrow \text{TEMPO} + 1$

*U.D* ← TEMPO

*U.COR* ← CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE V.COR == BRANCO ENTÃO

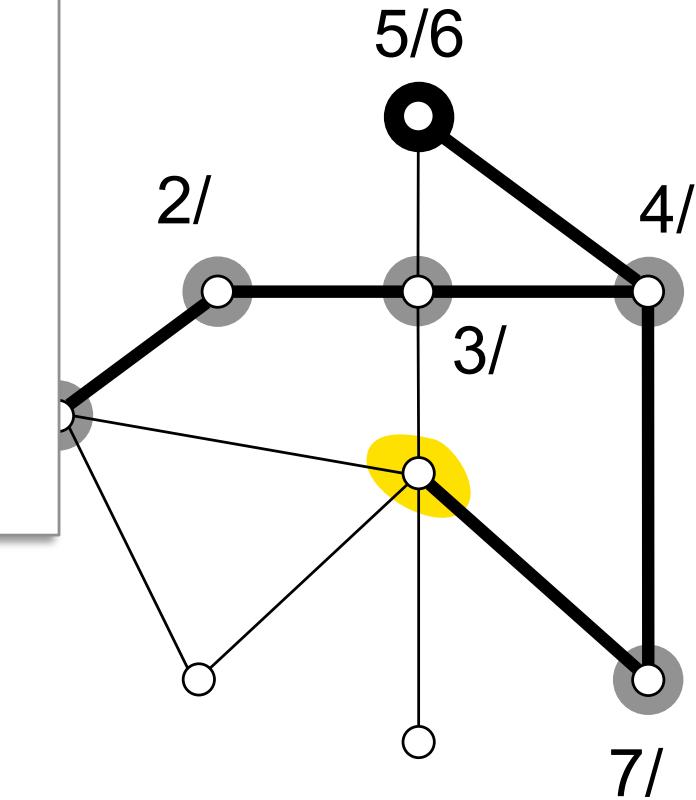
$V.\pi \leftarrow U$

DFS\_VISITA( $G, v$ )

*U.COR* ← PRETO

$\text{TEMPO} \leftarrow \text{TEMPO} + 1$

*U.F ← TEMPO*



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

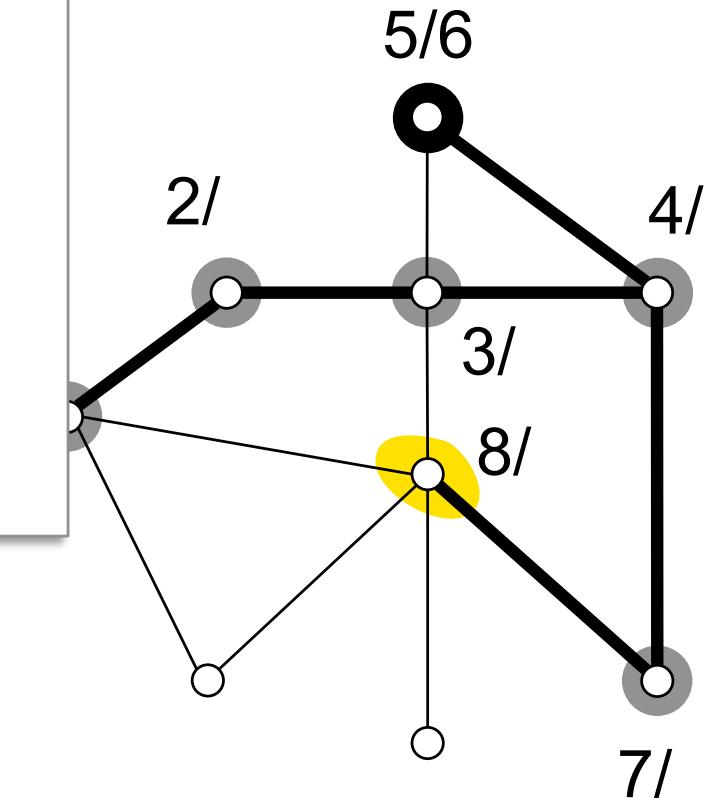
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

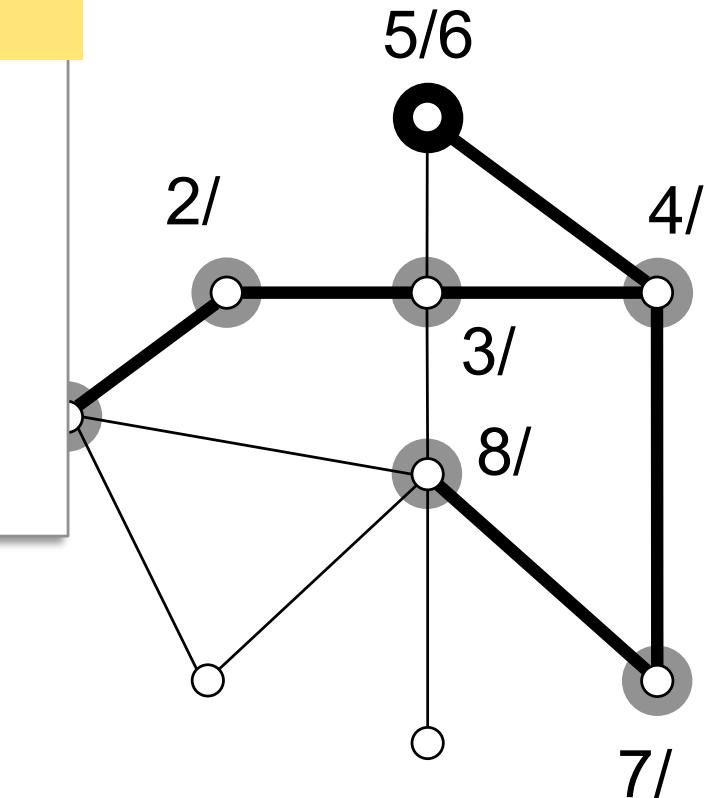
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

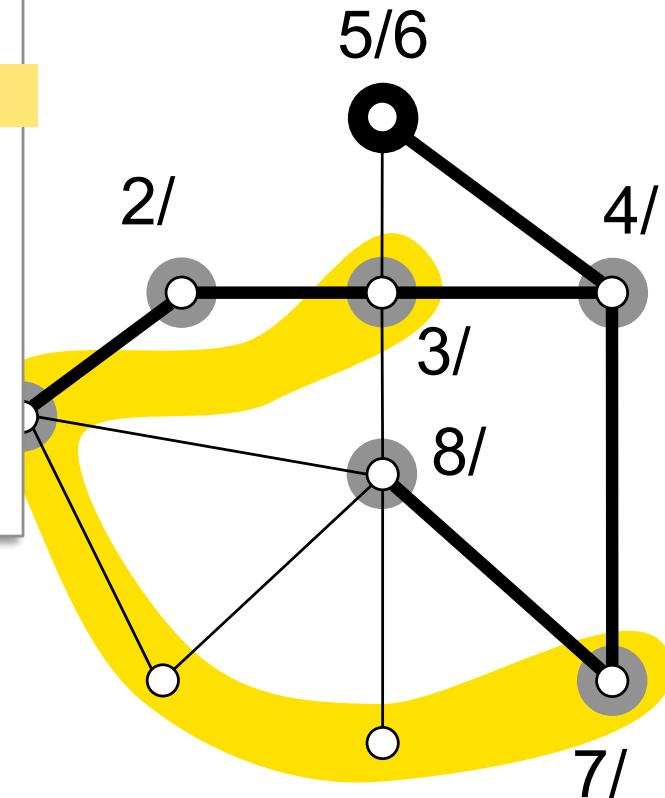
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

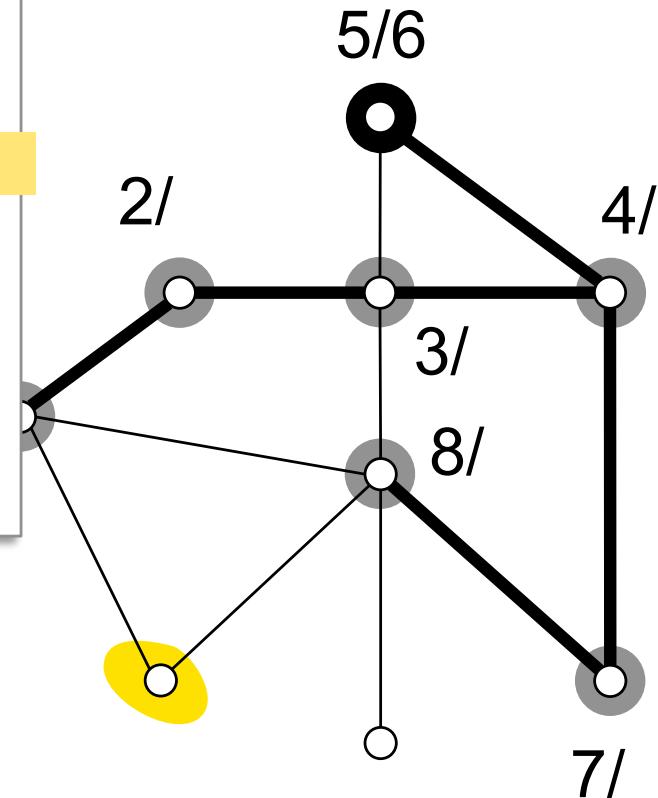
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

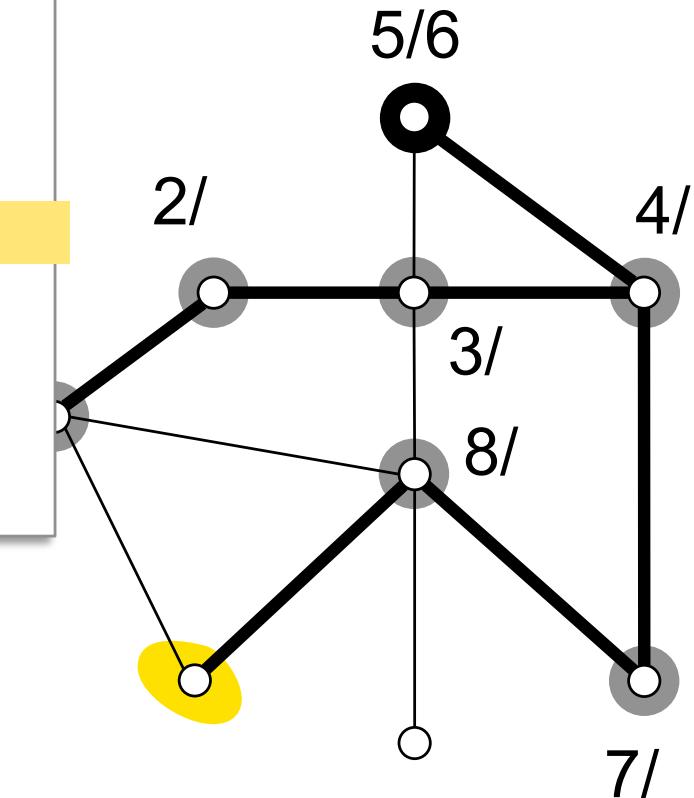
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

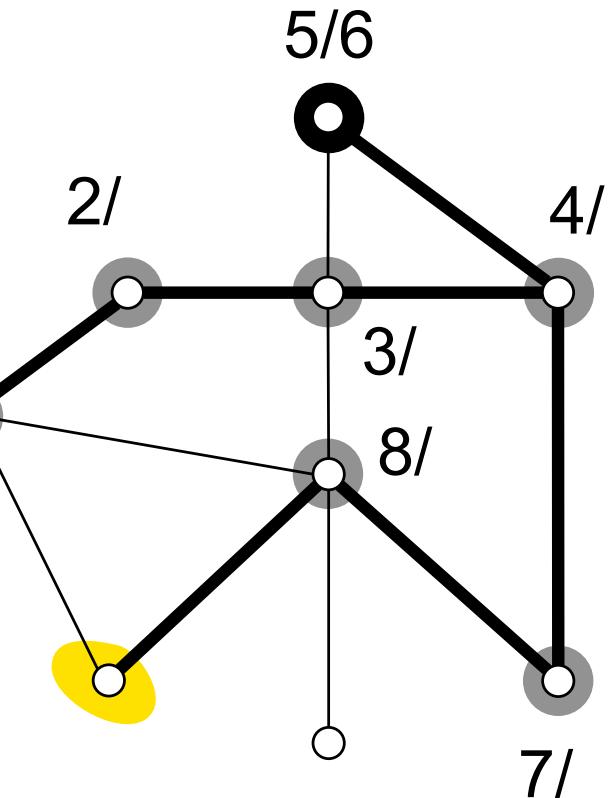
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



## ALGORITMO DFS\_VISITA( $G, u$ )

## ALGORITMO DFS VISITA(G, u)

## ALGORITMO DFS VISITA( $G, u$ )

## ALGORITMO DFS VISITA(G, u)

## ALGORITMO DFS VISITA(G, u)

## ALGORITMO DFS VISITA(G, u)

## ALGORITMO DFS\_VISITA( $G, u$ )

*TEMPO*  $\leftarrow$  *TEMPO* + 1

$U.D \leftarrow TEMPO$

*U.COR* ← CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE V.COR == BRANCO ENTÃO

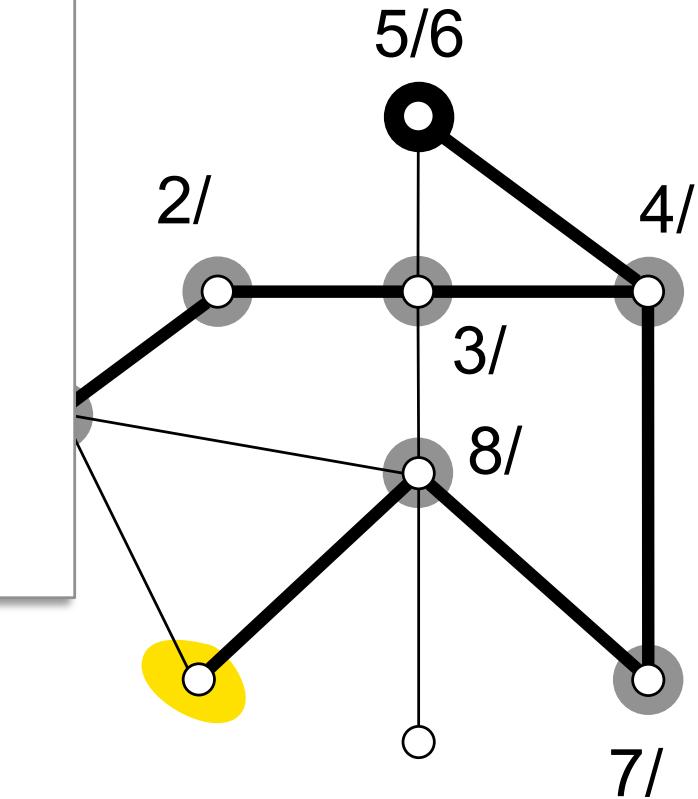
$V.\pi \leftarrow U$

DFS\_VISITA( $G, v$ )

$U.COR \leftarrow \text{PRETO}$

**TEMPO**  $\leftarrow$  **TEMPO** + 1

$U.F \leftarrow TEMPO$



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

$u.D \leftarrow$  TEMPO

$u.COR \leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.ADJ[u]$

SE  $v.COR ==$  BRANCO ENTÃO

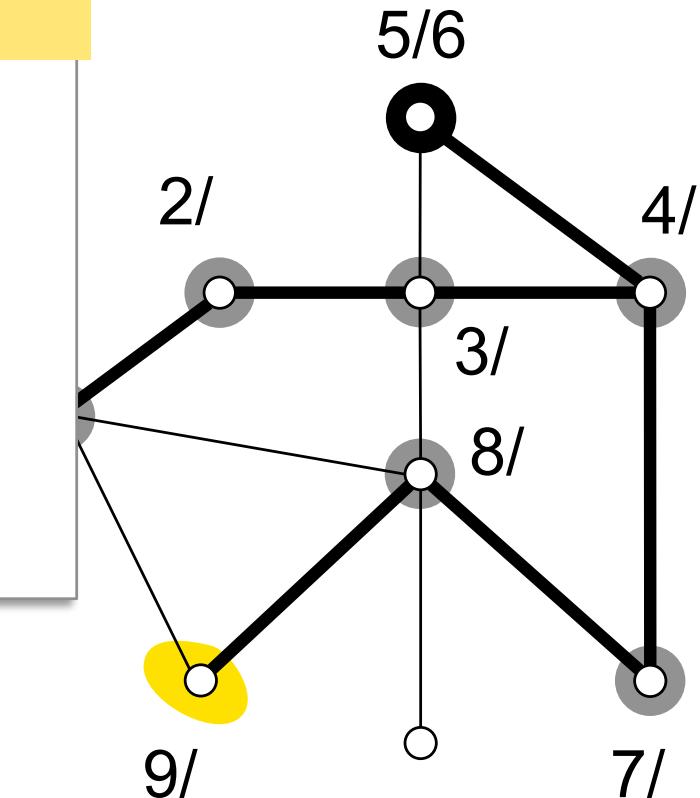
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$u.COR \leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

$u.F \leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

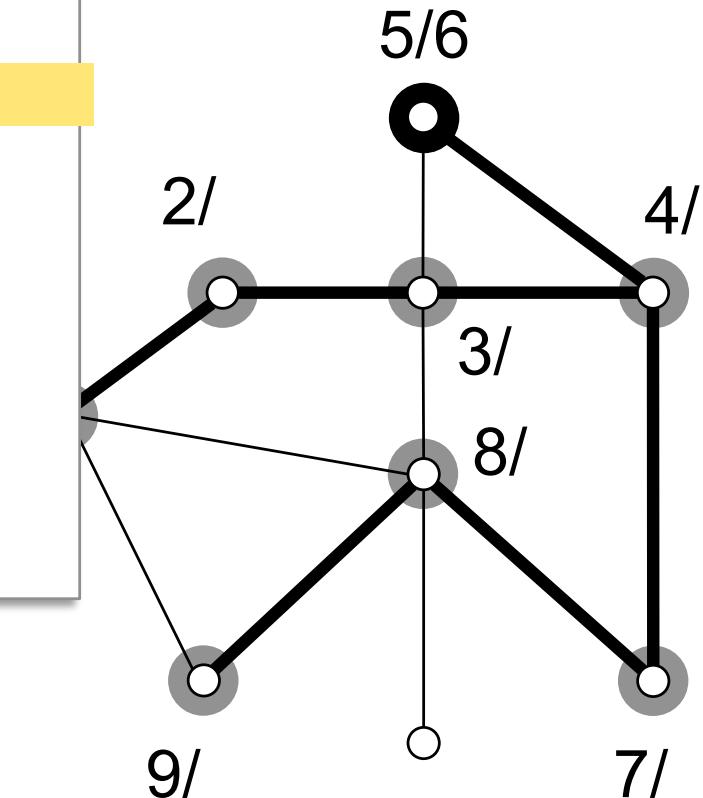
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

