

Estruturas de Dados II

Grafos - Conceitos e Busca em Profundidade

Prof^a. Juliana de Santi

Prof. Rodrigo Minetto

Universidade Tecnológica Federal do Paraná

Material compilado de: Cormen, Notas de aula IC-UNICAMP e
IME-USP

Sumário

1 Introdução

2 Busca em Grafos

3 Busca em profundidade

Grafos - Motivação

Motivação: grafos são estruturas abstratas que podem modelar diversos problemas do mundo real. Por exemplo, um grafo pode representar conexões entre cidades por estradas ou uma rede de computadores.

Grafos - Motivação

Motivação: o interesse em estudar algoritmos para problemas em grafos é que conhecer um algoritmo para um determinado problema em grafos pode significar conhecer algoritmos para diversos problemas reais.

Aplicações:

- **Caminho mínimo:** dado um conjunto de cidades, as distâncias entre elas e duas cidades A e B , determinar um caminho (trajeto) mais curto de A até B .
- **Árvore geradora de peso mínimo:** dado um conjunto de computadores, onde cada par de computadores pode ser ligado usando uma quantidade de fibra óptica, encontrar uma rede que interconecte estes computadores que use a menor quantidade possível de fibra óptica.

Grafos - Aplicações

- **Emparelhamento máximo:** dado um conjunto de pessoas e um conjunto de vagas para diferentes empregos, onde cada pessoa é qualificada para certos empregos e cada vaga pode ser ocupada por uma pessoa, encontrar um modo de empregar o maior número possível de pessoas.
- **Problema do caixeiro viajante:** dado um conjunto de cidades, encontrar um passeio que sai de uma cidade, passa por todas as cidades e volta para a cidade inicial tal que a distância total a ser percorrida seja menor possível.

Definição de grafo (*do inglês graphs*):

um **grafo** é um par $G = (V, E)$ onde:

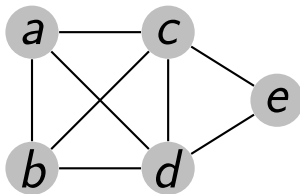
- V é um conjunto finito de elementos chamados **vértices**;
- E é um conjunto finito de pares não-ordenados de vértices chamados **arestas**;

Introdução

Exemplo:

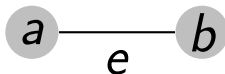
$$V = \{a, b, c, d, e\}$$

$$E = \{(a, b), (a, c), (a, d), (b, c), (b, d), (c, d), (c, e), (d, e)\}$$



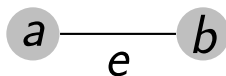
Introdução

Dada uma aresta $e = (a, b)$, dizemos que os vértices a e b são os extremos da aresta e e que a e b são vértices adjacentes.



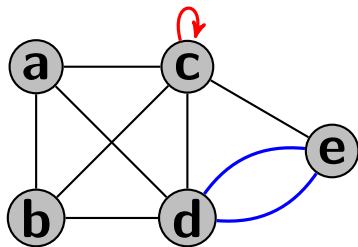
Introdução

Dizemos também que a aresta e é incidente aos vértices a e b , e que os vértices a e b são incidentes à aresta e .



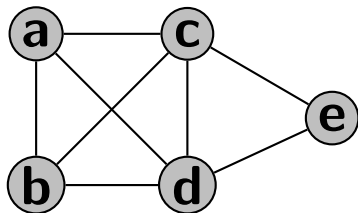
Introdução

Um **laço** é uma aresta com extremos idênticos e **arestas múltiplas** são duas ou mais arestas com o mesmo par de vértices como extremos.



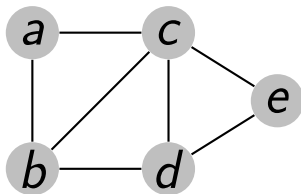
Introdução

Dizemos que um grafo é **simples** quando não possui *laços* ou *arestas múltiplas*.



Introdução

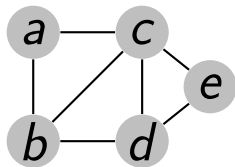
Denotamos por $|V|$ e $|E|$ a cardinalidade dos conjuntos de vértices e arestas de um grafo G , respectivamente. No exemplo abaixo temos $|V| = 5$ e $|E| = 7$.



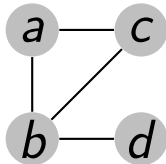
Tamanho do grafo $G = |V| + |E|$.

Introdução

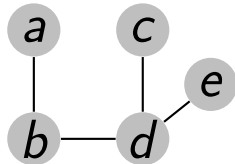
Um **subgrafo** $H = (V', E')$ de um grafo $G = (V, E)$ é um grafo tal que $V' \subset V$ e $E' \subset E$. Um **subgrafo gerador** de G é um subgrafo H com $V' = V$.



Grafo **G**



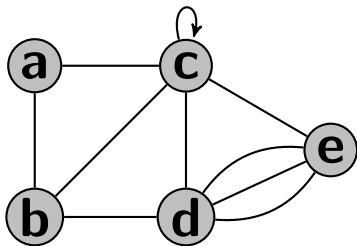
Subgrafo não gerador



Subgrafo gerador

Introdução

O **grau** (*degree*) de um vértice **v**, denotado por **d(v)** é o número de arestas incidentes a **v**, com laços contados duas vezes.
Exemplo:



$$d(a) = 2$$

$$d(b) = 3$$

$$d(c) = 6$$

$$d(d) = 5$$

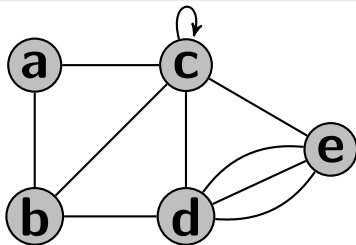
$$d(e) = 4$$

Introdução

Teorema (*Handshaking lemma*)

Para todo grafo $G = (V, E)$ temos:

$$\sum_{v \in V} d(v) = 2|E|$$



$$d(a) = 2$$

$$d(b) = 3$$

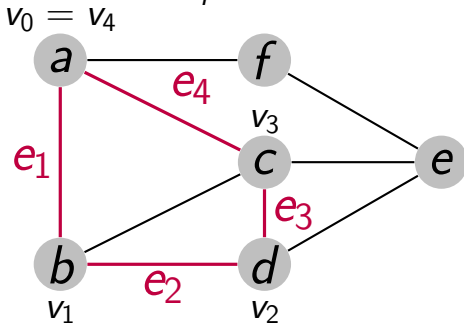
$$d(c) = 6$$

$$d(d) = 5$$

$$d(e) = 4$$

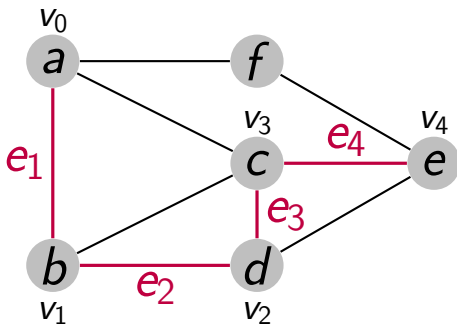
Introdução

Um **caminho** P de v_0 a v_n no grafo G é uma sequência finita e não vazia $(v_0, e_1, v_1, \dots, e_n, v_n)$ cujos elementos são alternadamente vértices e arestas e tal que, para todo $1 \leq i \leq n$, v_{i-1} e v_i são os extremos de e_i .



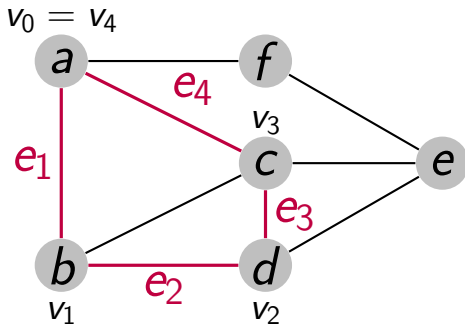
Introdução

Um **caminho simples** é um caminho em que não há repetição de vértices e nem de arestas na sequência. Exemplo:



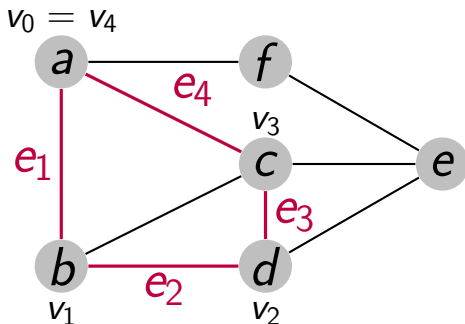
Introdução

Um **ciclo** ou **caminho fechado** é um caminho em que $v_0 = v_n$. Exemplo:



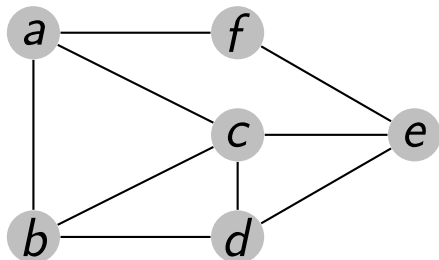
Introdução

O **comprimento** do caminho P é dado pelo seu número de arestas, ou seja, n .



Introdução

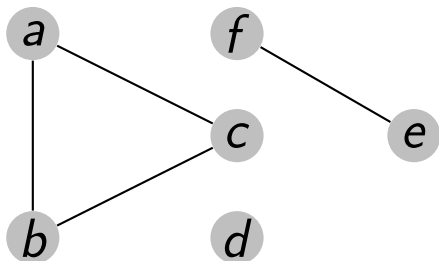
Dizemos que um grafo é **conexo** se, para qualquer par de vértices u e v de \mathbf{G} , existe um caminho de u a v em \mathbf{G} . Exemplo:



Grafo conexo

Introdução

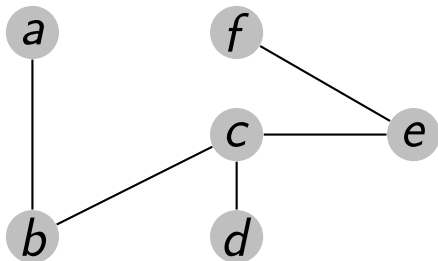
Quando o grafo **G não é conexo**, podemos particionar em **componentes conexos**. Dois vértices u e v de **G** estão no mesmo componente conexo de **G** se há caminho de u a v em **G**. Exemplo:



Grafo não-conexo com 3 componentes conexas

Introdução

Um grafo G é uma **árvore** se é conexo e não possui ciclos (**acíclico**).