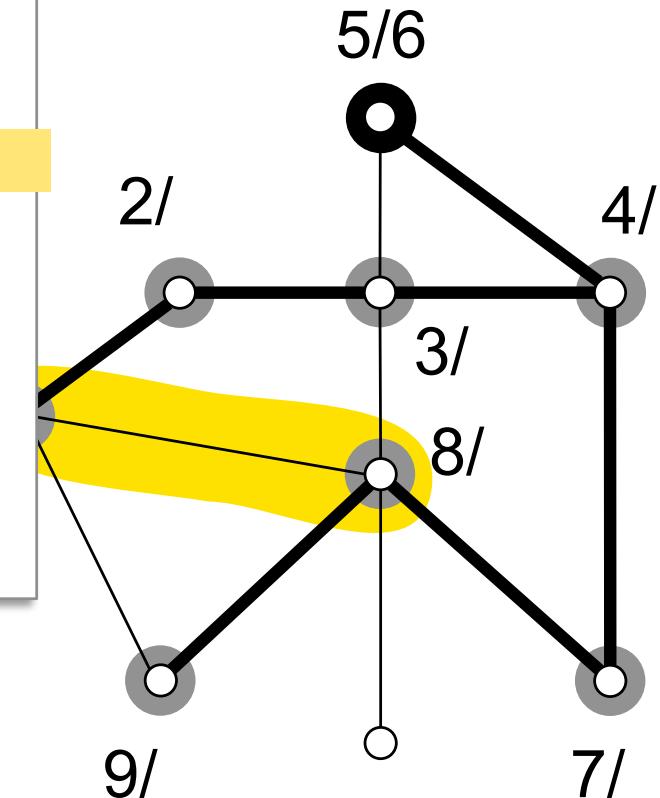
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

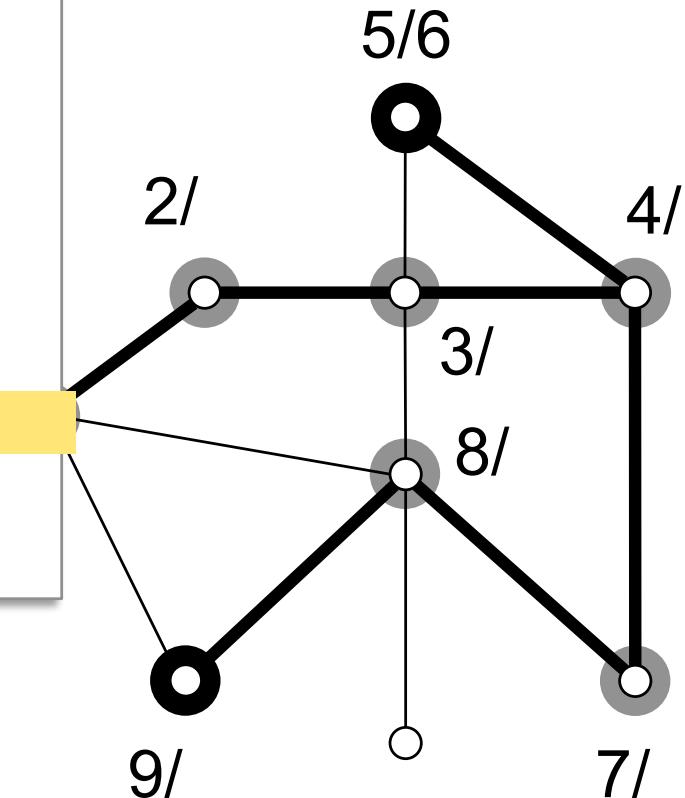
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

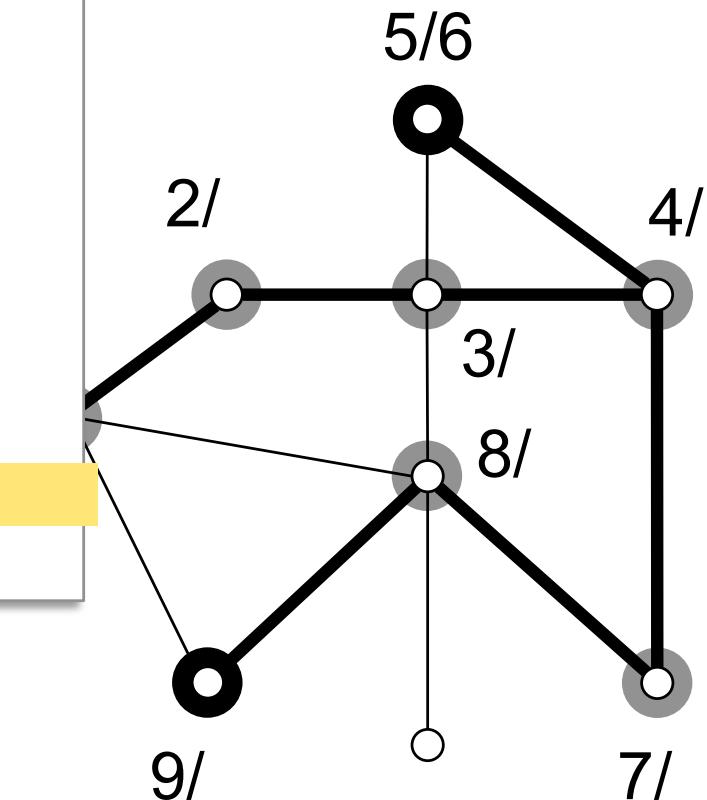
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

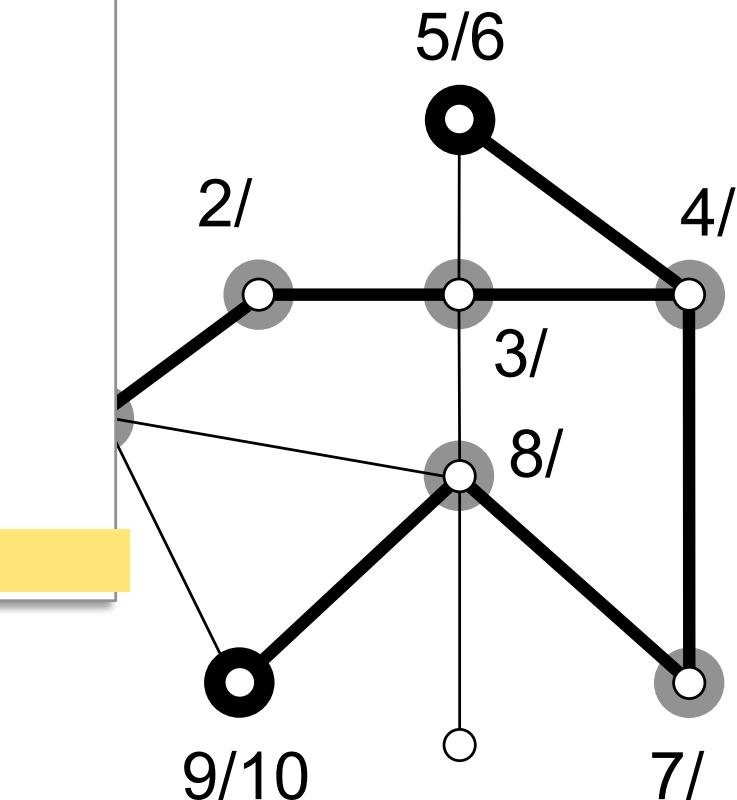
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

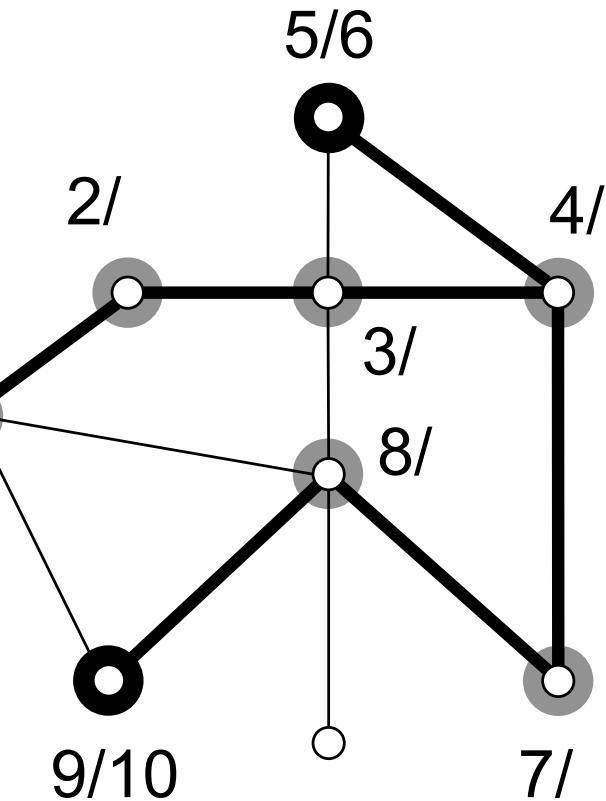
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

  ALGORITMO DFS\_VISITA( $G, u$ )

    ALGORITMO DFS\_VISITA( $G, u$ )

      ALGORITMO DFS\_VISITA( $G, u$ )

      ALGORITMO DFS\_VISITA( $G, u$ )

        TEMPO  $\leftarrow$  TEMPO + 1

        U.D  $\leftarrow$  TEMPO

        U.COR  $\leftarrow$  CINZA

        PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

          SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

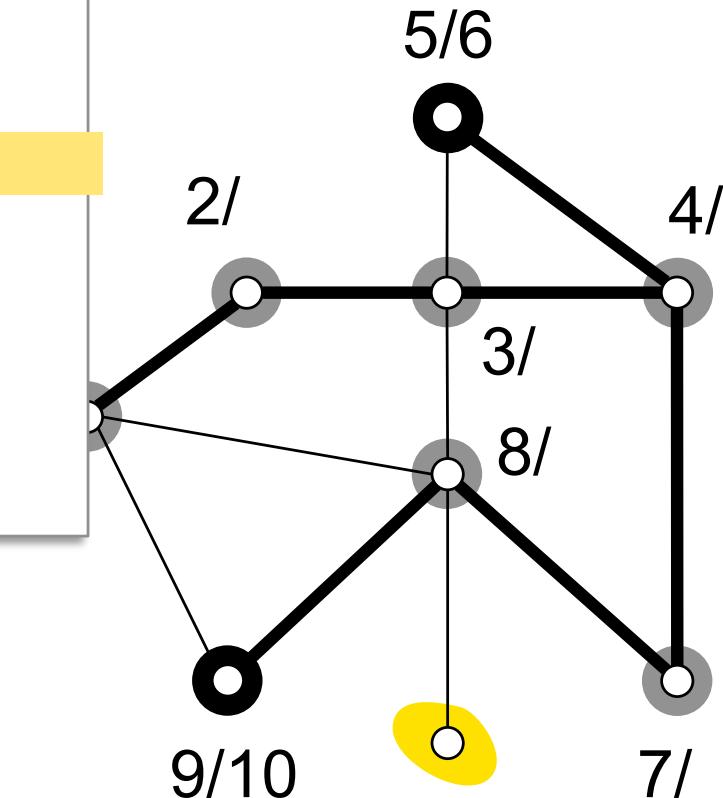
$v.\pi \leftarrow u$

            DFS\_VISITA( $G, v$ )

        U.COR  $\leftarrow$  PRETO

        TEMPO  $\leftarrow$  TEMPO + 1

        U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

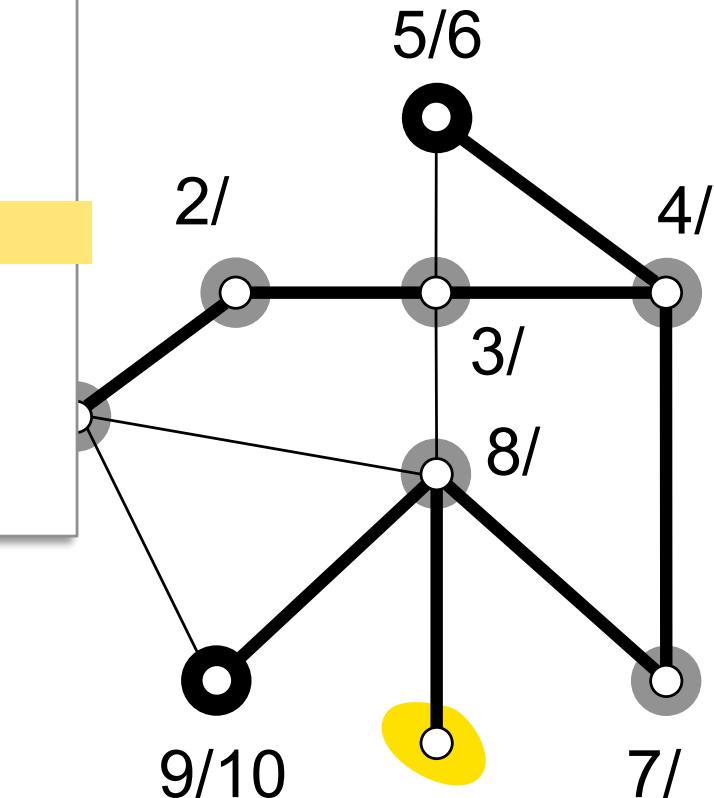
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

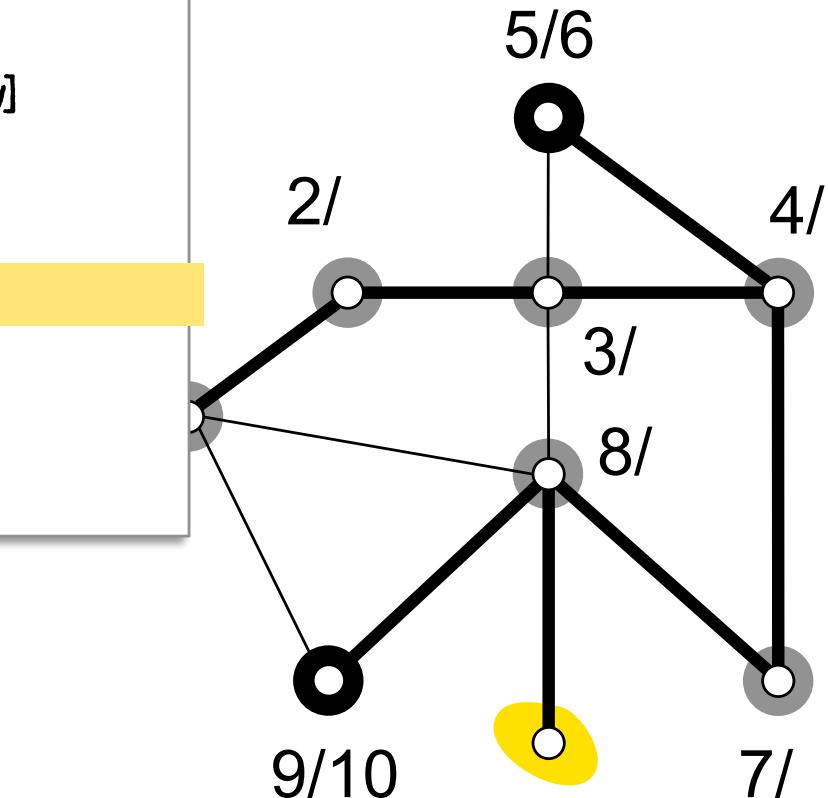
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

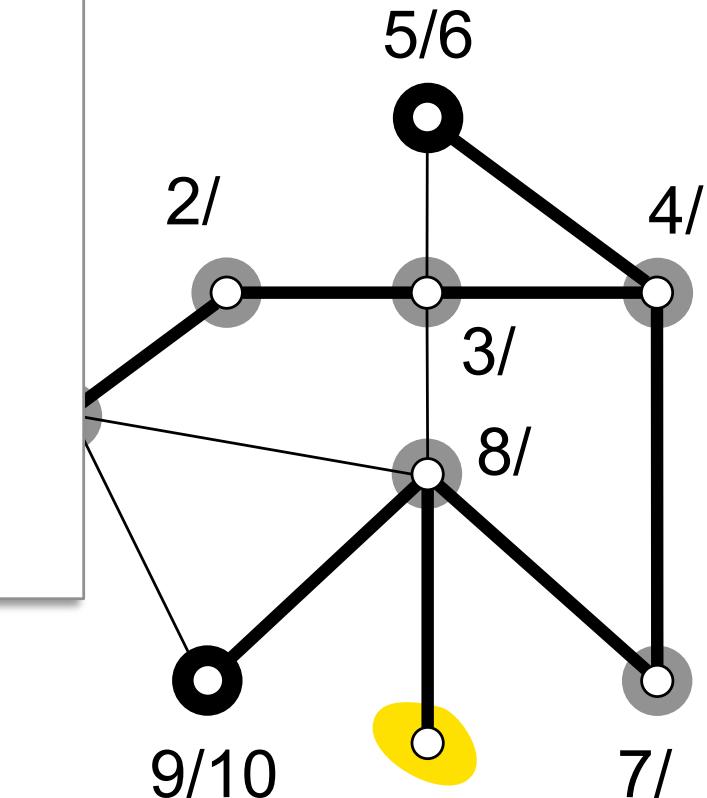
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

$u.D \leftarrow$  TEMPO

$u.COR \leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.ADJ[u]$

SE  $v.COR ==$  BRANCO ENTÃO

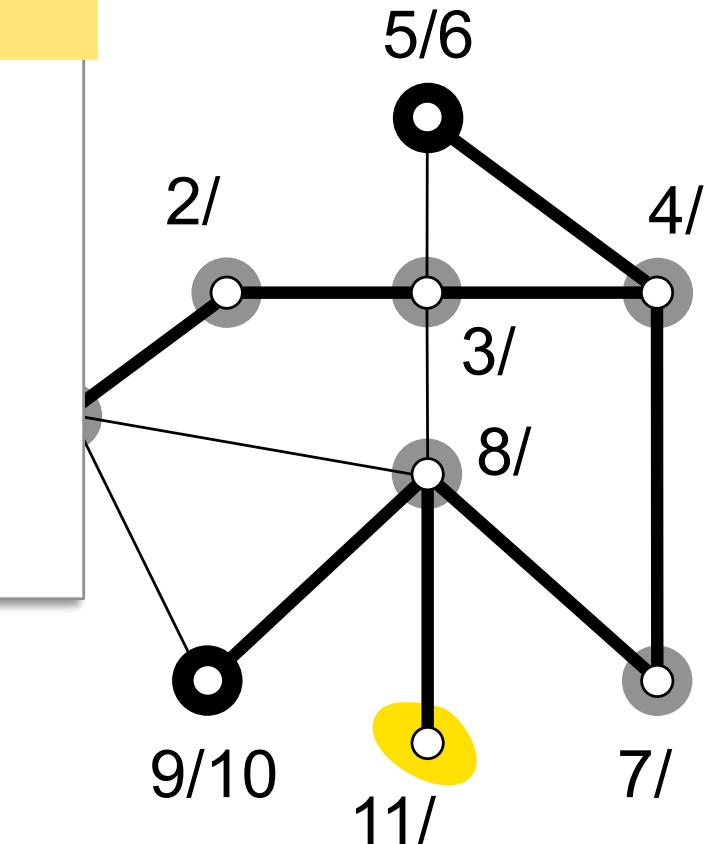
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$u.COR \leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

$u.F \leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

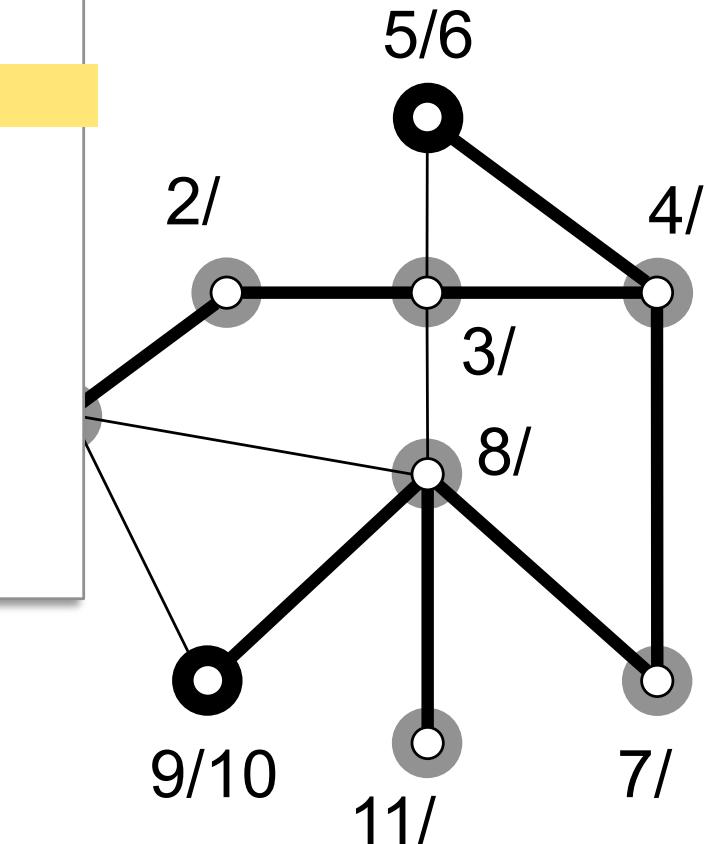
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

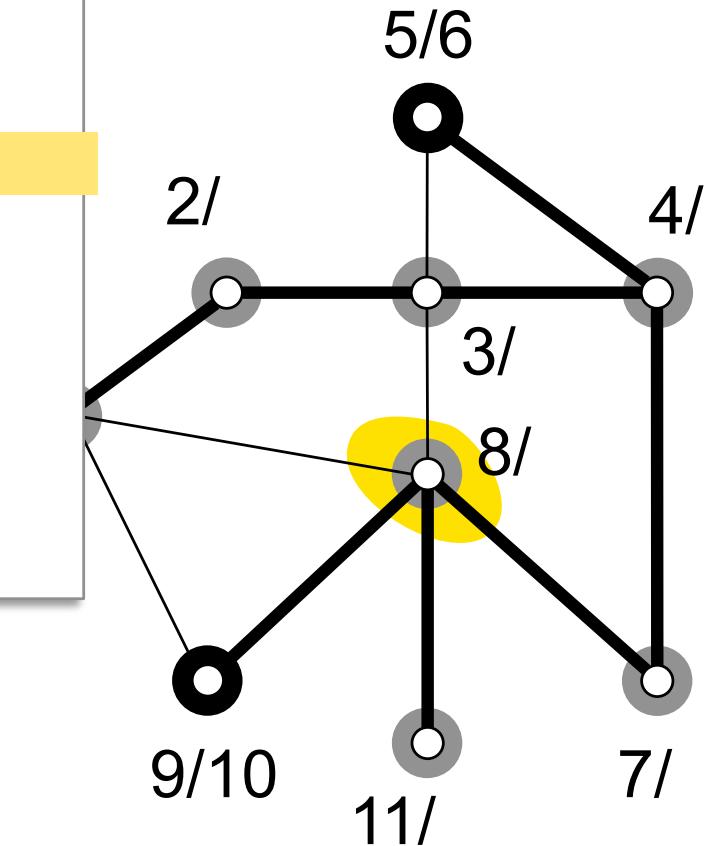
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

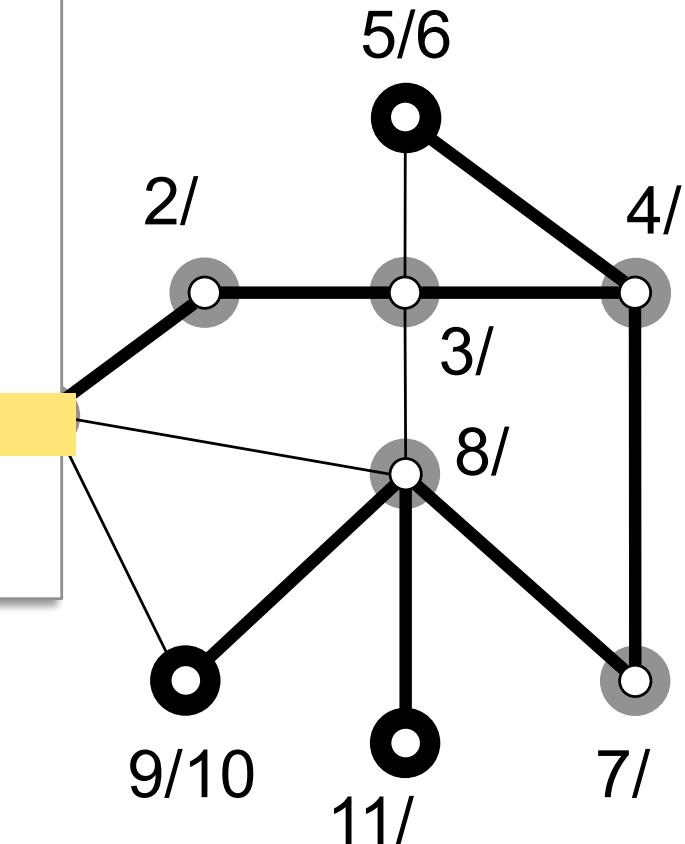
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

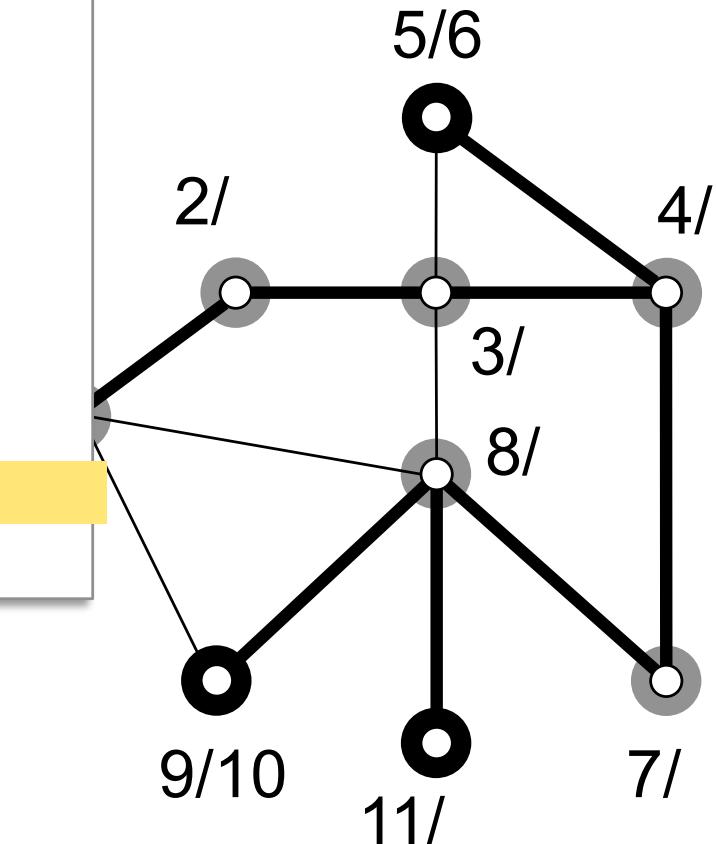
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

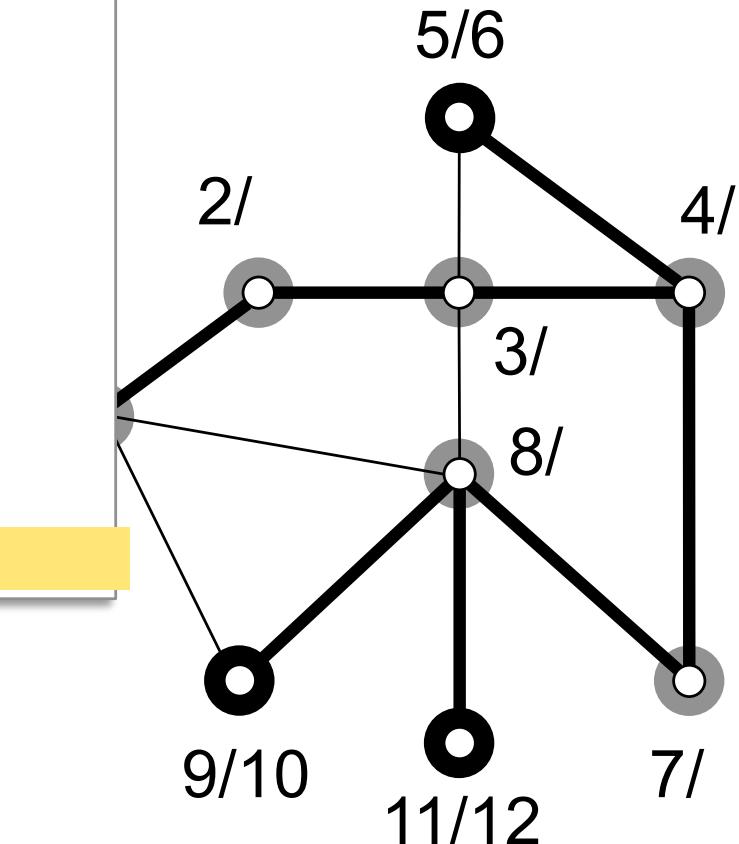
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

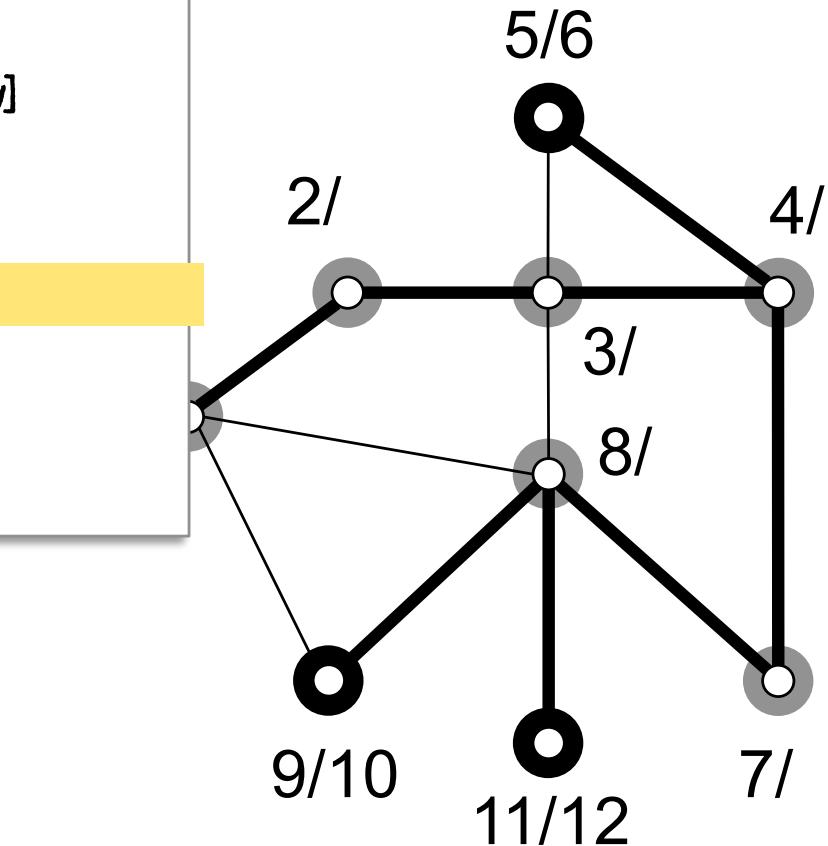
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

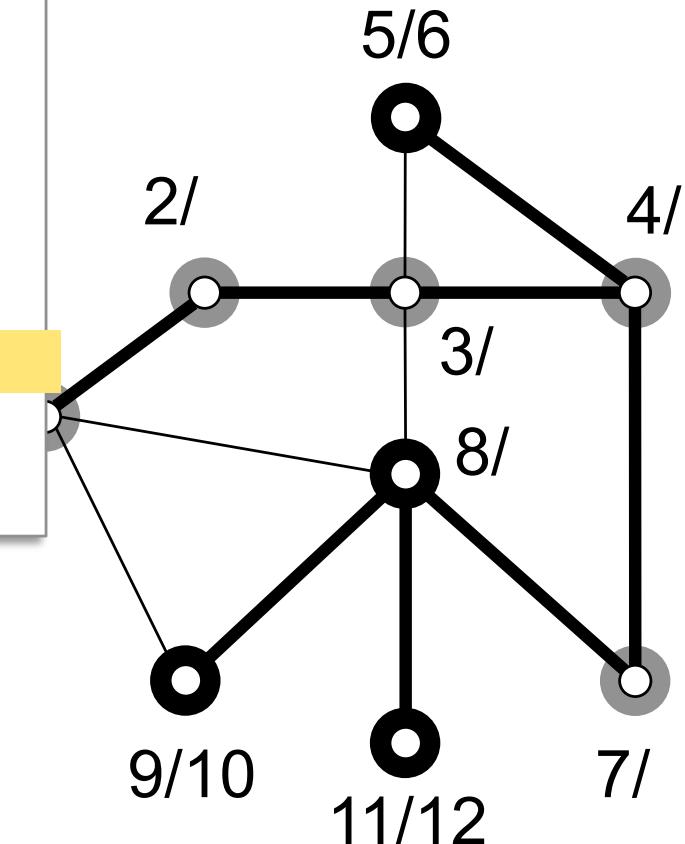
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

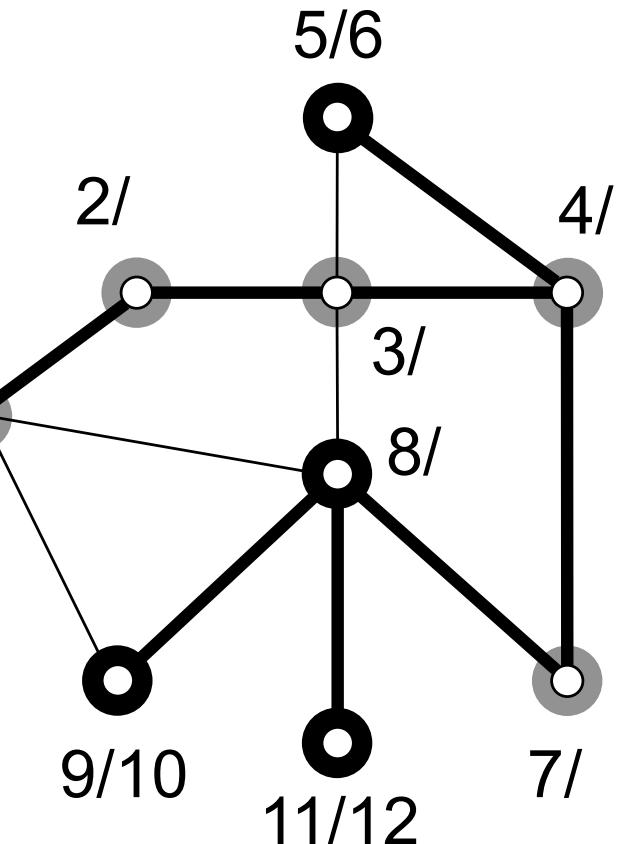
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

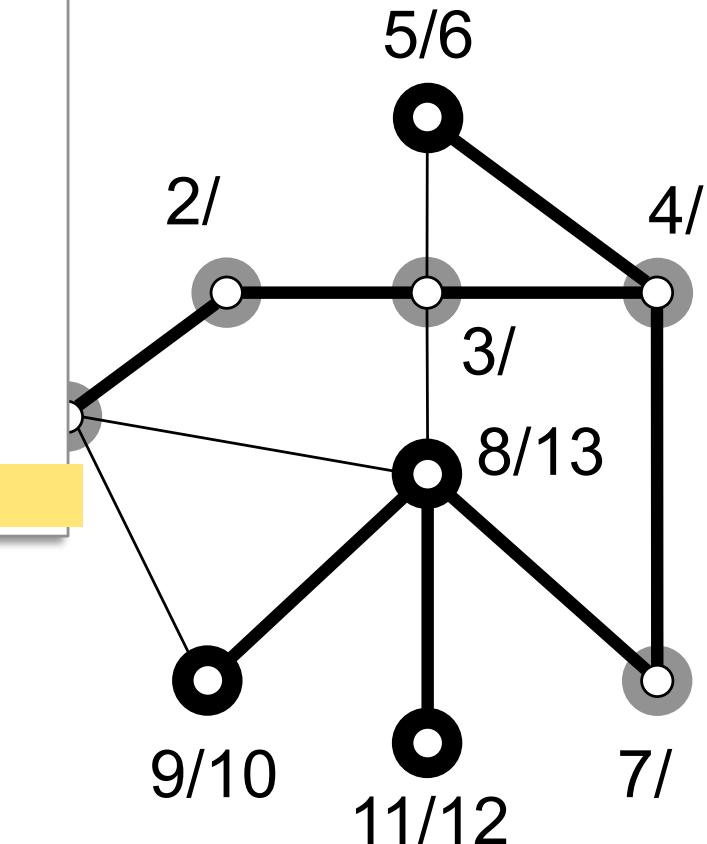
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

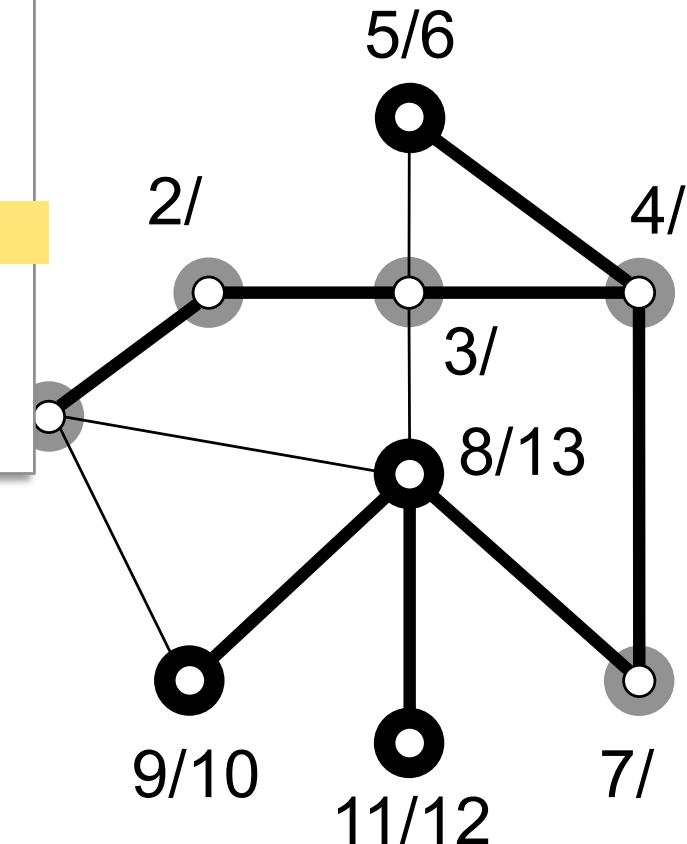
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