Introdução

Se G é uma árvore as seguintes afirmações são equivalentes:

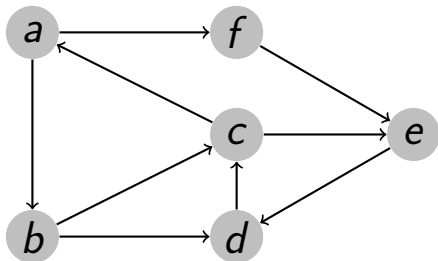
- G é conexo e possui exatamente $|V| - 1$ arestas.
- G é conexo e a remoção de qualquer aresta desconecta o grafo (**minimal** conexo).
- Para todo par de vértices u, v de G , existe um único caminho de u a v em G .

Alguns exemplos de grafos

- **Floresta:** grafo acíclico (não precisa ser conexo). Cada componente é uma árvore!
- **Grafo completo (k_n):** para todo par de vértices u, v a aresta (u, v) pertence ao grafo.
- **Grafo bipartido:** possui uma bipartição (A, B) do conjunto de vértices tal que toda aresta tem um extremo em A e outro em B .
- **Grafo planar:** pode ser desenhado no plano de modo que arestas se interceptam apenas nos extremos.

Introdução

As definições que vimos até agora são para grafos **não orientados**. Um **grafo orientado** é definido de forma semelhante, com a diferença que as arestas (às vezes chamadas de **arcos**) consistem de **pares ordenados de vértices**. Exemplo:



Introdução

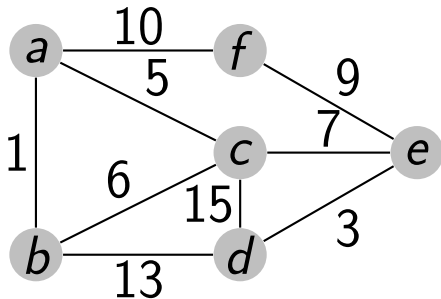
Se $e = (u, v)$ é uma aresta de um grafo orientado G , então dizemos que e sai de u e entra em v . O grau de saída $d^+(v)$ de um vértice v é o número de arestas que saem de v . O grau de entrada $d^-(v)$ de v é o número de arestas que entram em v .

Teorema: para todo grafo orientado $G = (V, E)$ temos:

$$\sum_{v \in V} d^+(v) = \sum_{v \in V} d^-(v) = |E|$$

Introdução

Um grafo (orientado ou não) é **ponderado** se a cada aresta e do grafo está associado um valor real $c(e)$, o qual denominamos **custo** (ou **peso**) da aresta. Exemplo:



Representação interna de grafos

A complexidade dos algoritmos para solução de problemas modelados por grafos depende fortemente da sua representação interna. Existem duas representações canônicas: **matriz de adjacência** e **listas de adjacência**. O uso de uma ou outra num determinado algoritmo depende da natureza das operações que ditam a complexidade do algoritmo.

Matriz de adjacência

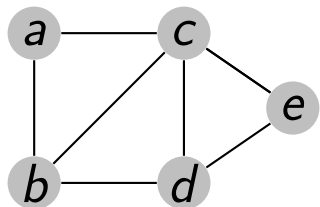
Seja $G = (V, E)$ um grafo simples (orientado ou não). A **matriz de adjacência** de G é uma matriz quadrada A de ordem $|V|$, cujas linhas e colunas são indexadas pelos vértices em V , e tal que:

$$A[i, j] = \begin{cases} 1 & \text{se } (i, j) \in E, \\ 0 & \text{caso contrário} \end{cases}$$

Note que se G é não-orientado, então a matriz A correspondente é simétrica.

Matriz de adjacência

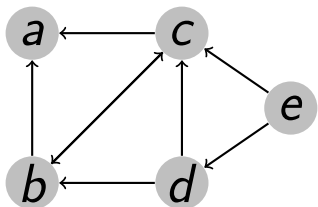
Exemplo de um grafo e a matriz de adjacência correspondente.



| | a | b | c | d | e |
|---|---|---|---|---|---|
| a | 0 | 1 | 1 | 0 | 0 |
| b | 1 | 0 | 1 | 1 | 0 |
| c | 1 | 1 | 0 | 1 | 1 |
| d | 0 | 1 | 1 | 0 | 1 |
| e | 0 | 0 | 1 | 1 | 0 |

Matriz de adjacência

Exemplo de um grafo orientado e a matriz de adjacência correspondente.



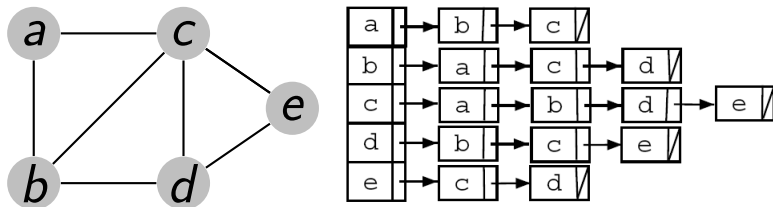
| | a | b | c | d | e |
|---|---|---|---|---|---|
| a | 0 | 0 | 0 | 0 | 0 |
| b | 1 | 0 | 1 | 0 | 0 |
| c | 1 | 1 | 0 | 0 | 0 |
| d | 0 | 1 | 1 | 0 | 0 |
| e | 0 | 0 | 1 | 1 | 0 |

Lista de adjacências

Seja $G = (V, E)$ um grafo simples (orientado ou não). A representação de G por uma **lista de adjacências** consiste no seguinte: para cada vértice v , temos uma lista ligada $Adj[v]$ dos vértices adjacentes a v , ou seja, w aparece em $Adj[v]$ se (v, w) é uma aresta de G . Os vértices podem estar em qualquer ordem em uma lista.

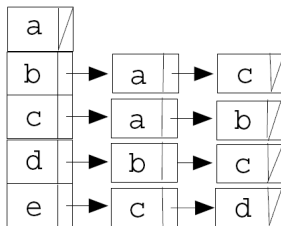
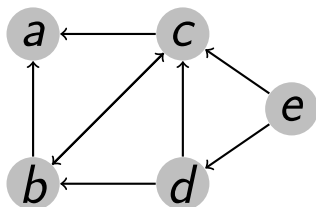
Lista de adjacências

Exemplo de um grafo não-orientado e a lista de adjacências correspondente.



Lista de adjacências

Exemplo de um grafo orientado e a lista de adjacências correspondente.



Matriz de adjacência vs Lista de adjacência

- Matriz de adjacência: é fácil verificar se (i, j) é uma aresta de G .
- Lista de adjacência: é fácil descobrir os vértices adjacentes a um dado vértice v (ou seja, listar $Adj[v]$).
- Matriz de adjacência: espaço $\Theta(|V|^2)$. Geralmente mais adequada a grafos densos ($|E| = \Theta(|V|^2)$).
- Lista de adjacência: espaço $\Theta(|V| + |E|)$. Geralmente mais adequada a grafos esparsos ($|E| = \Theta(|V|)$).

Extensões

Há outras alternativas para representar grafos, mas matrizes e listas de adjacência são as mais usadas. Elas podem ser adaptadas para representar grafos ponderados, grafos com laços e arestas múltiplas, grafos com pesos nos vértices etc.

Sumário

- 1 Introdução
- 2 Busca em Grafos
- 3 Busca em profundidade

Busca em grafos

Grafos são estruturas mais complicadas do que listas, vetores e árvores (binárias). Precisamos de métodos para **explorar/percorrer** um grafo (orientado ou não-orientado).

Busca em grafos

Métodos de buscas em grafos podem ser utilizados para se obter várias **informações** sobre a estrutura do grafo que podem ser úteis para projetar algoritmos eficientes para determinados problemas.

- **Busca em profundidade** (Depth-First-Search)
- **Busca em largura** (Breadth-First-Search)

Notação

Para um grafo G (orientado ou não) podemos denotar por $V[G]$ seu conjunto de vértices e por $E[G]$ seu conjunto de arestas. Para denotar complexidades nas expressões com \mathcal{O} ou Θ usaremos V e E em vez de $|V[G]|$ ou $|E[G]|$. Por exemplo, $\Theta(V + E)$ ou $\mathcal{O}(V^2)$.

Sumário

1 Introdução

2 Busca em Grafos

3 Busca em profundidade

Busca em profundidade

Busca em profundidade (depth first search): a estratégia consiste em pesquisar o grafo o mais “profundamente” sempre que possível. Aplicável tanto a grafos orientados quanto não-orientados.

Busca em profundidade

Possui um número enorme de aplicações:

- determinar os componentes de um grafo;
- ordenação topológica;
- determinar componentes fortemente conexos;
- subrotina para outros algoritmos.

Busca em profundidade

Recebe um grafo $G = (V, E)$ (representado por listas de adjacências). A busca inicia-se em um vértice qualquer. Busca em profundidade é um **método recursivo**. A idéia básica consiste no seguinte:

Busca em profundidade

- Suponha que a busca atingiu um vértice u .
- Escolhe-se um vizinho não visitado v de u para prosseguir a busca.
- “Recursivamente” a busca em profundidade prossegue a partir de v .
- Quando esta busca termina, tenta-se prosseguir a busca a partir de outro vizinho de u . Se não for possível, ela retorna (backtracking) ao nível anterior da recursão.

Busca em profundidade

Outra forma de entender busca em profundidade é **imaginar** que os vértices são armazenados em uma **pilha** à medida que são visitados.