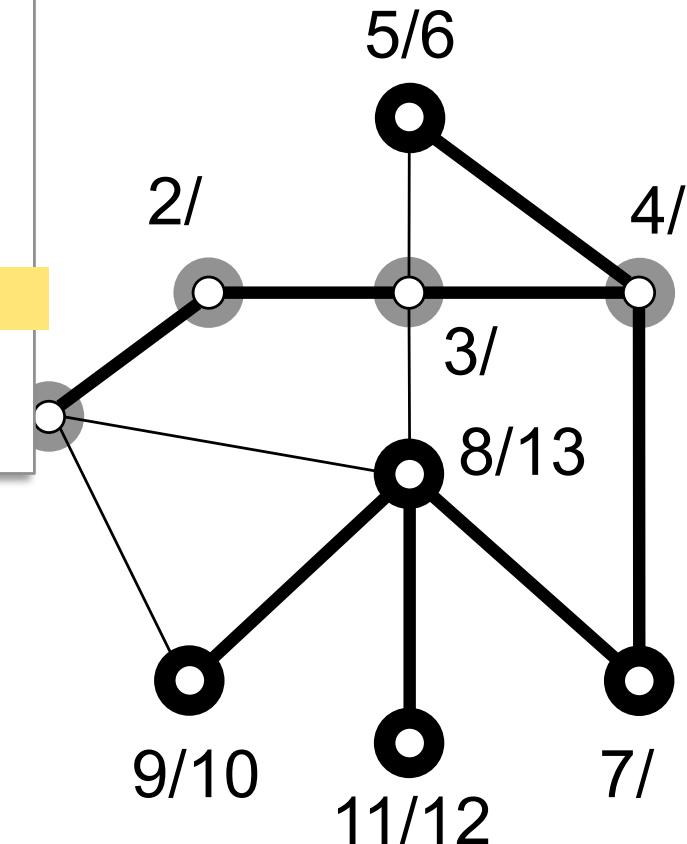
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

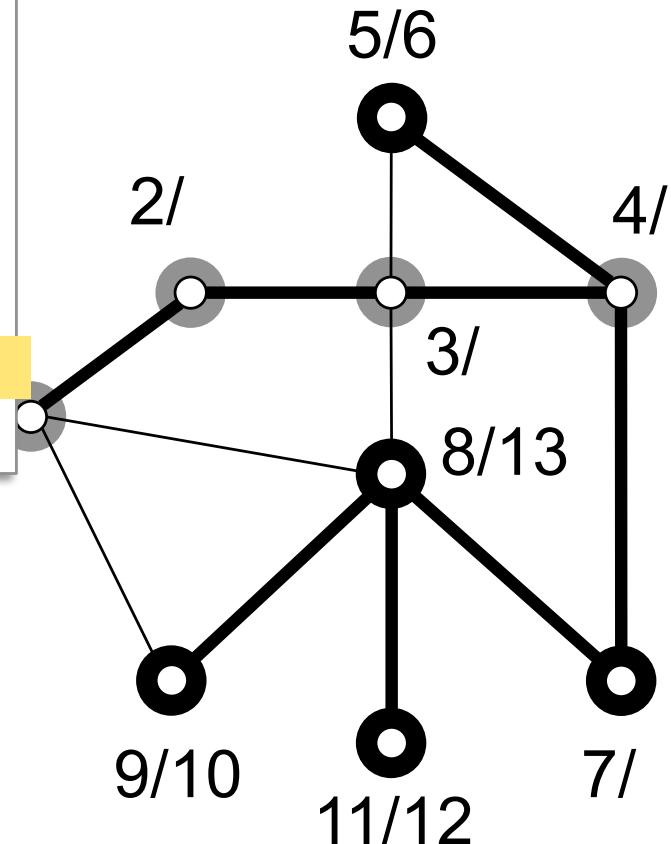
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

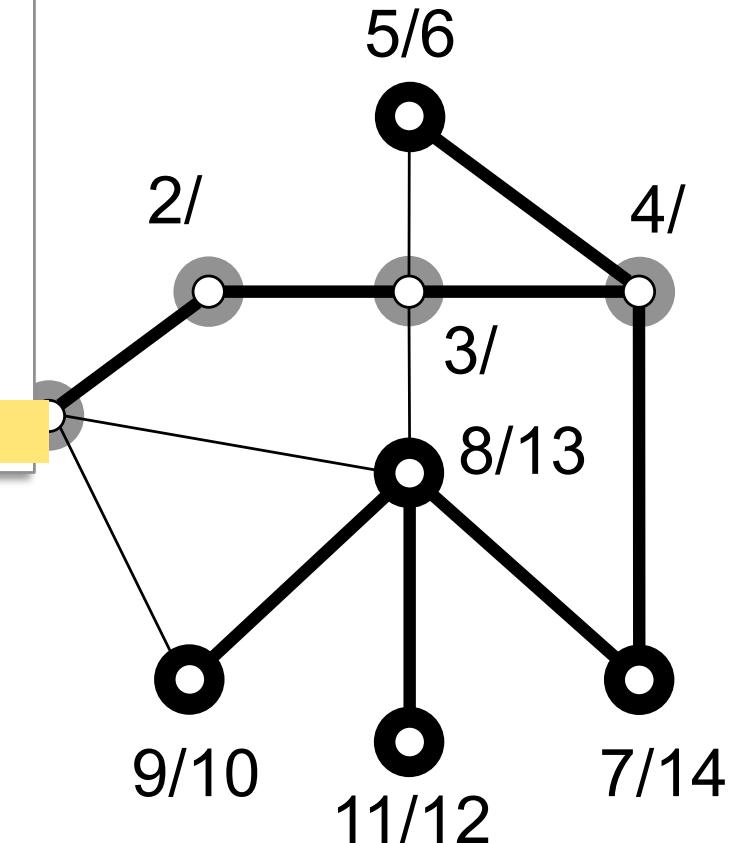
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

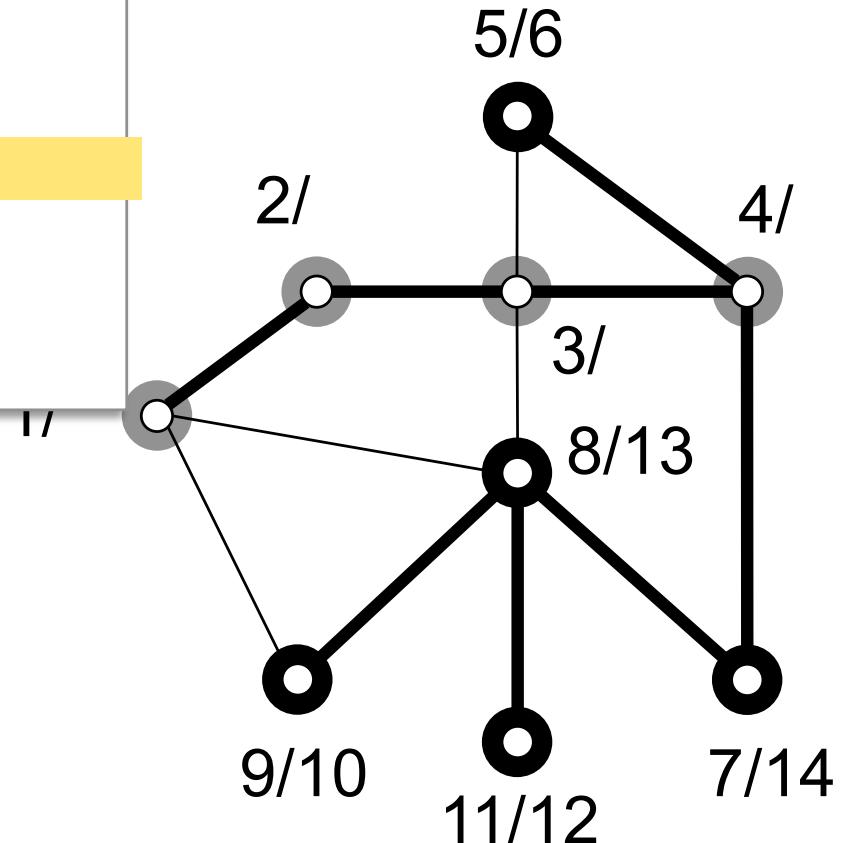
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

  ALGORITMO DFS\_VISITA( $G, u$ )

    ALGORITMO DFS\_VISITA( $G, u$ )

      ALGORITMO DFS\_VISITA( $G, u$ )

       TEMPO  $\leftarrow$  TEMPO + 1

.D  $\leftarrow$  TEMPO

.COR  $\leftarrow$  CINZA

       PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

          SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

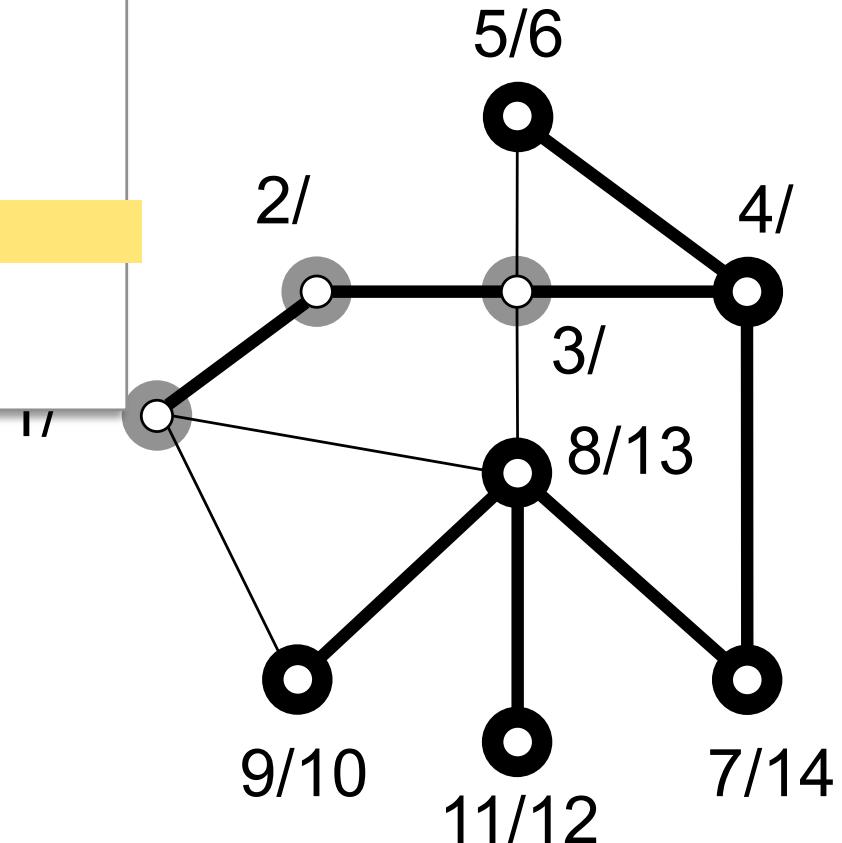
$v.\pi \leftarrow u$

            DFS\_VISITA( $G, v$ )

.COR  $\leftarrow$  PRETO

       TEMPO  $\leftarrow$  TEMPO + 1

.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

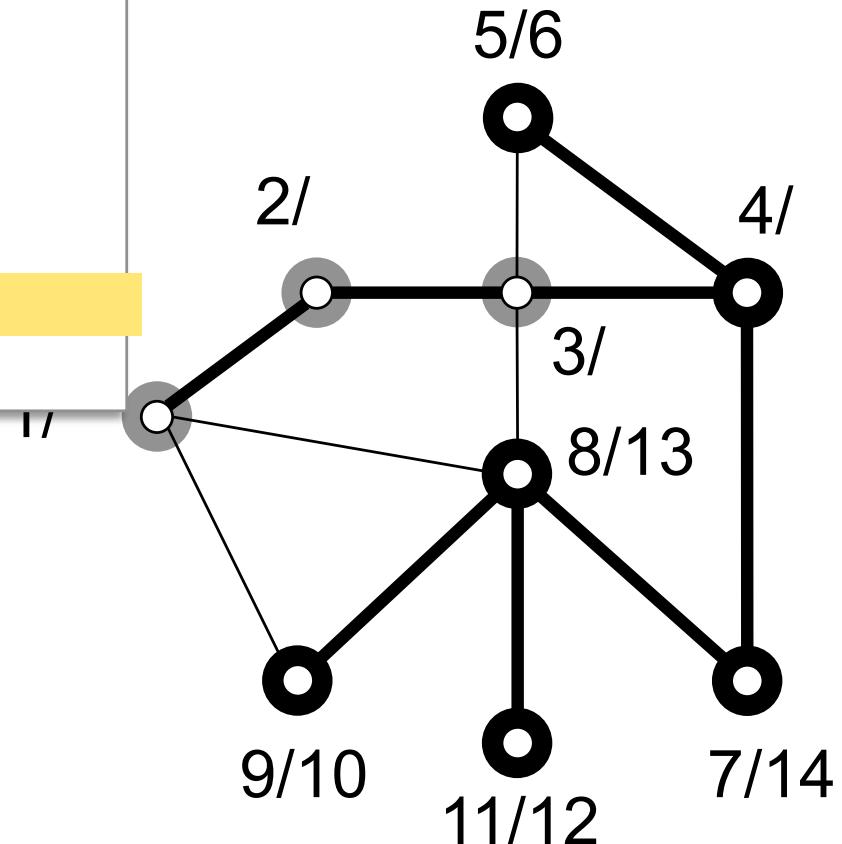
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



## ALGORITMO DFS\_VISITA( $G, u$ )

## ALGORITMO DFS VISITA( $G, u$ )

## ALGORITMO DFS VISITA(G, u)

**ALGORITMO DFS\_VISITA( $G, u$ )**

$TEMPO \leftarrow TEMPO + 1$

$U.D \leftarrow TEMPO$

*U.COR* ← CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE V.COR == BRANCO ENTÃO

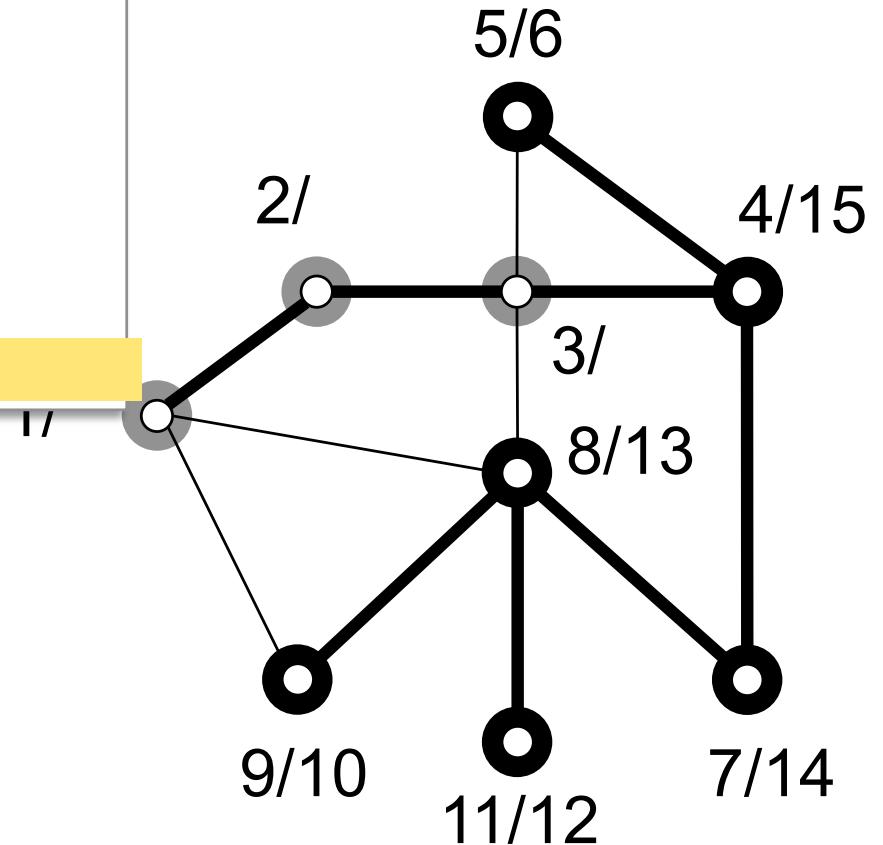
V.π ← U

**DFS\_VISITA( $G, v$ )**

*U.COR* ← PRETO

$\text{TEMPO} \leftarrow \text{TEMPO} + 1$

$U.F \leftarrow TEMPO$



ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

$u.D \leftarrow$  TEMPO

$u.COR \leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.COR ==$  BRANCO ENTÃO

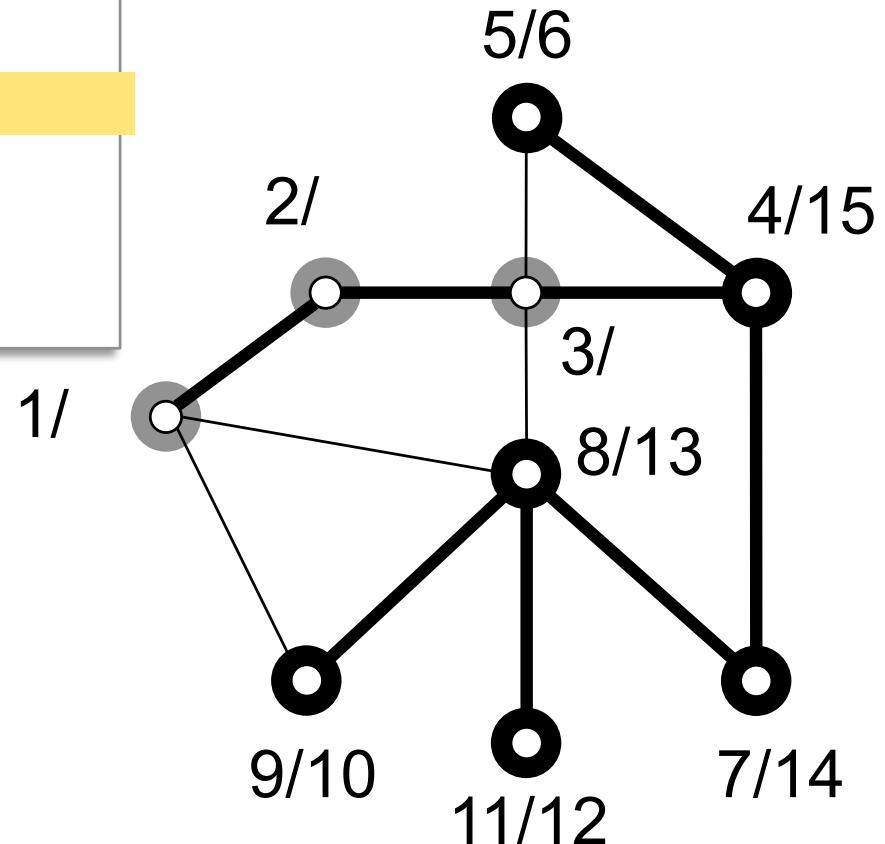
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$u.COR \leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

$u.F \leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

$U.D \leftarrow$  TEMPO

$U.COR \leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.COR ==$  BRANCO ENTÃO

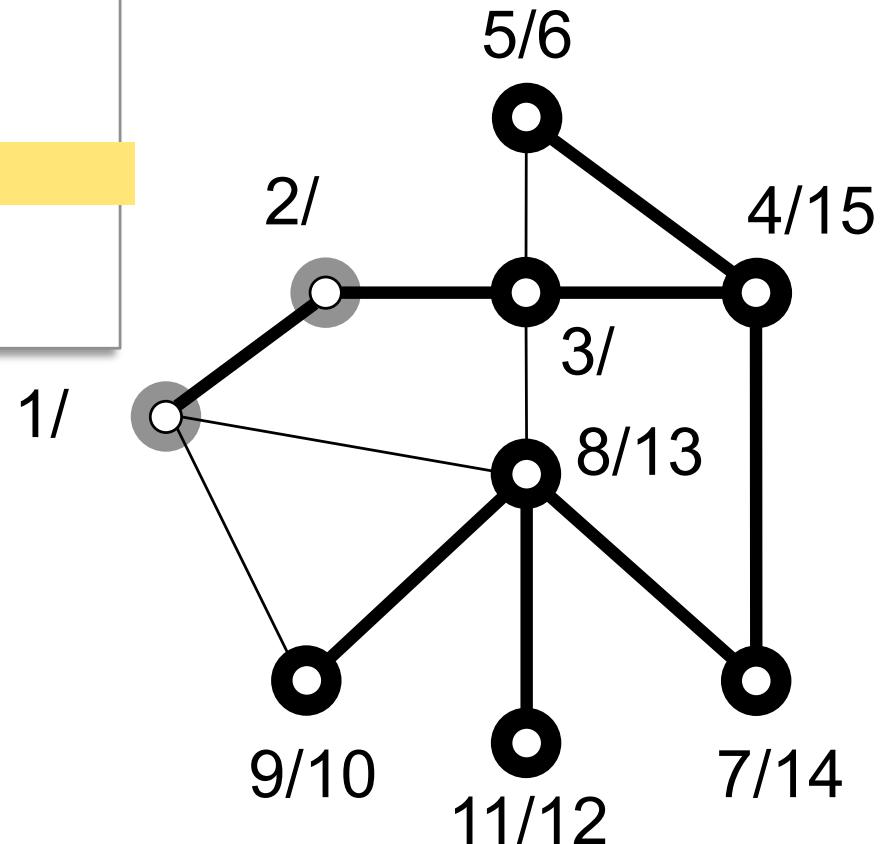
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$U.COR \leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

$U.F \leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

U.D  $\leftarrow$  TEMPO

U.COR  $\leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO

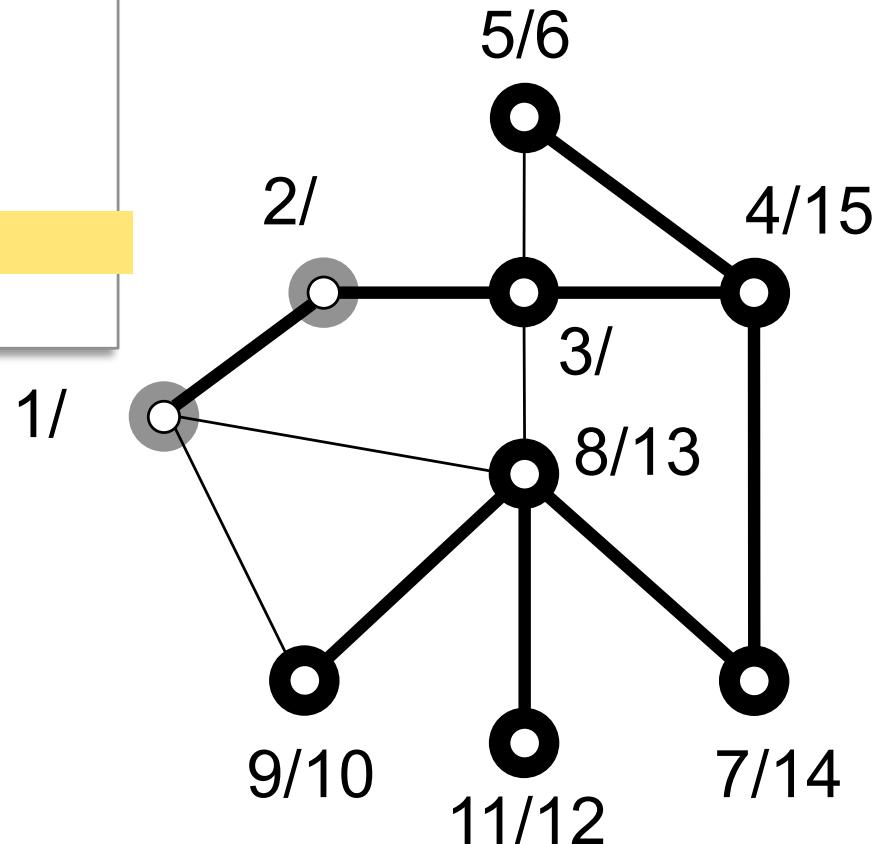
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

U.COR  $\leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

U.F  $\leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

TEMPO  $\leftarrow$  TEMPO + 1

$U.D \leftarrow$  TEMPO

$U.COR \leftarrow$  CINZA

PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$

SE  $v.COR ==$  BRANCO ENTÃO

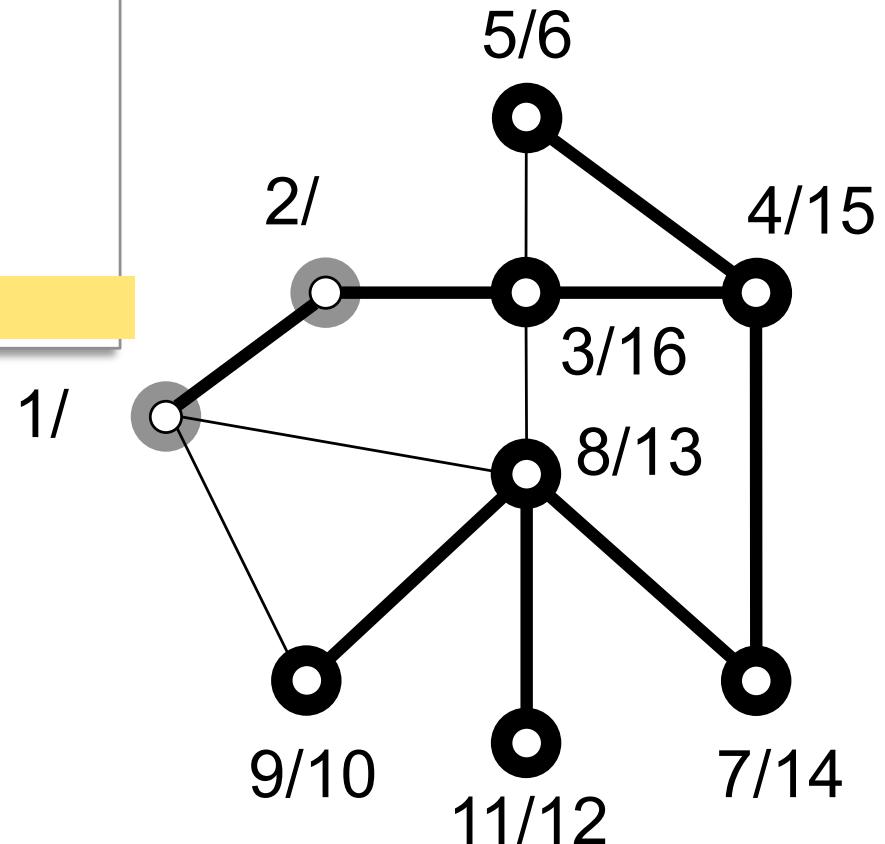
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$U.COR \leftarrow$  PRETO

TEMPO  $\leftarrow$  TEMPO + 1

$U.F \leftarrow$  TEMPO



ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

$TEMPO \leftarrow TEMPO + 1$

$U.D \leftarrow TEMPO$

$U.COR \leftarrow CINZA$

PARA CADA VÉRTICE  $v \in G.ADJ[u]$

SE  $v.COR == BRANCO$  ENTÃO

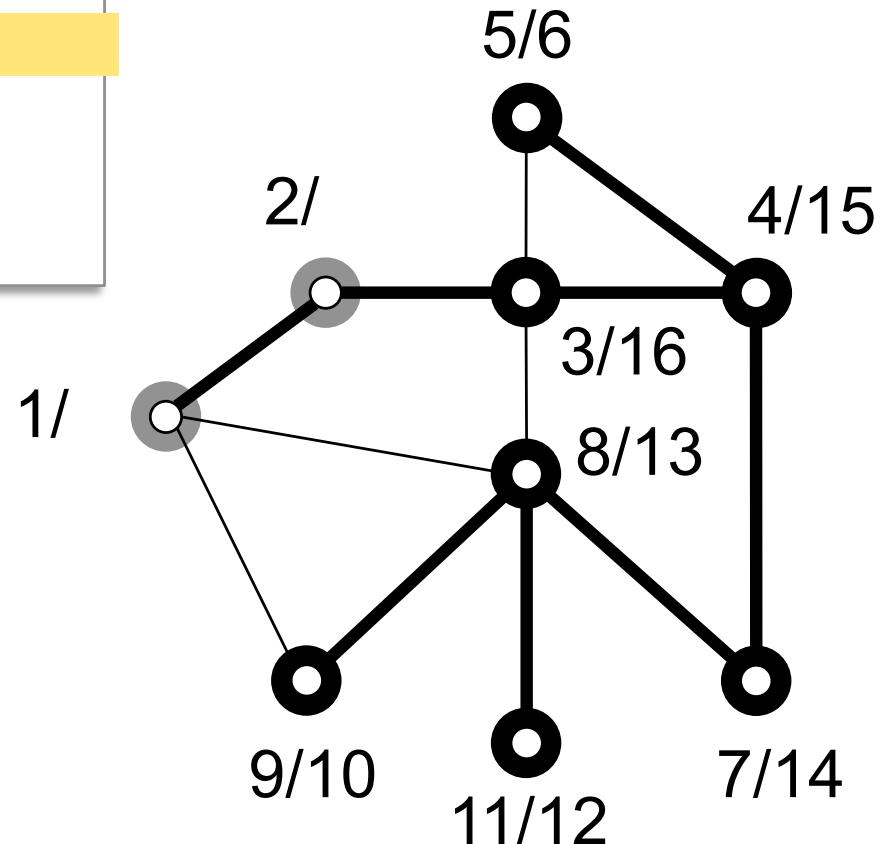
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$U.COR \leftarrow PRETO$

$TEMPO \leftarrow TEMPO + 1$

$U.F \leftarrow TEMPO$



ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

$TEMPO \leftarrow TEMPO + 1$

$U.D \leftarrow TEMPO$

$U.COR \leftarrow CINZA$

PARA CADA VÉRTICE  $v \in G.ADJ[u]$

SE  $v.COR == BRANCO$  ENTÃO

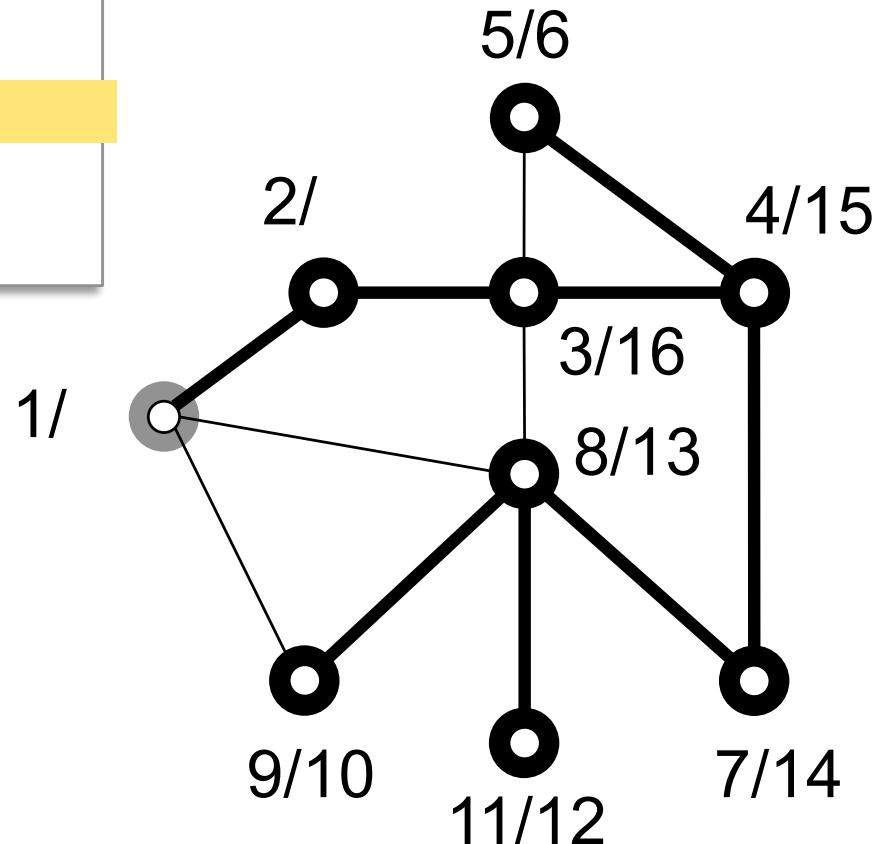
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$U.COR \leftarrow PRETO$

$TEMPO \leftarrow TEMPO + 1$

$U.F \leftarrow TEMPO$



ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

$TEMPO \leftarrow TEMPO + 1$

$U.D \leftarrow TEMPO$

$U.COR \leftarrow CINZA$

PARA CADA VÉRTICE  $v \in G.ADJ[u]$

SE  $v.COR == BRANCO$  ENTÃO

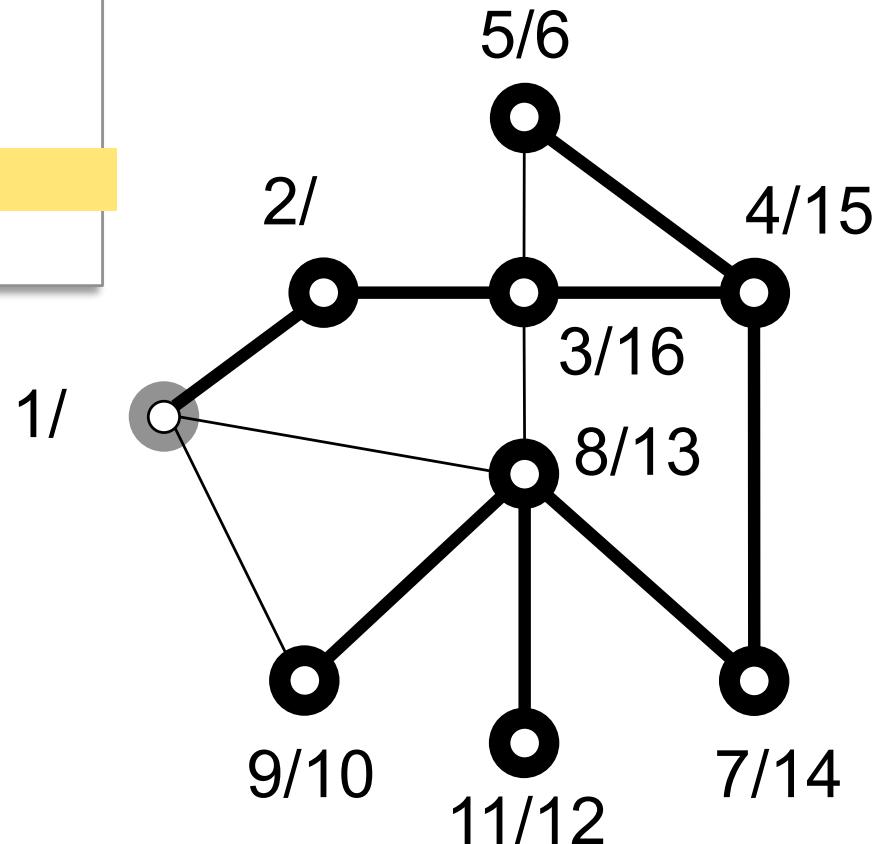
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$U.COR \leftarrow PRETO$

$TEMPO \leftarrow TEMPO + 1$

$U.F \leftarrow TEMPO$



ALGORITMO DFS\_VISITA( $G, u$ )

ALGORITMO DFS\_VISITA( $G, u$ )

$TEMPO \leftarrow TEMPO + 1$

$U.D \leftarrow TEMPO$

$U.COR \leftarrow CINZA$

PARA CADA VÉRTICE  $v \in G.ADJ[u]$

SE  $v.COR == BRANCO$  ENTÃO

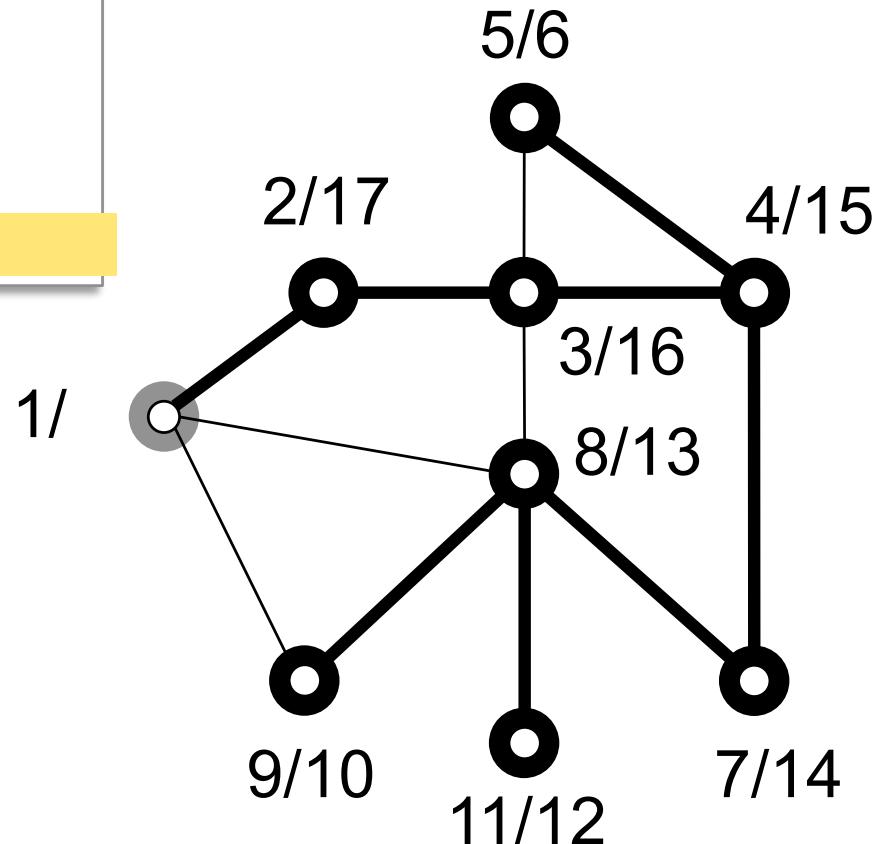
$v.\pi \leftarrow u$

DFS\_VISITA( $G, v$ )