Busca em profundidade

- Suponha que a busca atingiu um vértice u .
- Escolhe-se um vizinho não visitado v de u para prosseguir a busca.
- Empilhe v e repita o passo anterior com v .
- Se nenhum vértice não visitado foi encontrado, então desempilhe um vértice da pilha, digamos u , e volte ao primeiro passo.

Busca em profundidade

A medida que o grafo é percorrido, os vértices visitados vão sendo coloridos. Cada vértice tem uma das seguintes cores:

- Cor branca = “vértice ainda não visitado”.
- Cor **cinza** = “vértice visitado mas ainda não finalizado”.
- Cor **preta** = “vértice visitado e finalizado”.

Busca em profundidade

A busca em profundidade associa a cada vértice x dois rótulos:

- $d[x]$: instante de descoberta de x . Neste instante x torna-se cinza.
- $f[x]$: instante de finalização de x . Neste instante x torna-se preto.

Os rótulos são inteiros entre 1 e $2|V|$.

Busca em profundidade

- Cada vértice v (diferente de s) possui um pai $\pi[v]$.
- O caminho de v a s é dado por:
 $v, \pi[v], \pi[\pi[v]], \pi[\pi[\pi[v]]], \dots, s$.

Busca em profundidade

BUSCA-EM-PROFUNDIDADE (G)

Recebe um grafo G (na forma de listas de adjacências) e devolve: (i) os instantes $d[v]$, $f[v]$ para cada $v \in V$ e (ii) uma floresta de busca em profundidade.

BUSCA-EM-PROFUNDIDADE (G)

▷ Para cada $u \in V[G]$ faça

$\text{cor}[u] \leftarrow \text{branco};$

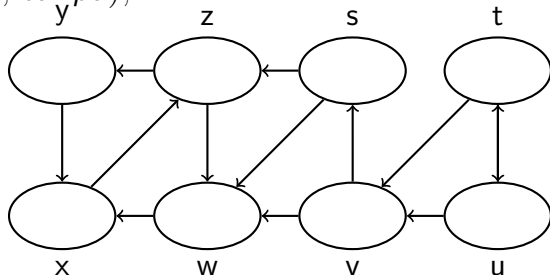
$\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0$;

Para cada $u \in V[G]$ faça

 Se $\text{cor}[u] = \text{branco}$ então

 DFS-VISIT(G, V, u, tempo);



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ faça

▷ $cor[u] \leftarrow \text{branco};$

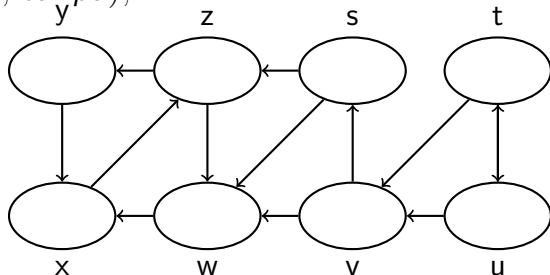
$\pi[u] \leftarrow \text{NIL};$

$tempo \leftarrow 0;$

Para cada $u \in V[G]$ faça

Se $cor[u] = \text{branco}$ então

$\text{DFS-VISIT}(G, V, u, tempo);$



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ faça

$\text{cor}[u] \leftarrow \text{branco};$

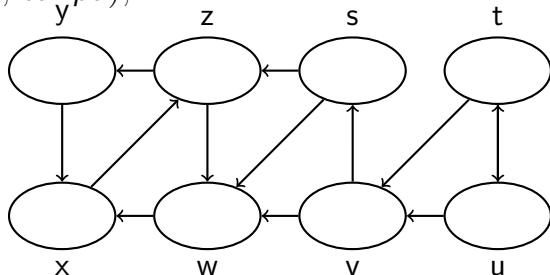
▷ $\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0;$

Para cada $u \in V[G]$ faça

Se $\text{cor}[u] = \text{branco}$ **então**

 DFS-VISIT(G, V, u, tempo);



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ faça

$\text{cor}[u] \leftarrow \text{branco};$

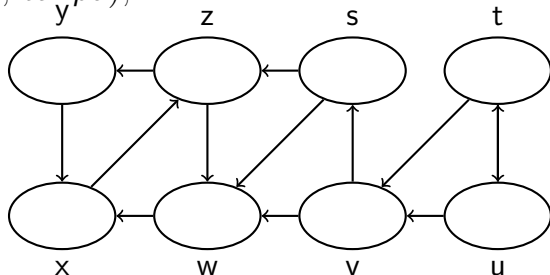
$\pi[u] \leftarrow \text{NIL};$

$\triangleright \text{tempo} \leftarrow 0;$

Para cada $u \in V[G]$ faça

Se $\text{cor}[u] = \text{branco}$ **então**

 DFS-VISIT(G, V, u, tempo);



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ faça

$\text{cor}[u] \leftarrow \text{branco};$

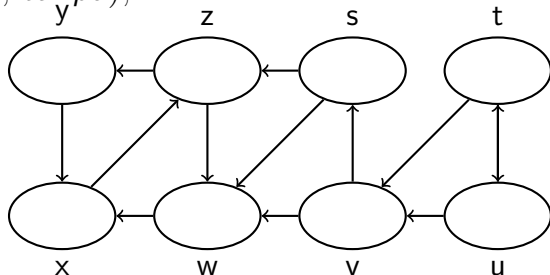
$\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0;$

▷ **Para cada $u \in V[G]$ faça**

Se $\text{cor}[u] = \text{branco}$ então

 DFS-VISIT(G, V, u, tempo);



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ faça

$\text{cor}[u] \leftarrow \text{branco};$

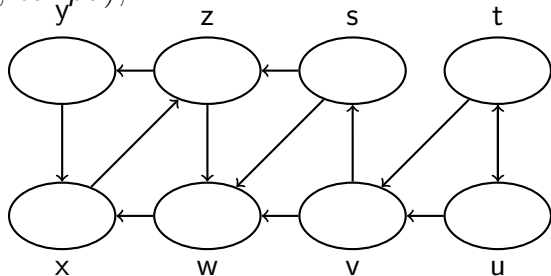
$\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0$;

Para cada $u \in V[G]$ faça

Se $\text{cor}[s] = \text{branco}$ então

 DFS-VISIT(G, V, s, tempo);



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ **faça**

$\text{cor}[u] \leftarrow \text{branco};$

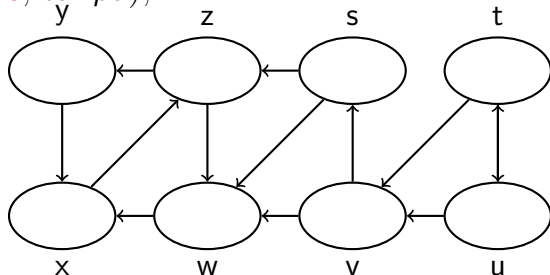
$\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0$;

Para cada $u \in V[G]$ **faça**

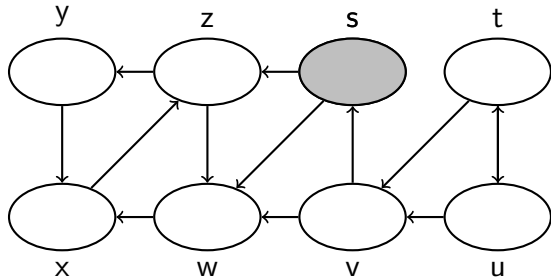
Se $\text{cor}[s] = \text{branco}$ **então**

$\triangleright \text{DFS-VISIT}(G, V, s, \text{tempo});$



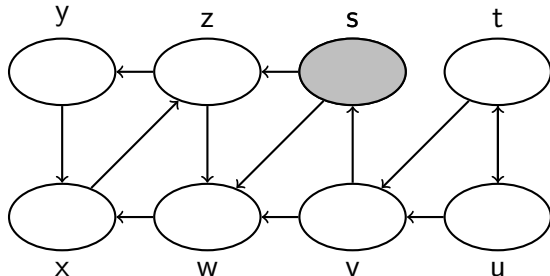
DFS-VISIT ($G, V, s, 0$)

```
▷ cor[s] ← cinza;  
  tempo ← ++tempo;  
d[s] ← tempo;  
Para cada  $z \in Adj[s]$  faça  
  Se cor[z] = branco então  
     $\pi[z] \leftarrow s$ ;  
    DFS-VISIT ( $G, V, z, 1$ );  
cor[s] ← preto;  
f[s] ← ++tempo;
```



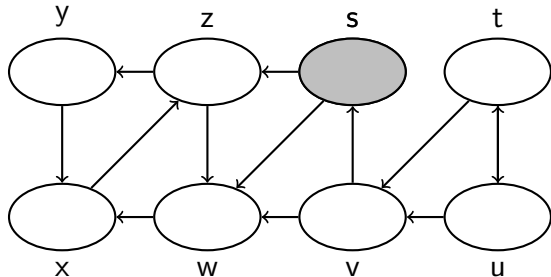
DFS-VISIT ($G, V, s, 0$)

```
cor[s] ← cinza;  
▷ tempo ← ++tempo;  
d[s] ← tempo;  
Para cada  $z \in Adj[s]$  faça  
    Se cor[z] = branco então  
         $\pi[z] \leftarrow s$ ;  
        DFS-VISIT ( $G, V, z, 1$ );  
cor[s] ← preto;  
f[s] ← ++tempo;
```



DFS-VISIT ($G, V, s, 0$)

```
cor[s] ← cinza;  
▷ tempo ← 1;  
d[s] ← tempo;  
Para cada  $z \in Adj[s]$  faça  
  Se cor[z] = branco então  
     $\pi[z] \leftarrow s$ ;  
    DFS-VISIT ( $G, V, z, 1$ );  
cor[s] ← preto;  
f[s] ← ++tempo;
```



DFS-VISIT ($G, V, s, 0$)