$U.COR \leftarrow PRETO$

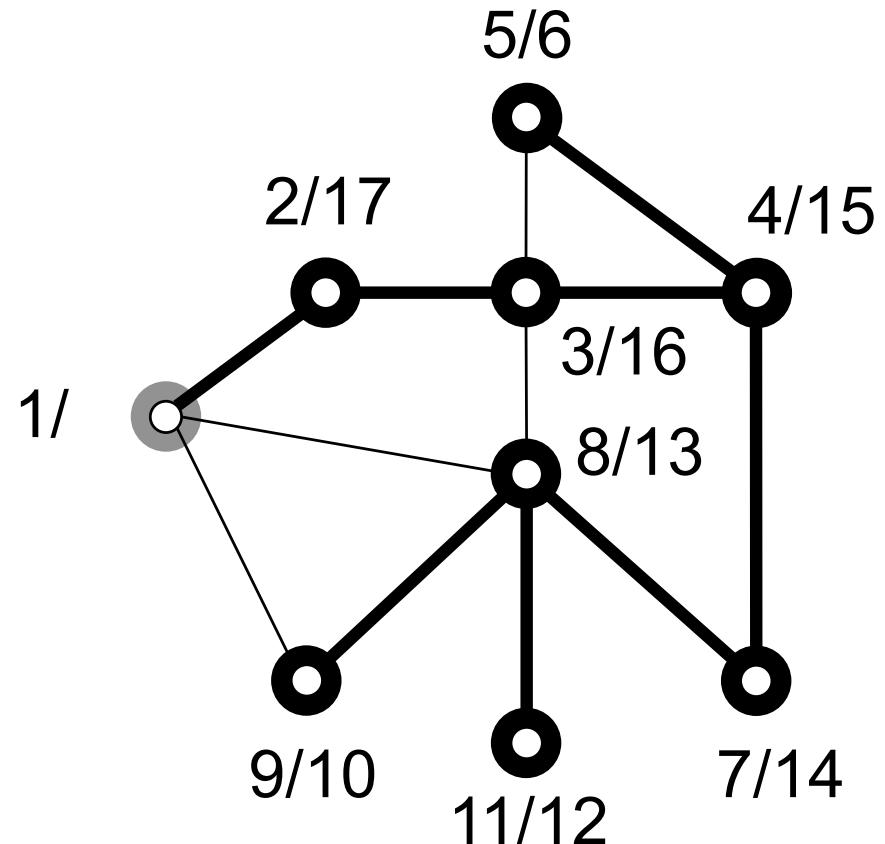
$TEMPO \leftarrow TEMPO + 1$

$U.F \leftarrow TEMPO$



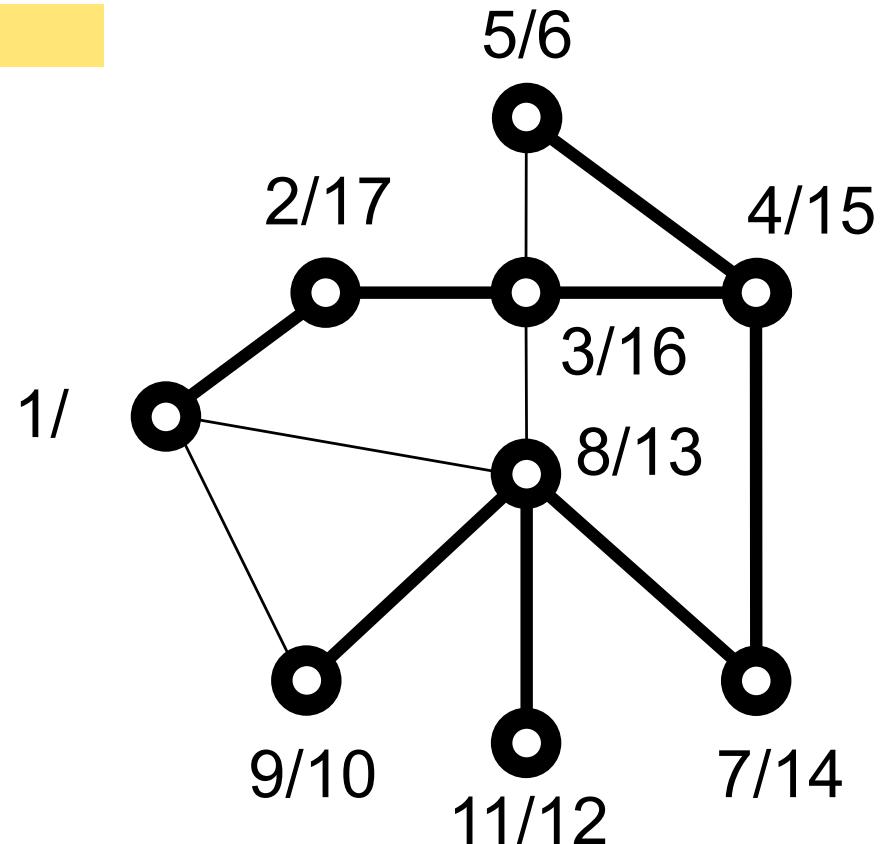
## ALGORITMO DFS\_VISITA( $G, u$ )

```
TEMPO  $\leftarrow$  TEMPO + 1  
 $U.D \leftarrow$  TEMPO  
 $U.COR \leftarrow$  CINZA  
PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$   
SE  $v.COR ==$  BRANCO ENTÃO  
     $v.\pi \leftarrow u$   
    DFS_VISITA( $G, v$ )  
 $U.COR \leftarrow$  PRETO  
TEMPO  $\leftarrow$  TEMPO + 1  
 $U.F \leftarrow$  TEMPO
```



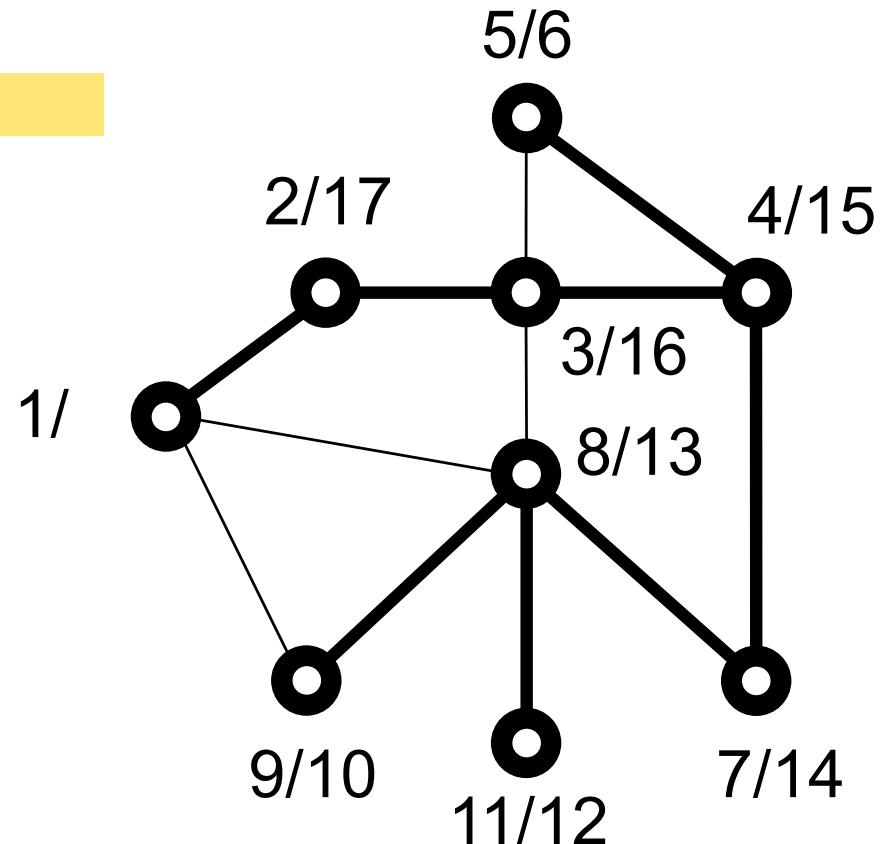
## ALGORITMO DFS\_VISITA( $G, u$ )

```
TEMPO  $\leftarrow$  TEMPO + 1  
 $U.D \leftarrow$  TEMPO  
 $U.COR \leftarrow$  CINZA  
PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$   
SE  $v.COR ==$  BRANCO ENTÃO  
     $v.\pi \leftarrow u$   
    DFS_VISITA( $G, v$ )  
 $U.COR \leftarrow$  PRETO  
TEMPO  $\leftarrow$  TEMPO + 1  
 $U.F \leftarrow$  TEMPO
```



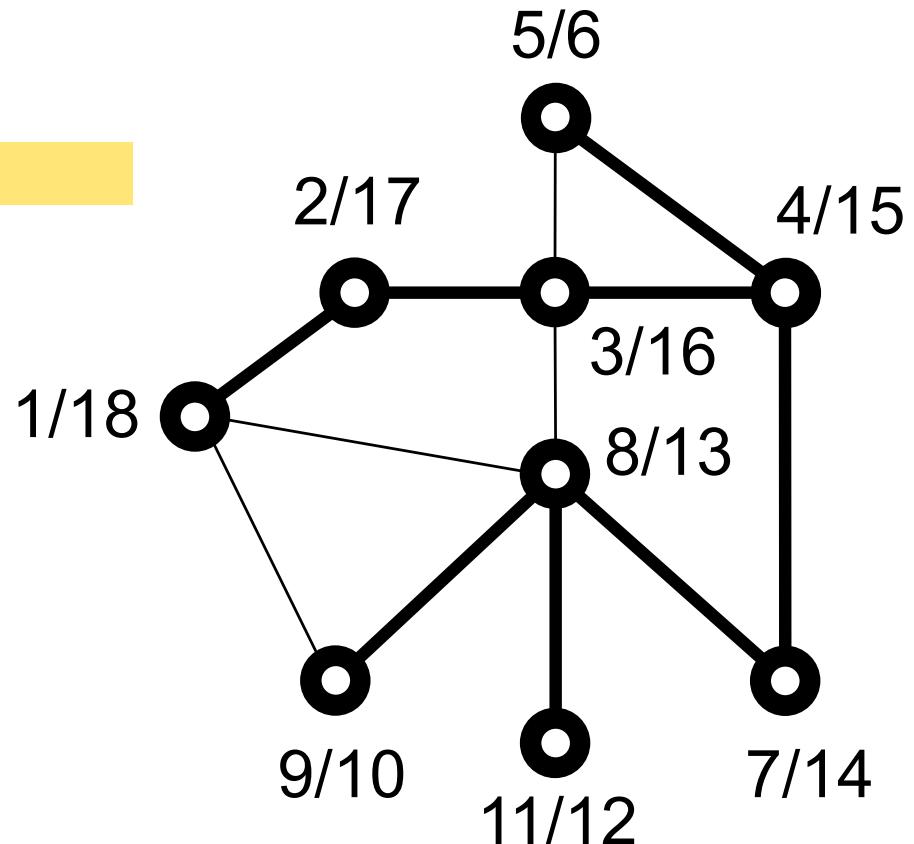
## ALGORITMO DFS\_VISITA( $G, u$ )

```
TEMPO  $\leftarrow$  TEMPO + 1  
 $U.D \leftarrow$  TEMPO  
 $U.COR \leftarrow$  CINZA  
PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$   
    SE  $v.COR ==$  BRANCO ENTÃO  
         $v.\pi \leftarrow u$   
        DFS_VISITA( $G, v$ )  
 $U.COR \leftarrow$  PRETO  
TEMPO  $\leftarrow$  TEMPO + 1  
 $U.F \leftarrow$  TEMPO
```



## ALGORITMO DFS\_VISITA( $G, u$ )

```
TEMPO ← TEMPO + 1  
U.D ← TEMPO  
U.COR ← CINZA  
PARA CADA VÉRTICE  $v \in G.\text{ADJ}[u]$   
    SE  $v.\text{COR} == \text{BRANCO}$  ENTÃO  
         $v.\pi \leftarrow u$   
        DFS_VISITA( $G, v$ )  
    U.COR ← PRETO  
    TEMPO ← TEMPO + 1  
    U.F ← TEMPO
```



## ALGORITMO DFS( $G$ )

PARA CADA VÉRTICE  $u \in G.V$

$u.COR \leftarrow \text{BRANCO}$

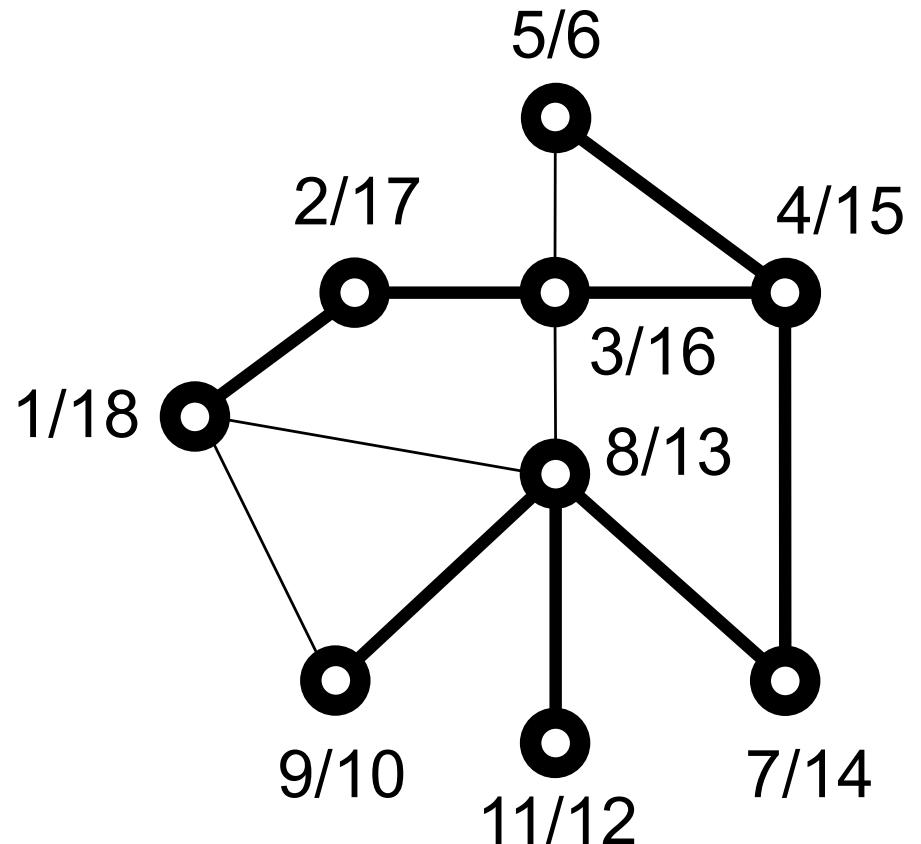
$u.\pi \leftarrow \text{NULO}$

$\text{TEMPO} \leftarrow 0$

PARA CADA VÉRTICE  $u \in G.V$

SE  $u.COR == \text{BRANCO}$  ENTÃO

DFS\_VISITA( $G, u$ )



## ALGORITMO DFS( $G$ )

PARA CADA VÉRTICE  $u \in G.V$

$u.COR \leftarrow \text{BRANCO}$

$u.\pi \leftarrow \text{NULO}$

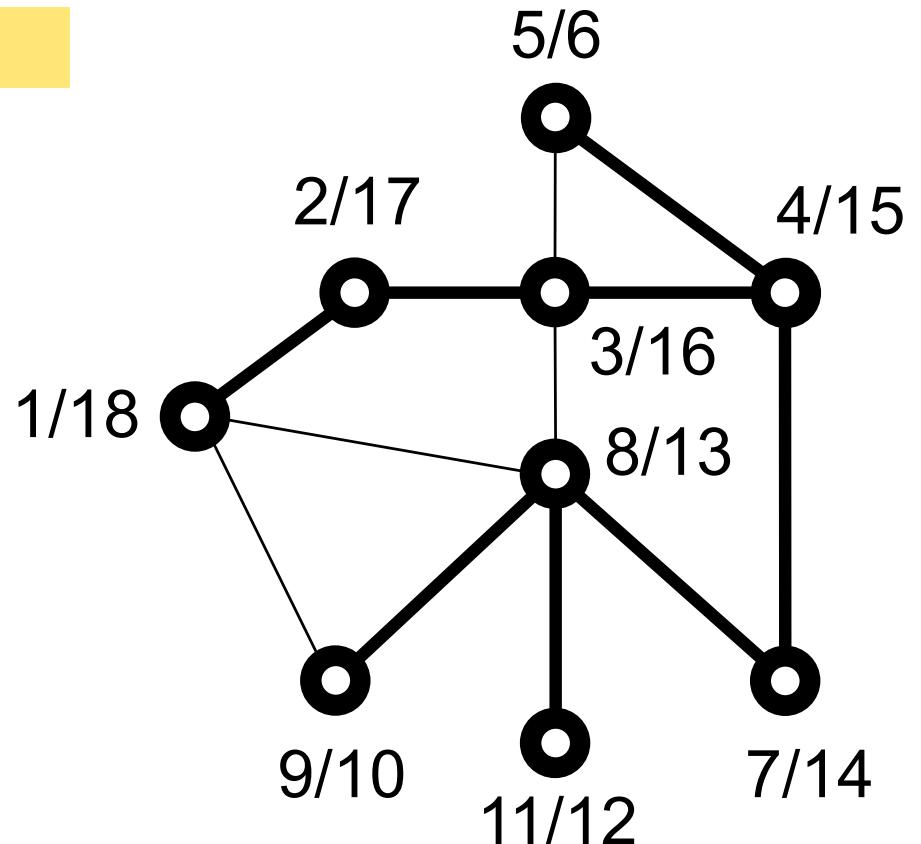
$\text{TEMPO} \leftarrow 0$

PARA CADA VÉRTICE  $u \in G.V$

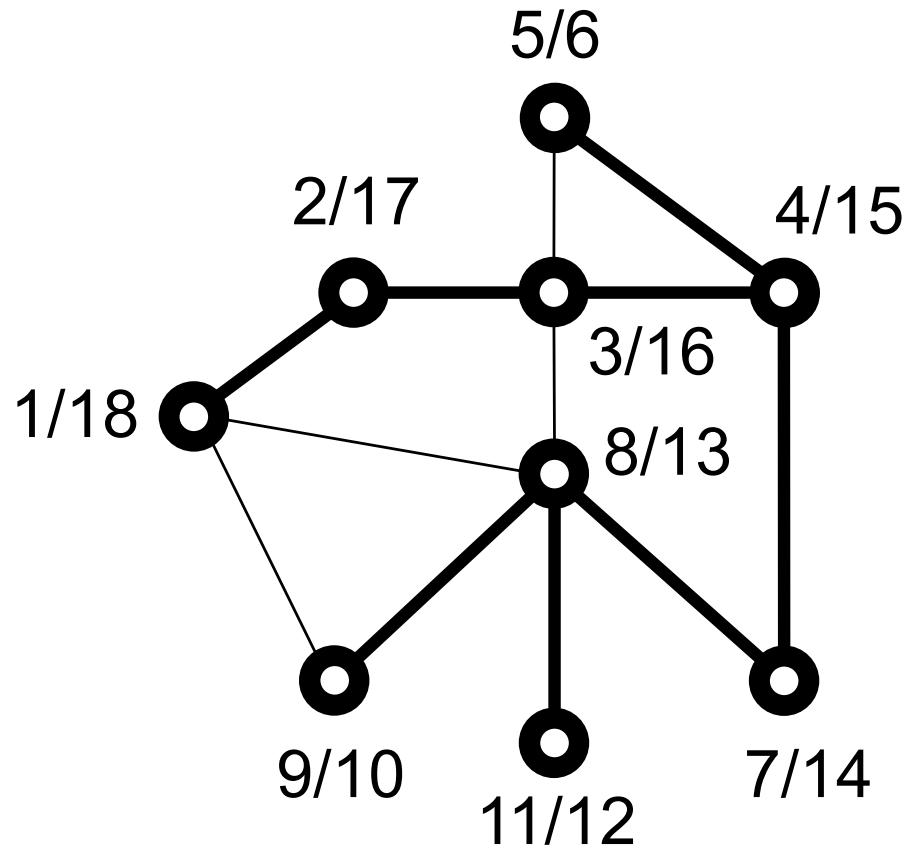
SE  $u.COR == \text{BRANCO}$  ENTÃO

DFS\_VISITA( $G, u$ )