$cor[s] \leftarrow \text{cinza};$

$tempo \leftarrow 1;$

$\triangleright d[s] \leftarrow tempo;$

Para cada $z \in Adj[s]$ **faça**

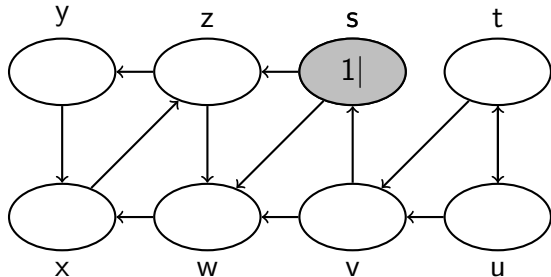
Se $cor[z] = \text{branco}$ **então**

$\pi[z] \leftarrow s;$

DFS-VISIT ($G, V, z, 1$);

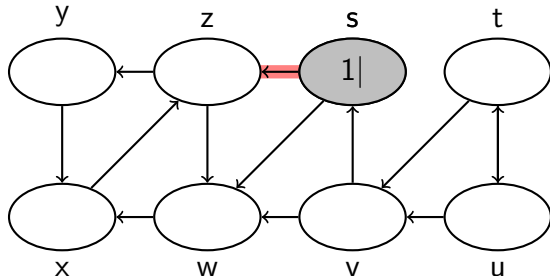
$cor[s] \leftarrow \text{preto};$

$f[s] \leftarrow ++tempo;$



DFS-VISIT ($G, V, s, 0$)

```
cor[s] ← cinza;  
tempo ← 1;  
d[s] ← tempo;  
▷ Para cada  $z \in Adj[s]$  faça  
  Se cor[z] = branco então  
     $\pi[z] \leftarrow s$ ;  
    DFS-VISIT ( $G, V, z, 1$ );  
cor[s] ← preto;  
f[s] ← ++tempo;
```



DFS-VISIT ($G, V, s, 0$)

$cor[s] \leftarrow \text{cinza};$

$tempo \leftarrow 1;$

$d[s] \leftarrow tempo;$

Para cada $z \in Adj[s]$ **faça**

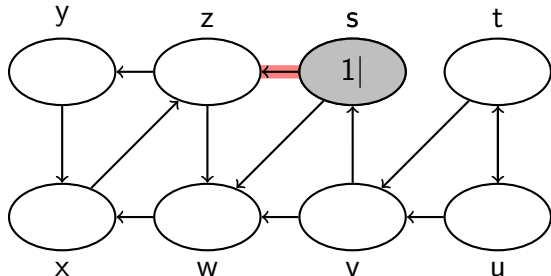
▷ **Se** $cor[z] = \text{branco}$ **então**

$\pi[z] \leftarrow s;$

DFS-VISIT ($G, V, z, 1$);

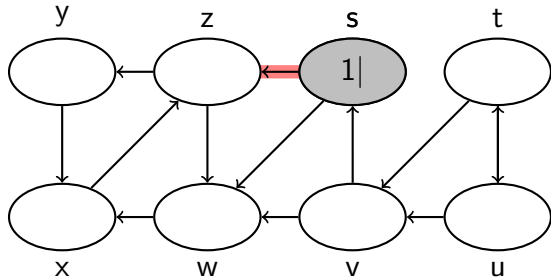
$cor[s] \leftarrow \text{preto};$

$f[s] \leftarrow ++tempo;$



DFS-VISIT ($G, V, s, 0$)

```
cor[s] ← cinza;  
tempo ← 1;  
d[s] ← tempo;  
Para cada  $z \in Adj[s]$  faça  
    Se cor[z] = branco então  
        ▷  $\pi[z] \leftarrow s$ ;  
        DFS-VISIT ( $G, V, z, 1$ );  
cor[s] ← preto;  
f[s] ← ++tempo;
```



DFS-VISIT ($G, V, s, 0$)

$cor[s] \leftarrow \text{cinza};$

$tempo \leftarrow 1;$

$d[s] \leftarrow tempo;$

Para cada $z \in Adj[s]$ **faça**

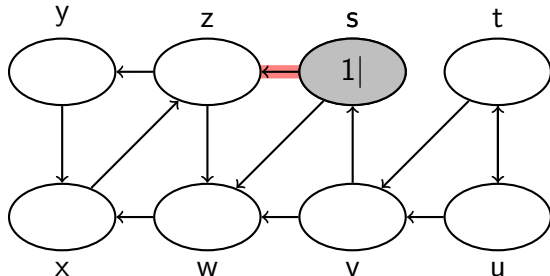
Se $cor[z] = \text{branco}$ **então**

$\pi[z] \leftarrow s;$

▷ DFS-VISIT ($G, V, z, 1$);

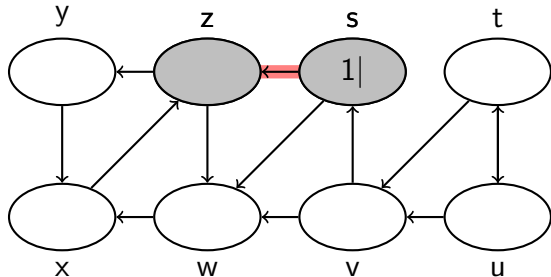
$cor[s] \leftarrow \text{preto};$

$f[s] \leftarrow ++tempo;$



DFS-VISIT ($G, V, z, 1$)

```
▷ cor[z] ← cinza;  
tempo ← ++tempo;  
d[z] ← tempo;  
Para cada  $y \in Adj[z]$  faça  
  Se cor[y] = branco então  
     $\pi[y] \leftarrow z$ ;  
    DFS-VISIT ( $G, V, y, tempo$ );  
cor[z] ← preto;  
f[z] ← ++tempo;
```



DFS-VISIT ($G, V, z, 1$)

$\text{cor}[z] \leftarrow \text{cinza};$

$\triangleright \text{tempo} \leftarrow ++\text{tempo};$

$d[z] \leftarrow \text{tempo};$

Para cada $y \in \text{Adj}[z]$ **faça**

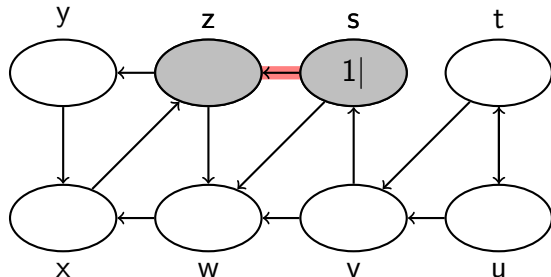
Se $\text{cor}[y] = \text{branco}$ **então**

$\pi[y] \leftarrow z;$

DFS-VISIT (G, V, y, tempo);

$\text{cor}[z] \leftarrow \text{preto};$

$f[z] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, z, 1$)

$\text{cor}[z] \leftarrow \text{cinza};$

$\triangleright \text{tempo} \leftarrow 2;$

$d[z] \leftarrow \text{tempo};$

Para cada $y \in \text{Adj}[z]$ **faça**

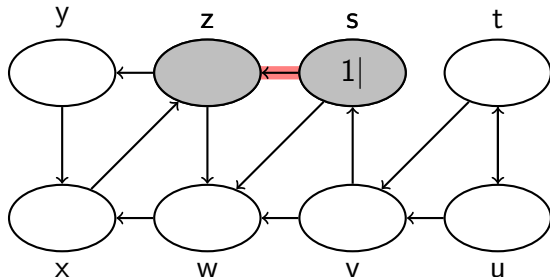
Se $\text{cor}[y] = \text{branco}$ **então**

$\pi[y] \leftarrow z;$

 DFS-VISIT (G, V, y, tempo);

$\text{cor}[z] \leftarrow \text{preto};$

$f[z] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, z, 1$)

$\text{cor}[z] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 2;$

$\triangleright \text{d}[z] \leftarrow \text{tempo};$

Para cada $y \in \text{Adj}[z]$ **faça**

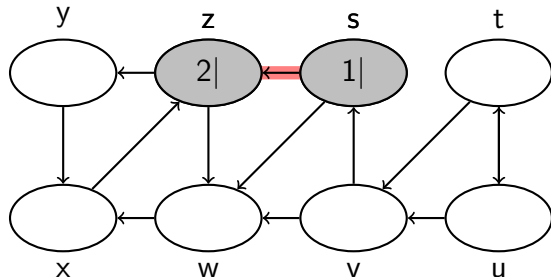
Se $\text{cor}[y] = \text{branco}$ **então**

$\pi[y] \leftarrow z;$

DFS-VISIT (G, V, y, tempo);

$\text{cor}[z] \leftarrow \text{preto};$

$f[z] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, z, 1$)

$\text{cor}[z] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 2;$

$d[z] \leftarrow \text{tempo};$

▷ **Para cada** $y \in \text{Adj}[z]$ **faça**

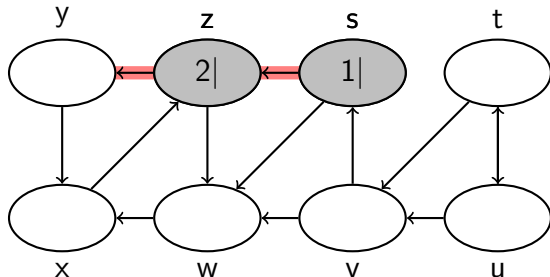
Se $\text{cor}[y] = \text{branco}$ **então**

$\pi[y] \leftarrow z;$

 DFS-VISIT (G, V, y, tempo);

$\text{cor}[z] \leftarrow \text{preto};$

$f[z] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, z, 1$)

$\text{cor}[z] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 2;$

$d[z] \leftarrow \text{tempo};$

Para cada $y \in \text{Adj}[z]$ **faça**

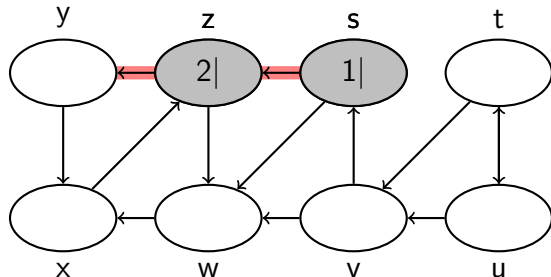
▷ **Se** $\text{cor}[y] = \text{branco}$ **então**

$\pi[y] \leftarrow z;$

DFS-VISIT (G, V, y, tempo);

$\text{cor}[z] \leftarrow \text{preto};$

$f[z] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, z, 1$)

$\text{cor}[z] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 2;$

$d[z] \leftarrow \text{tempo};$

Para cada $y \in \text{Adj}[z]$ **faça**

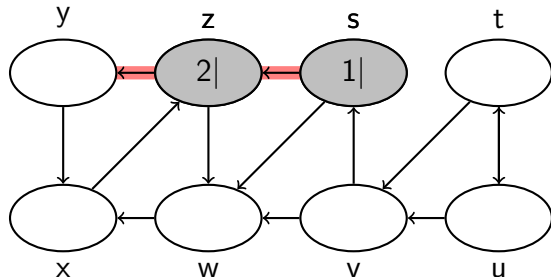
Se $\text{cor}[y] = \text{branco}$ **então**

▷ $\pi[y] \leftarrow z;$

DFS-VISIT (G, V, y, tempo);

$\text{cor}[z] \leftarrow \text{preto};$

$f[z] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, z, 1$)

$\text{cor}[z] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 2;$

$d[z] \leftarrow \text{tempo};$

Para cada $y \in \text{Adj}[z]$ **faça**

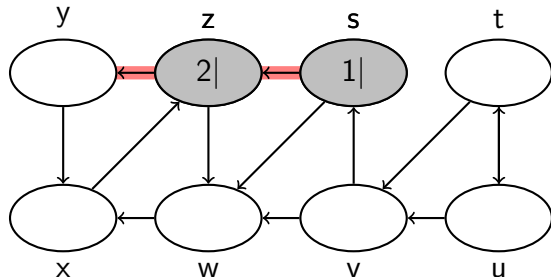
Se $\text{cor}[y] = \text{branco}$ **então**

$\pi[y] \leftarrow z;$

▷ DFS-VISIT (G, V, y, tempo);

$\text{cor}[z] \leftarrow \text{preto};$

$f[z] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, z, 1$)

$\text{cor}[z] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 2;$

$d[z] \leftarrow \text{tempo};$

Para cada $y \in \text{Adj}[z]$ **faça**

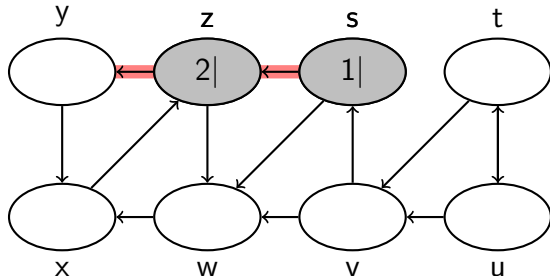
Se $\text{cor}[y] = \text{branco}$ **então**

$\pi[y] \leftarrow z;$

▷ DFS-VISIT ($G, V, y, 2$);

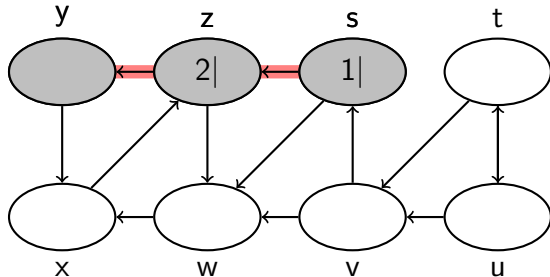
$\text{cor}[z] \leftarrow \text{preto};$

$f[z] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, y, 2$)

```
▷ cor[y] ← cinza;  
tempo ← ++tempo;  
d[y] ← tempo;  
Para cada  $x \in Adj[y]$  faça  
  Se cor[x] = branco então  
     $\pi[x] \leftarrow y$ ;  
    DFS-VISIT ( $G, V, x, tempo$ );  
cor[y] ← preto;  
f[y] ← ++tempo;
```



DFS-VISIT ($G, V, y, 2$)

$\text{cor}[y] \leftarrow \text{cinza};$

$\triangleright \text{tempo} \leftarrow ++\text{tempo};$

$d[y] \leftarrow \text{tempo};$

Para cada $x \in \text{Adj}[y]$ **faça**

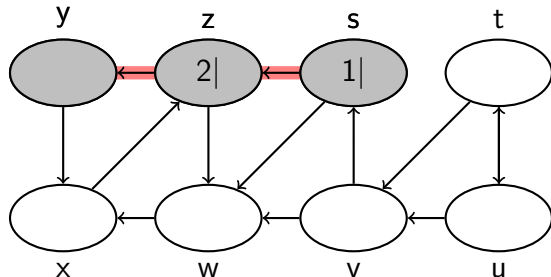
Se $\text{cor}[x] = \text{branco}$ **então**

$\pi[x] \leftarrow y;$

DFS-VISIT (G, V, x, tempo);

$\text{cor}[y] \leftarrow \text{preto};$

$f[y] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, y, 2$)

$\text{cor}[y] \leftarrow \text{cinza};$

$\triangleright \text{tempo} \leftarrow 3;$

$d[y] \leftarrow \text{tempo};$

Para cada $x \in \text{Adj}[y]$ **faça**

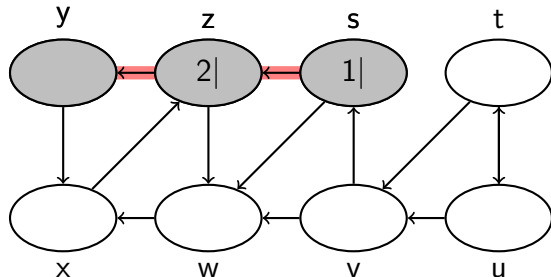
Se $\text{cor}[x] = \text{branco}$ **então**

$\pi[x] \leftarrow y;$

DFS-VISIT (G, V, x, tempo);

$\text{cor}[y] \leftarrow \text{preto};$

$f[y] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, y, 2$)

$\text{cor}[y] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 3;$

$\triangleright d[y] \leftarrow \text{tempo};$

Para cada $x \in \text{Adj}[y]$ **faça**

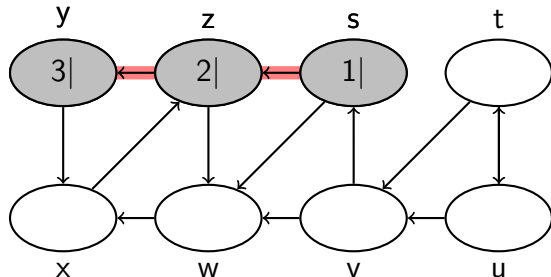
Se $\text{cor}[x] = \text{branco}$ **então**

$\pi[x] \leftarrow y;$

DFS-VISIT (G, V, x, tempo);

$\text{cor}[y] \leftarrow \text{preto};$

$f[y] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, y, 2$)

$\text{cor}[y] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 3;$

$d[y] \leftarrow \text{tempo};$

▷ **Para cada** $x \in \text{Adj}[y]$ **faça**

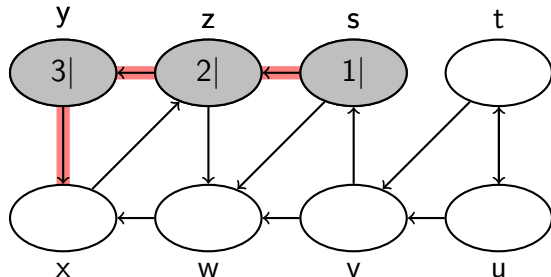
Se $\text{cor}[x] = \text{branco}$ **então**

$\pi[x] \leftarrow y;$

 DFS-VISIT ($G, V, x, \text{tempo} + 1$);

$\text{cor}[y] \leftarrow \text{preto};$

$f[y] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, y, 2$)

$\text{cor}[y] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 3;$

$d[y] \leftarrow \text{tempo};$

Para cada $x \in \text{Adj}[y]$ **faça**

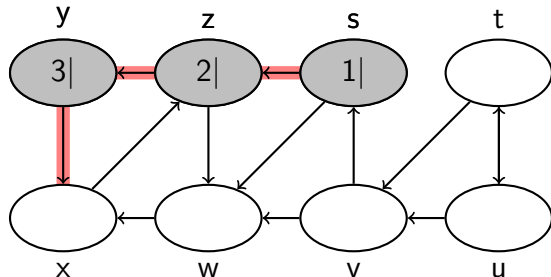
▷ **Se** $\text{cor}[x] = \text{branco}$ **então**

$\pi[x] \leftarrow y;$

DFS-VISIT ($G, V, x, \text{tempo} + 1$);

$\text{cor}[y] \leftarrow \text{preto};$

$f[y] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, y, 2$)

$\text{cor}[y] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 3;$

$d[y] \leftarrow \text{tempo};$

Para cada $x \in \text{Adj}[y]$ **faça**

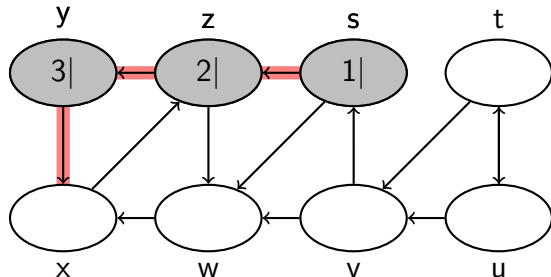
Se $\text{cor}[x] = \text{branco}$ **então**

▷ $\pi[x] \leftarrow y;$

DFS-VISIT (G, V, x, tempo);

$\text{cor}[y] \leftarrow \text{preto};$

$f[y] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, y, 2$)