FIM

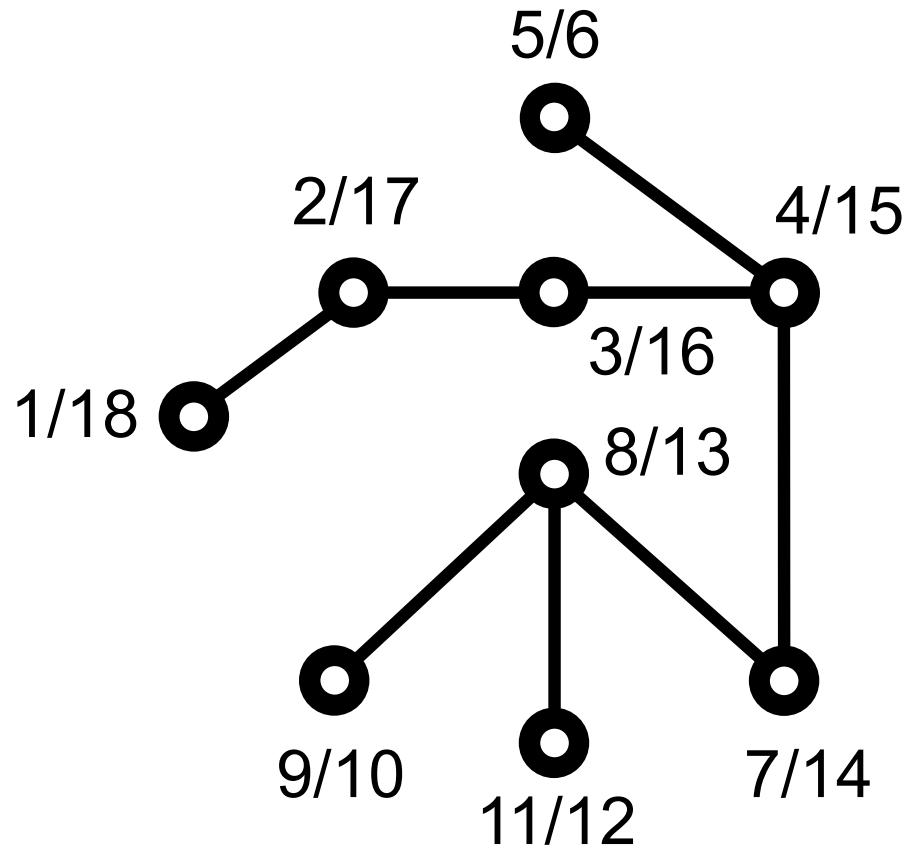


# Busca em profundidade



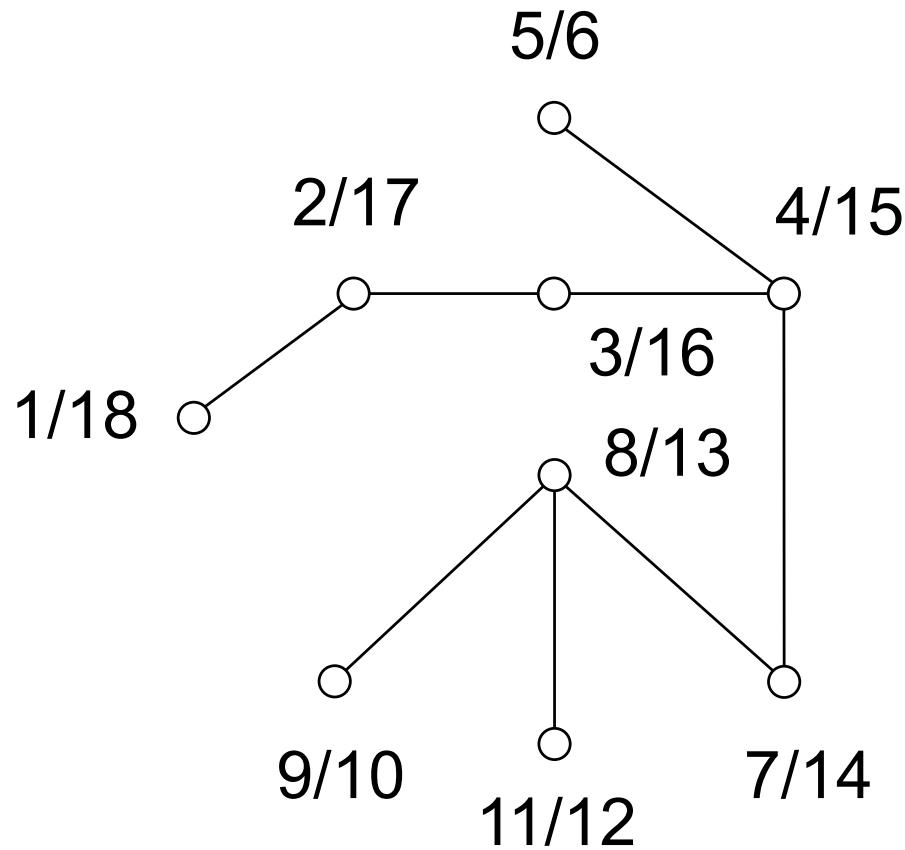
Árvore busca em profundidade

# Busca em profundidade



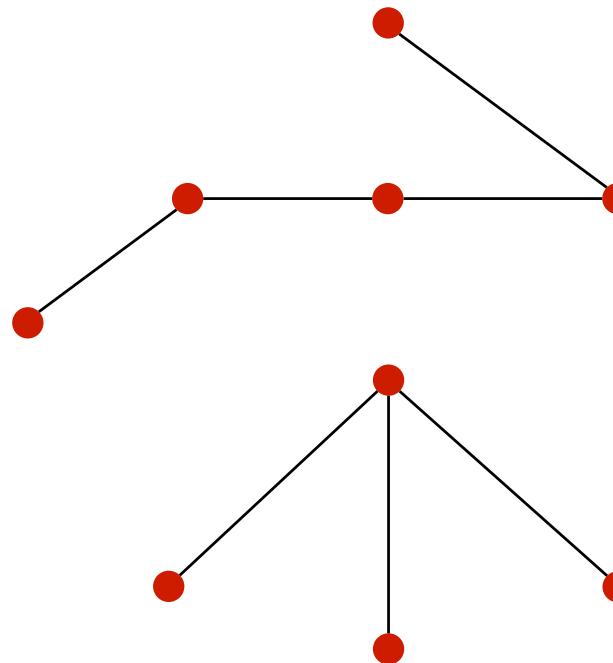
Árvore busca em profundidade

# Busca em profundidade



Árvore busca em profundidade

# Busca em profundidade



Árvore busca em profundidade

# Busca em profundidade

## Eficiência de tempo

Matriz de adjacência:  $\Theta(|V|^2)$ .

Lista de adjacência:  $\Theta(|V| + |E|)$ .

# Busca em profundidade

## Propriedades da DFS

Descobrir uma floresta de árvores de busca em profundidade ( $G_\pi$ ).

Contar componentes conexas de um grafo.

Descobrir a saída de labirintos.

Verificar estruturas de parênteses ( $d$  e  $f$ ).

Pode ser utilizada para **ordenação topológica**.

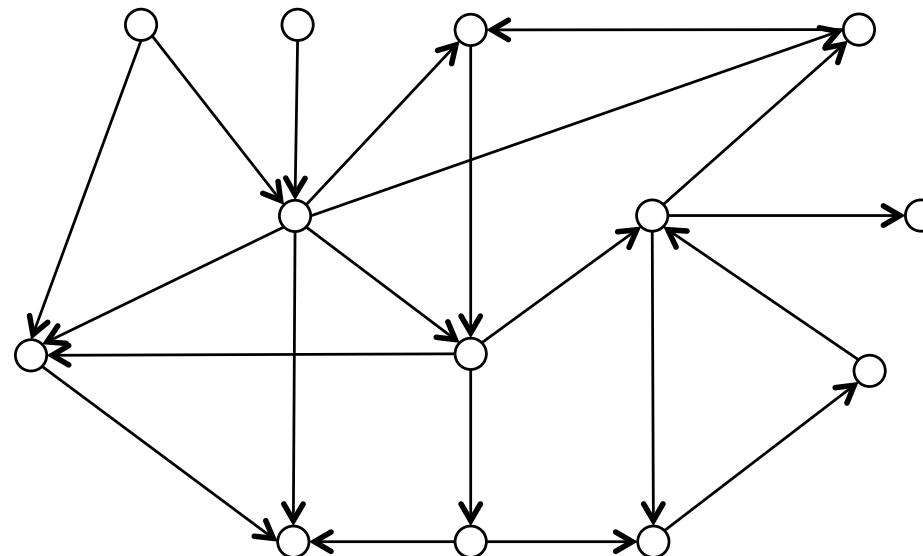
# Ordenação topológica

*(topological sorting)*

# Grafo direcionado/Digrafo

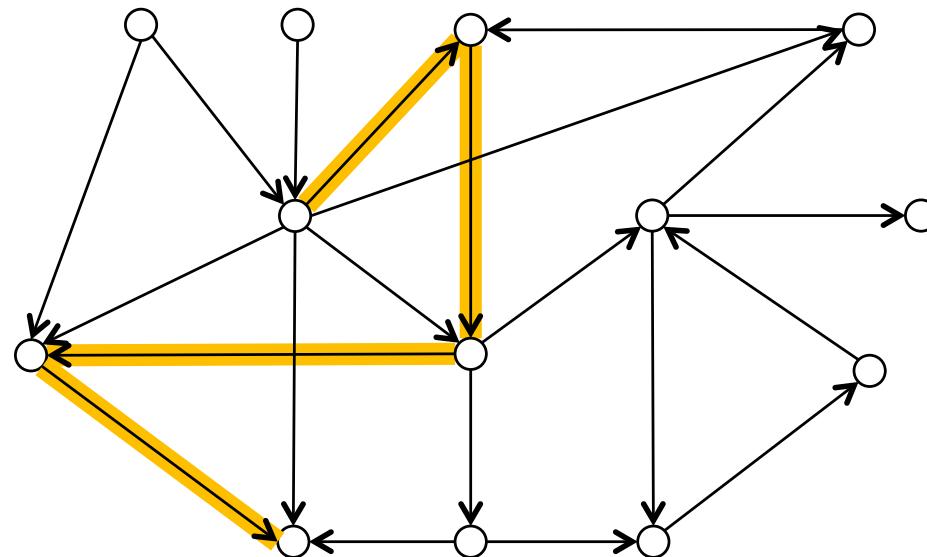
(*directed graph/digraph*)

Grafo em que cada aresta é uma par ordenado de vértices, isto é, cada aresta possui uma “direção” (sentido).



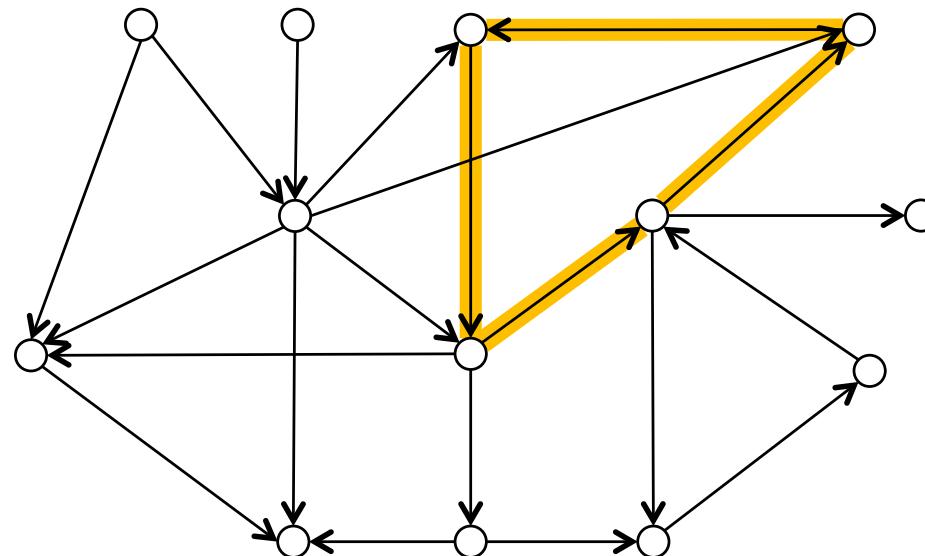
# Caminho (direcionado)

É um grafo em que os vértices são dispostos em sequência em um único sentido.



# Ciclo (direcionado)

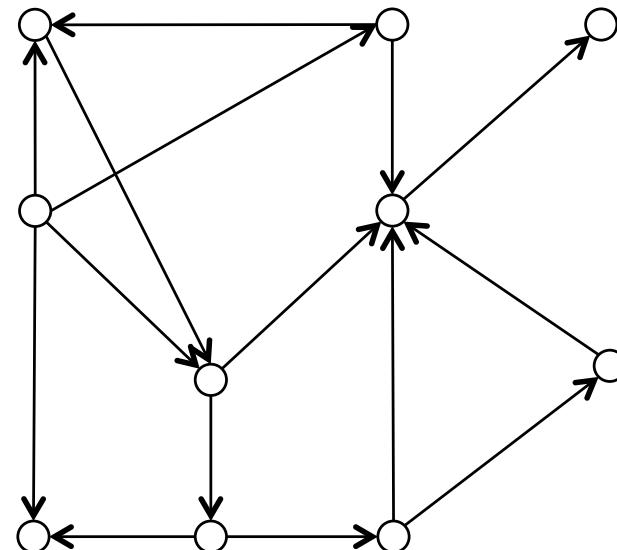
É um caminho cujo primeiro vértice coincide com o último.



# Grafo acíclico direcionado

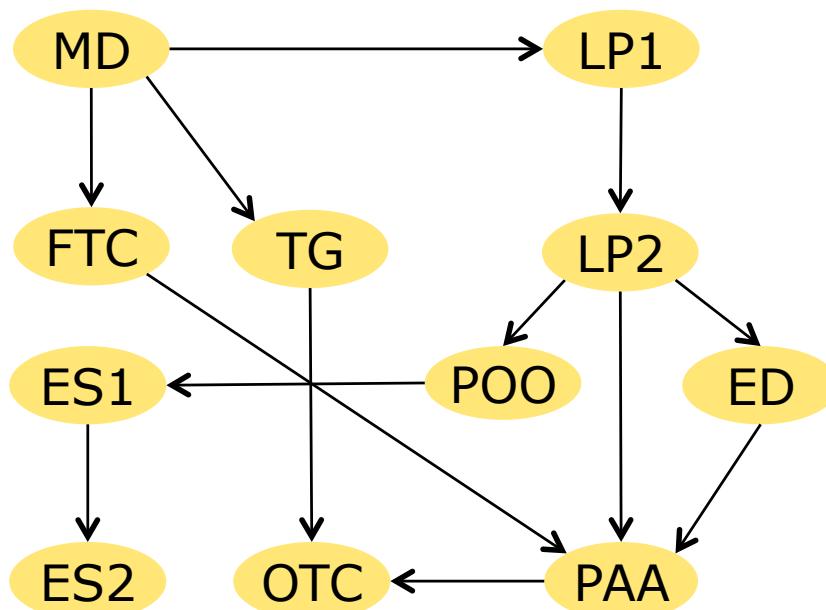
*(directed acyclic graph - DAG)*

É um grafo direcionado que não contém ciclos.



# Grafo acíclico direcionado

Um DAG modela problemas em que objetos mantém alguma relação de precedência entre si (prioridades/ pré-requisitos, etc).



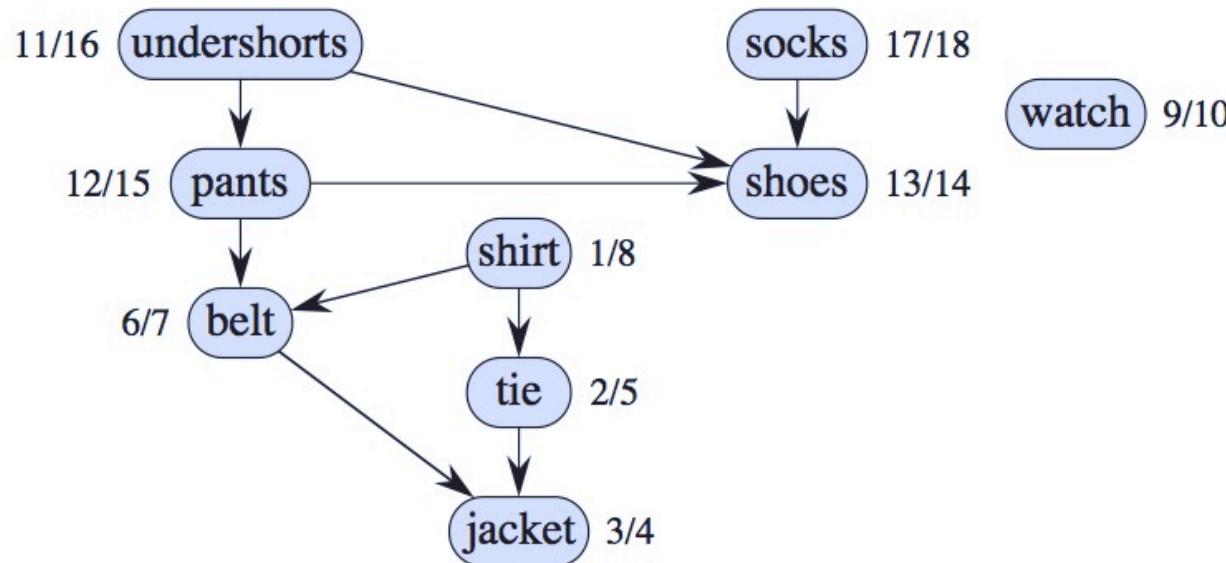
- Planejamento de projetos;
- Controle de versões;
- Disciplinas de um curso;
- Escalonamento de tarefas, etc.

# Ordenação topológica

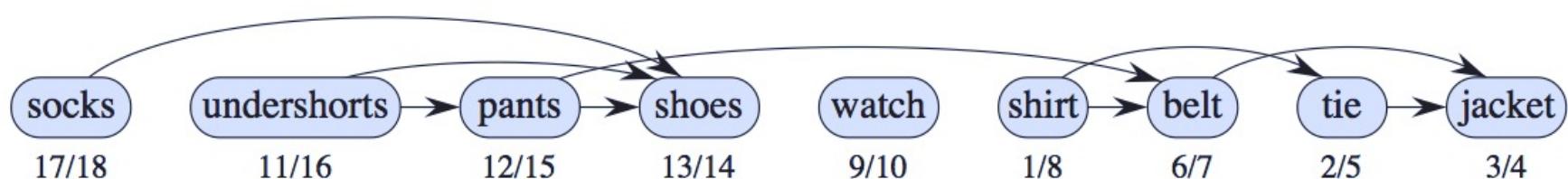
Toma um DAG como entrada e retorna uma lista ordenada dos vértices de forma que todas as arestas direcionadas vão da esquerda para a direita (mantém precedências do DAG).

# Ordenação topológica

Dag



Ordenação topológica



## ALGORITMO ORDENAÇÃO TOPOLOGICA( G )

// ENTRADA: DAG G.

// SAÍDA: LISTA ENCADEADA DE VÉRTICES ORDENADOS  
TOPOLOGICAMENTE.

- 1 CHAME DFS( $G$ ) PARA COMPUTAR OS TEMPOS FINAIS  $v.f$   
PARA CADA VÉRTICE  $v$
- 2 INSIRA CADA VÉRTICE  $v$  EM UMA LISTA ENCADEADA  $L$  DO  
MAIOR  $v.f$  PARA O MENOR
- 3 RETORNE  $L$

## ALGORITMO ORDENAÇÃO TOPOLOGICA( G )

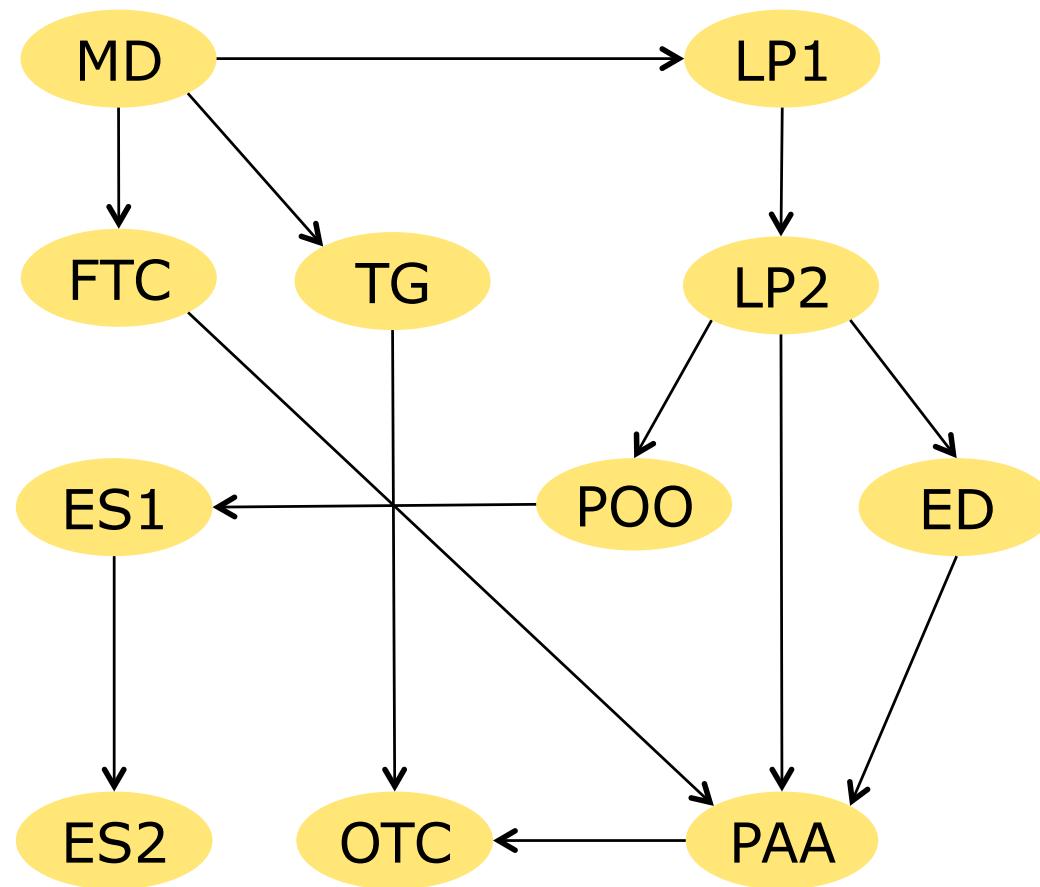
// ENTRADA: DAG G.

// SAÍDA: LISTA ENCADEADA DE VÉRTICES ORDENADOS  
TOPOLOGICAMENTE.

- 1 CHAME DFS( $G$ ) PARA COMPUTAR OS TEMPOS FINAIS  $v.f$   
PARA CADA VÉRTICE  $v$
- 2 CONCLUÍDA A VISITA DE UM VÉRTICE  $v$ , INSIRA-O NA FRENTE  
DE UMA LISTA ENCADEADA  $L$
- 3 RETORNE  $L$

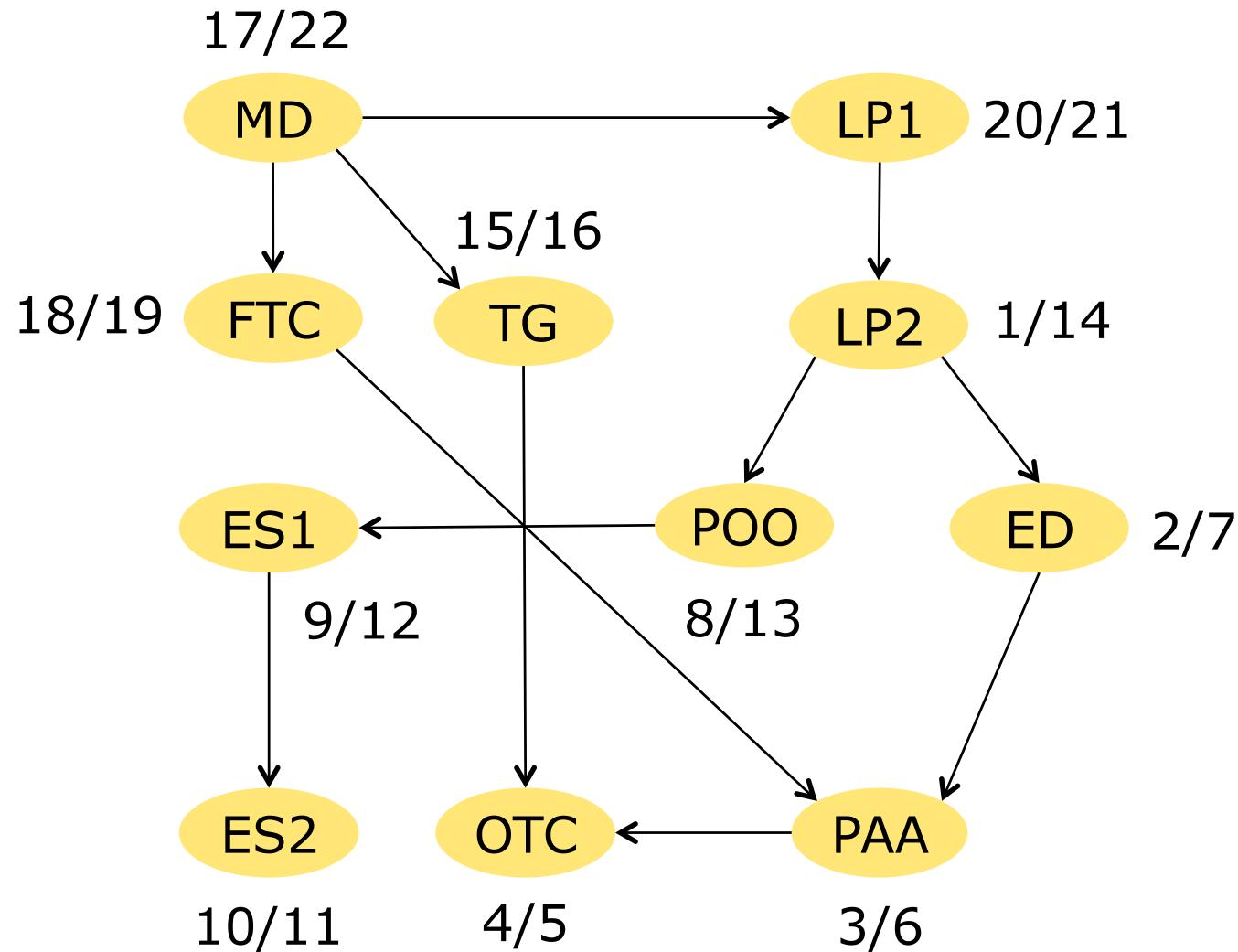
# Ordenação topológica

DAG



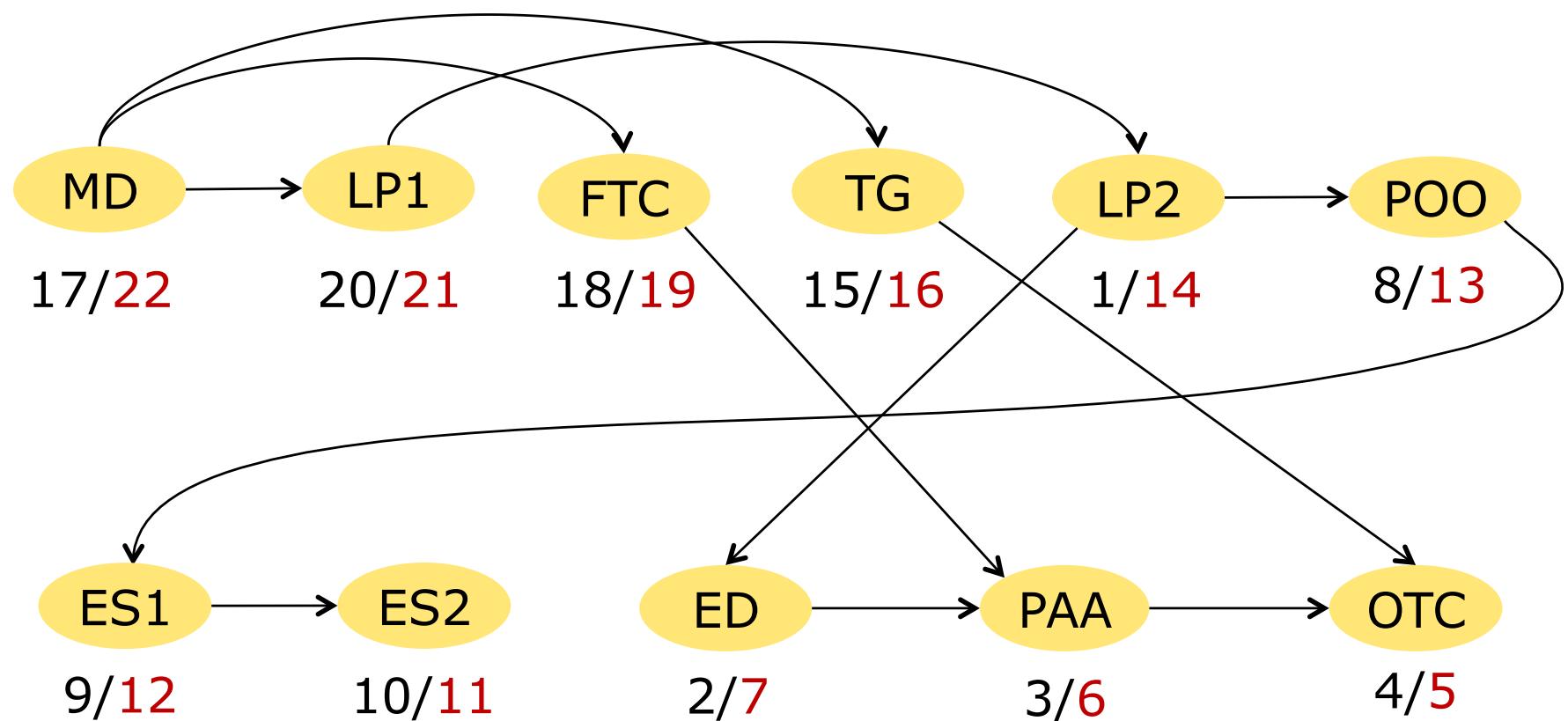
# Ordenação topológica

DFS



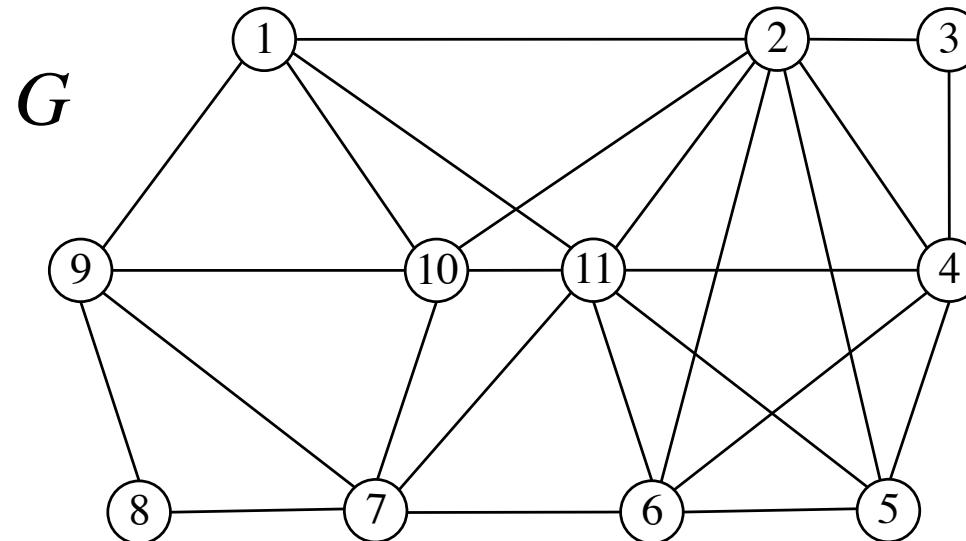
# Ordenação topológica

Ordenação topológica



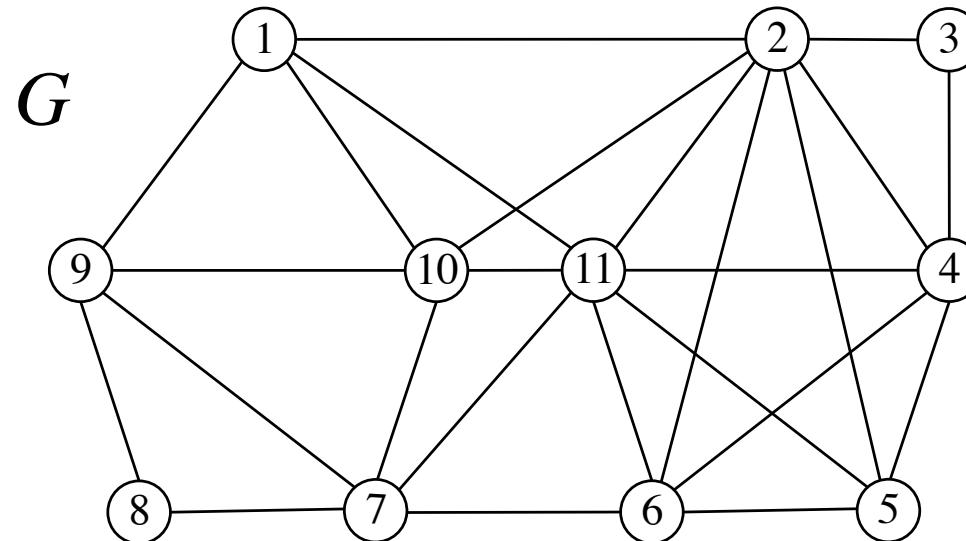
# Exercícios

1. SEJA  $G$  O GRAFO ABAIXO. DETERMINE UMA ÁRVORE EM PROFUNDIDADE EM  $G$  INICIANDO PELO VÉRTICE 5. ESCOLHA SEMPRE O MENOR VÉRTICE ADJACENTE.



# Exercícios

2. DETERMINE OUTRA ÁRVORE EM PROFUNDIDADE EM G INICIANDO PELO VÉRTICE 8. A ÁRVORE OBTIDA É IGUAL À ÁRVORE DA QUESTÃO 1?



# Bibliografia

- BONDY, J.; MURTY, U. ***Graduate Texts in Mathematics series: Graph Theory.*** Springer, 2008. Volume 244.
- DIESTEL, Reinhart. ***Graduate Texts in Mathematics series: Graph Theory.*** New York: Springer-Verlag, 2000. Volume 173.
- CORMEN, T. H. , LEISERSON, C. E., RIVEST, R.L., STEIN, C. ***Introduction to Algorithms***, 3rd edition, MIT Press, 2009.
- SZWARCFITER, J. ***Grafos e Algoritmos Computacionais.*** Rio de Janeiro: 2<sup>a</sup>. Ed. Campus, 1986.
- ZIVIANI, Nivio. ***Projeto de Algoritmos com Implementação em Pascal e C.*** 3a Edição revisada e ampliada de 2010. São Paulo: Thomson Learning, 2010.