$\text{cor}[y] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 3;$

$d[y] \leftarrow \text{tempo};$

Para cada $x \in \text{Adj}[y]$ **faça**

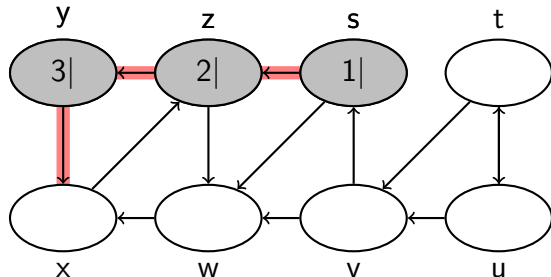
Se $\text{cor}[x] = \text{branco}$ **então**

$\pi[x] \leftarrow y;$

▷ DFS-VISIT (G, V, x, tempo);

$\text{cor}[y] \leftarrow \text{preto};$

$f[y] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, y, 2$)

$cor[y] \leftarrow \text{cinza};$

$tempo \leftarrow 3;$

$d[y] \leftarrow tempo;$

Para cada $x \in Adj[y]$ **faça**

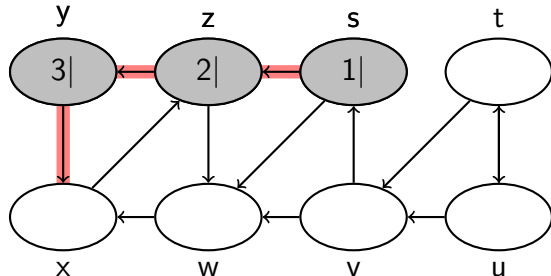
Se $cor[x] = \text{branco}$ **então**

$\pi[x] \leftarrow y;$

▷ DFS-VISIT ($G, V, x, 3$);

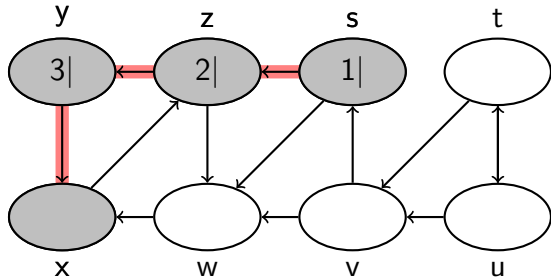
$cor[y] \leftarrow \text{preto};$

$f[y] \leftarrow ++tempo;$



DFS-VISIT ($G, V, x, 3$)

```
▷ cor[x] ← cinza;  
tempo ← ++tempo;  
d[x] ← tempo;  
Para cada  $z \in Adj[x]$  faça  
  Se cor[z] = branco então  
     $\pi[z] \leftarrow x$ ;  
    DFS-VISIT ( $G, V, z, tempo$ );  
cor[x] ← preto;  
f[x] ← ++tempo;
```



DFS-VISIT ($G, V, x, 3$)

$\text{cor}[x] \leftarrow \text{cinza};$

$\triangleright \text{tempo} \leftarrow ++\text{tempo};$

$d[x] \leftarrow \text{tempo};$

Para cada $z \in \text{Adj}[x]$ **faça**

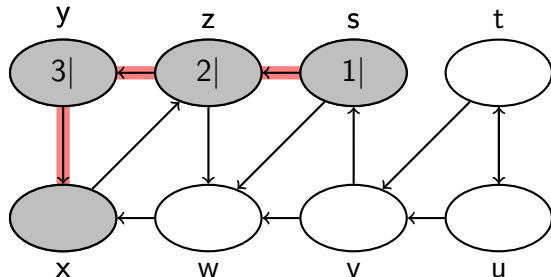
Se $\text{cor}[z] = \text{branco}$ **então**

$\pi[z] \leftarrow x;$

DFS-VISIT (G, V, z, tempo);

$\text{cor}[x] \leftarrow \text{preto};$

$f[x] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, x, 3$)

$\text{cor}[x] \leftarrow \text{cinza};$

$\triangleright \text{tempo} \leftarrow 4;$

$d[x] \leftarrow \text{tempo};$

Para cada $z \in \text{Adj}[x]$ **faça**

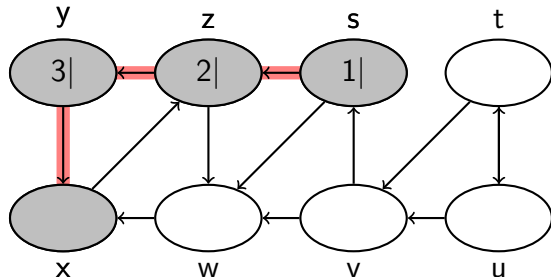
Se $\text{cor}[z] = \text{branco}$ **então**

$\pi[z] \leftarrow x;$

DFS-VISIT ($G, V, z, \text{tempo} + 1$);

$\text{cor}[x] \leftarrow \text{preto};$

$f[x] \leftarrow \text{tempo} + 1;$



DFS-VISIT ($G, V, x, 3$)

$\text{cor}[x] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 4;$

$\triangleright d[x] \leftarrow \text{tempo};$

Para cada $z \in \text{Adj}[x]$ **faça**

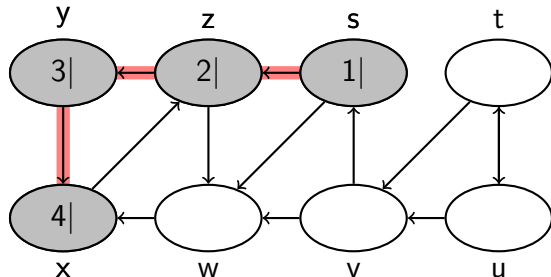
Se $\text{cor}[z] = \text{branco}$ **então**

$\pi[z] \leftarrow x;$

DFS-VISIT ($G, V, z, \text{tempo} + 1$);

$\text{cor}[x] \leftarrow \text{preto};$

$f[x] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, \mathbf{x}, 3$)

$\text{cor}[\mathbf{x}] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 4;$

$d[\mathbf{x}] \leftarrow \text{tempo};$

▷ **Para cada** $\mathbf{z} \in \text{Adj}[\mathbf{x}]$ **faça**

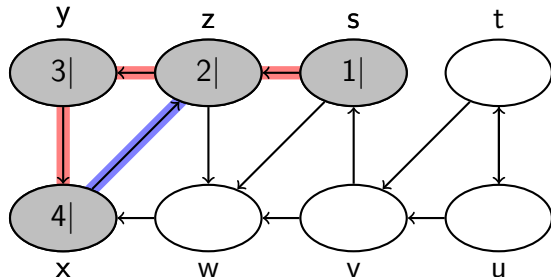
Se $\text{cor}[\mathbf{z}] = \text{branco}$ **então**

$\pi[\mathbf{x}] \leftarrow \mathbf{y};$

DFS-VISIT ($G, V, \mathbf{x}, \text{tempo}$);

$\text{cor}[\mathbf{x}] \leftarrow \text{preto};$

$f[\mathbf{x}] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, \mathbf{x}, 3$)

$\text{cor}[\mathbf{x}] \leftarrow \text{cinza};$

tempo $\leftarrow 4$;

$d[\mathbf{x}] \leftarrow \text{tempo};$

Para cada $\mathbf{z} \in \text{Adj}[\mathbf{x}]$ **faça**

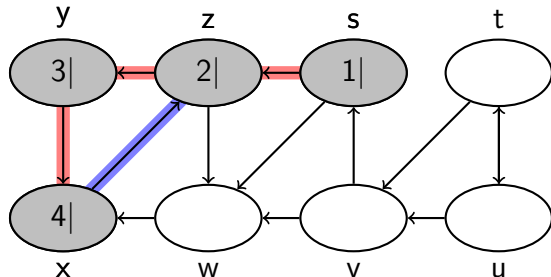
▷ **Se** $\text{cor}[\mathbf{z}] = \text{branco}$ **então**

$\pi[\mathbf{x}] \leftarrow \mathbf{y};$

DFS-VISIT ($G, V, \mathbf{x}, \text{tempo}$);

$\text{cor}[\mathbf{x}] \leftarrow \text{preto};$

$f[\mathbf{x}] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, x, 3$)

$\text{cor}[x] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 4;$

$d[x] \leftarrow \text{tempo};$

Para cada $z \in \text{Adj}[x]$ **faça**

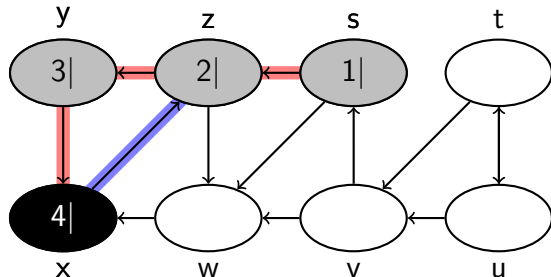
Se $\text{cor}[z] = \text{branco}$ **então**

$\pi[z] \leftarrow x;$

DFS-VISIT ($G, V, z, \text{tempo} + 1$);

$\triangleright \text{cor}[x] \leftarrow \text{preto};$

$f[x] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, x, 3$)

$\text{cor}[x] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 4;$

$d[x] \leftarrow \text{tempo};$

Para cada $z \in \text{Adj}[x]$ **faça**

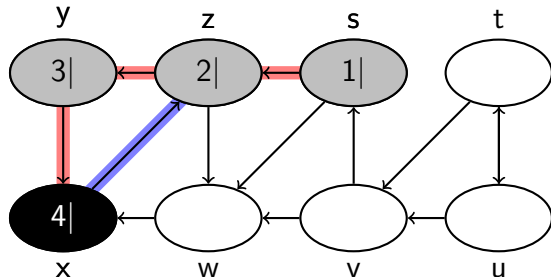
Se $\text{cor}[z] = \text{branco}$ **então**

$\pi[z] \leftarrow x;$

DFS-VISIT ($G, V, z, \text{tempo} + 1$);

$\text{cor}[x] \leftarrow \text{preto};$

$\triangleright f[x] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, x, 3$)

$\text{cor}[x] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 4;$

$d[x] \leftarrow \text{tempo};$

Para cada $z \in \text{Adj}[x]$ **faça**

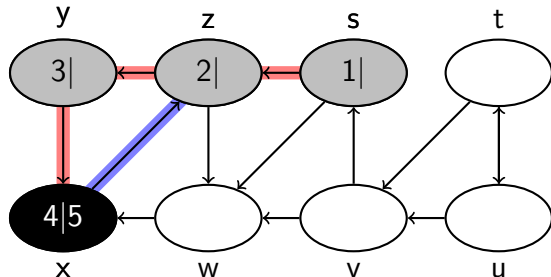
Se $\text{cor}[z] = \text{branco}$ **então**

$\pi[z] \leftarrow x;$

DFS-VISIT ($G, V, z, \text{tempo} + 1$);

$\text{cor}[x] \leftarrow \text{preto};$

$f[x] \leftarrow 5;$



DFS-VISIT ($G, V, y, 2$)

$\text{cor}[y] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[y] \leftarrow \text{tempo};$

Para cada $x \in \text{Adj}[y]$ **faça**

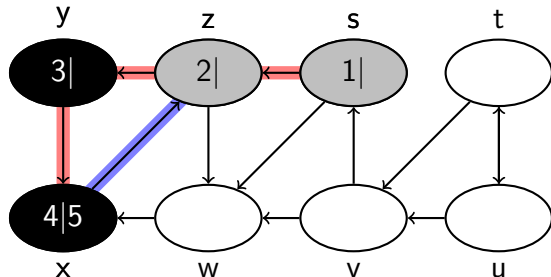
Se $\text{cor}[x] = \text{branco}$ **então**

$\pi[x] \leftarrow y;$

DFS-VISIT ($G, V, x, 5$);

$\triangleright \text{cor}[y] \leftarrow \text{preto};$

$f[y] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, y, 2$)

$\text{cor}[y] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[y] \leftarrow \text{tempo};$

Para cada $x \in \text{Adj}[y]$ **faça**

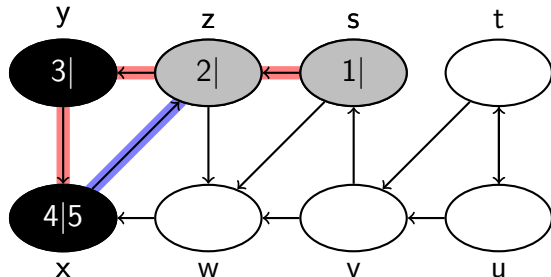
Se $\text{cor}[x] = \text{branco}$ **então**

$\pi[x] \leftarrow y;$

 DFS-VISIT ($G, V, x, 5$);

$\text{cor}[y] \leftarrow \text{preto};$

$\triangleright f[y] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, y, 2$)

$\text{cor}[y] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[y] \leftarrow \text{tempo};$

Para cada $x \in \text{Adj}[y]$ **faça**

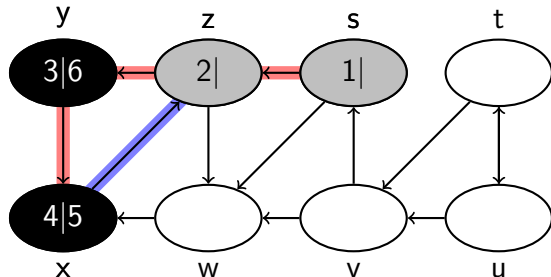
Se $\text{cor}[x] = \text{branco}$ **então**

$\pi[x] \leftarrow y;$

DFS-VISIT ($G, V, x, 5$);

$\text{cor}[y] \leftarrow \text{preto};$

$\triangleright f[y] \leftarrow 6;$



DFS-VISIT ($G, V, z, 1$)

$\text{cor}[z] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[z] \leftarrow \text{tempo};$

Para cada $y \in \text{Adj}[z]$ **faça**

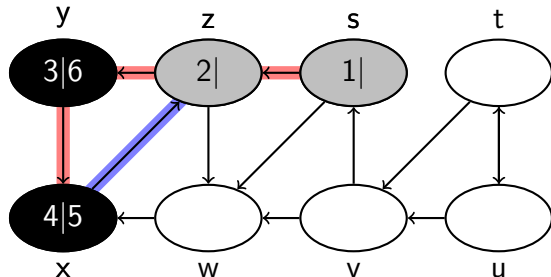
Se $\text{cor}[y] = \text{branco}$ **então**

$\pi[y] \leftarrow z;$

◁ DFS-VISIT ($G, V, y, 6$); **//retorna da recursão**

$\text{cor}[z] \leftarrow \text{preto};$

$f[z] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, z, 1$)

$\text{cor}[z] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[z] \leftarrow \text{tempo};$

▷ **Para** cada $w \in \text{Adj}[z]$ **faça**

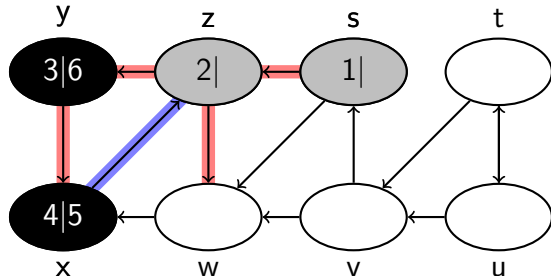
Se $\text{cor}[w] = \text{branco}$ **então**

$\pi[w] \leftarrow z;$

 DFS-VISIT ($G, V, w, 6$);

$\text{cor}[z] \leftarrow \text{preto};$

$f[z] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, z, 1$)

$\text{cor}[z] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[z] \leftarrow \text{tempo};$

Para cada $w \in \text{Adj}[z]$ **faça**

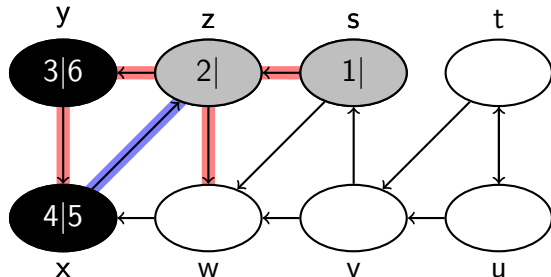
▷ **Se** $\text{cor}[w] = \text{branco}$ **então**

$\pi[w] \leftarrow z;$

DFS-VISIT ($G, V, w, 6$);

$\text{cor}[z] \leftarrow \text{preto};$

$f[z] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, z, 1$)

$\text{cor}[z] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[z] \leftarrow \text{tempo};$

Para cada $w \in \text{Adj}[z]$ **faça**

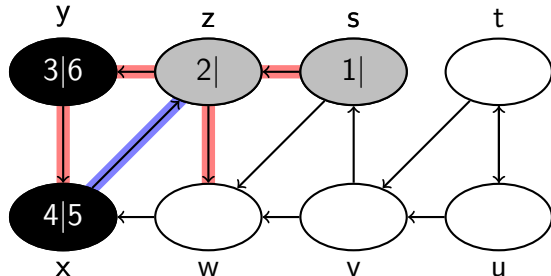
Se $\text{cor}[w] = \text{branco}$ **então**

▷ $\pi[w] \leftarrow z;$

DFS-VISIT ($G, V, w, 6$);

$\text{cor}[z] \leftarrow \text{preto};$

$f[z] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, z, 1$)

$\text{cor}[z] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[z] \leftarrow \text{tempo};$

Para cada $w \in \text{Adj}[z]$ **faça**

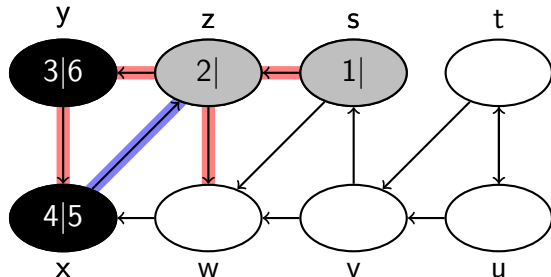
Se $\text{cor}[w] = \text{branco}$ **então**

$\pi[w] \leftarrow z;$

$\triangleright \text{DFS-VISIT } (G, V, w, 6);$

$\text{cor}[z] \leftarrow \text{preto};$

$f[z] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, w, 6$)

$\triangleright \text{cor}[w] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[w] \leftarrow \text{tempo};$

Para cada $x \in \text{Adj}[w]$ **faça**

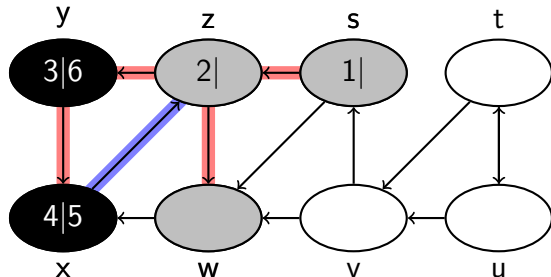
Se $\text{cor}[x] = \text{branco}$ **então**

$\pi[x] \leftarrow w;$

DFS-VISIT (G, V, x, tempo);

$\text{cor}[w] \leftarrow \text{preto};$

$f[w] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, w, 6$)

$\text{cor}[w] \leftarrow \text{cinza};$

$\triangleright \text{tempo} \leftarrow ++\text{tempo};$

$d[w] \leftarrow \text{tempo};$

Para cada $x \in \text{Adj}[w]$ **faça**

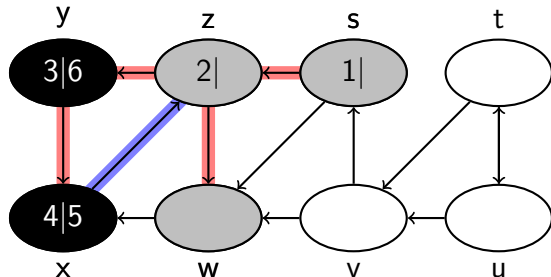
Se $\text{cor}[x] = \text{branco}$ **então**

$\pi[x] \leftarrow w;$

DFS-VISIT (G, V, x, tempo);

$\text{cor}[w] \leftarrow \text{preto};$

$f[w] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, w, 6$)

$cor[w] \leftarrow \text{cinza};$

$\triangleright \text{tempo} \leftarrow 7;$

$d[w] \leftarrow \text{tempo};$

Para cada $x \in Adj[w]$ **faça**

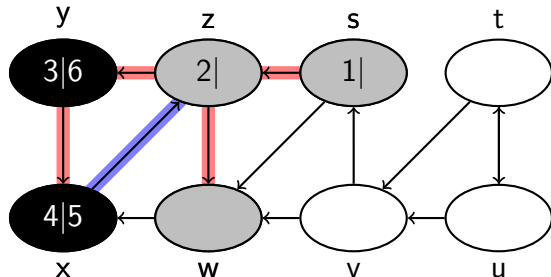
Se $cor[x] = \text{branco}$ **então**

$\pi[x] \leftarrow w;$

 DFS-VISIT (G, V, x, tempo);

$cor[w] \leftarrow \text{preto};$

$f[w] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, w, 6$)

$\text{cor}[w] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 7;$

$\triangleright d[w] \leftarrow \text{tempo};$

Para cada $x \in \text{Adj}[w]$ **faça**

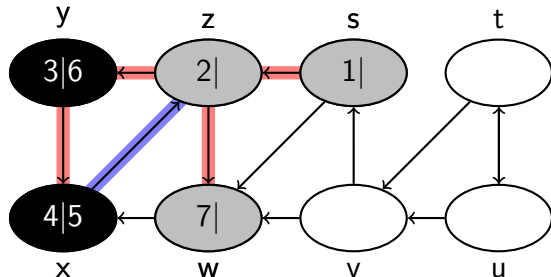
Se $\text{cor}[x] = \text{branco}$ **então**

$\pi[x] \leftarrow w;$

$\triangleright \text{DFS-VISIT}(G, V, x, \text{tempo});$

$\text{cor}[w] \leftarrow \text{preto};$

$f[w] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, w, 6$)

$\text{cor}[w] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 7;$

$d[w] \leftarrow \text{tempo};$

▷ Para cada $x \in \text{Adj}[w]$ faça

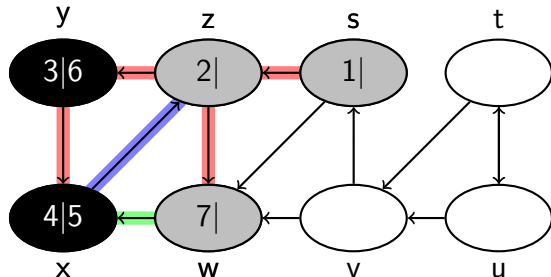
Se $\text{cor}[x] = \text{branco}$ então

$\pi[x] \leftarrow w;$

DFS-VISIT (G, V, x, tempo);

$\text{cor}[w] \leftarrow \text{preto};$

$f[w] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, w, 6$)

$\text{cor}[w] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 7;$

$d[w] \leftarrow \text{tempo};$

Para cada $x \in \text{Adj}[w]$ **faça**

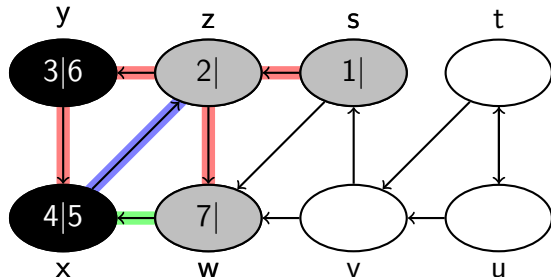
▷ **Se** $\text{cor}[x] = \text{branco}$ **então**

$\pi[x] \leftarrow w;$

DFS-VISIT (G, V, x, tempo);

$\text{cor}[w] \leftarrow \text{preto};$

$f[w] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, w, 6$)

$\text{cor}[w] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 7;$

$d[w] \leftarrow \text{tempo};$

Para cada $x \in \text{Adj}[w]$ **faça**

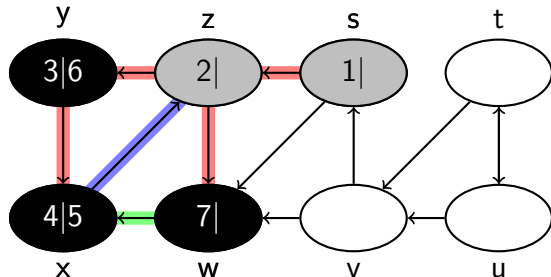
Se $\text{cor}[x] = \text{branco}$ **então**

$\pi[x] \leftarrow w;$

DFS-VISIT (G, V, x, tempo);

$\triangleright \text{cor}[w] \leftarrow \text{preto};$

$f[w] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, w, 6$)

$\text{cor}[w] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 7;$

$d[w] \leftarrow \text{tempo};$

Para cada $x \in \text{Adj}[w]$ **faça**

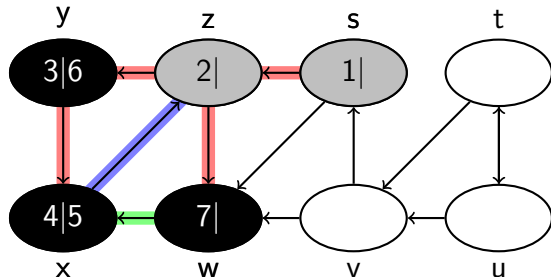
Se $\text{cor}[x] = \text{branco}$ **então**

$\pi[x] \leftarrow w;$

DFS-VISIT (G, V, x, tempo);

$\text{cor}[w] \leftarrow \text{preto};$

$\triangleright f[w] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, w, 6$)

$\text{cor}[w] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 7;$

$d[w] \leftarrow \text{tempo};$

Para cada $x \in \text{Adj}[w]$ **faça**

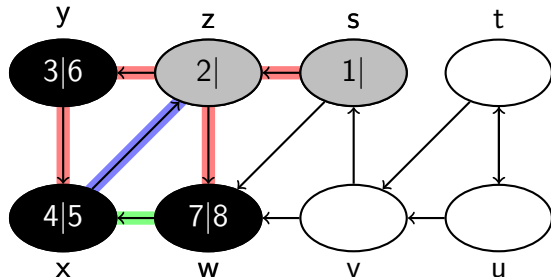
Se $\text{cor}[x] = \text{branco}$ **então**

$\pi[x] \leftarrow w;$

DFS-VISIT (G, V, x, tempo);

$\text{cor}[w] \leftarrow \text{preto};$

$\triangleright f[w] \leftarrow 8;$



DFS-VISIT ($G, V, z, 2$)

$\text{cor}[z] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[z] \leftarrow \text{tempo};$

Para cada $w \in \text{Adj}[z]$ **faça**

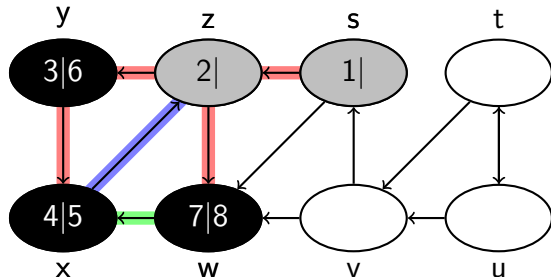
Se $\text{cor}[w] = \text{branco}$ **então**

$\pi[w] \leftarrow z;$

◁ DFS-VISIT ($G, V, w, 8$); **//retorna da recursão**

$\text{cor}[z] \leftarrow \text{preto};$

$f[z] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, z, 2$)

$\text{cor}[z] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[z] \leftarrow \text{tempo};$

Para cada $w \in \text{Adj}[z]$ **faça**

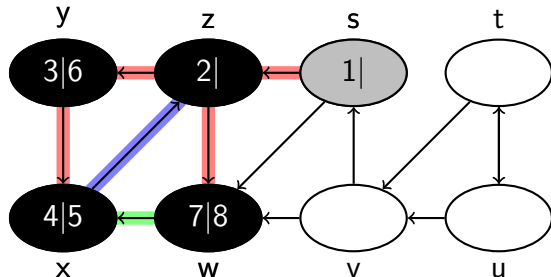
Se $\text{cor}[w] = \text{branco}$ **então**

$\pi[w] \leftarrow z;$

 DFS-VISIT ($G, V, w, 8$);

▷ $\text{cor}[z] \leftarrow \text{preto};$

$f[z] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, z, 2$)

$\text{cor}[z] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[z] \leftarrow \text{tempo};$

Para cada $w \in \text{Adj}[z]$ **faça**

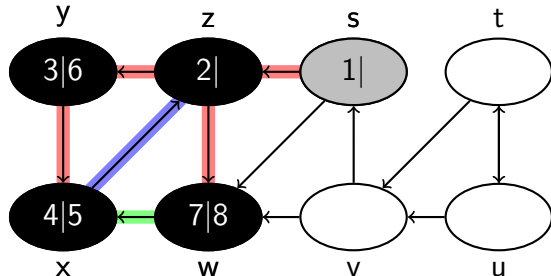
Se $\text{cor}[w] = \text{branco}$ **então**

$\pi[w] \leftarrow z;$

 DFS-VISIT ($G, V, w, 8$);

$\text{cor}[z] \leftarrow \text{preto};$

▷ $f[z] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, z, 2$)

$\text{cor}[z] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[z] \leftarrow \text{tempo};$

Para cada $w \in \text{Adj}[z]$ **faça**

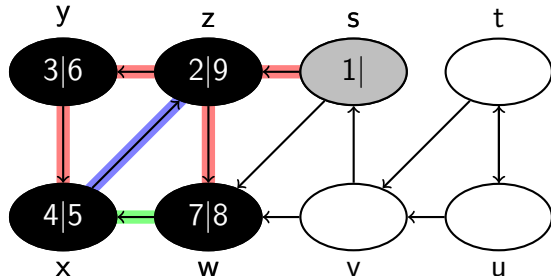
Se $\text{cor}[w] = \text{branco}$ **então**

$\pi[w] \leftarrow z;$

 DFS-VISIT ($G, V, w, 8$);

$\text{cor}[z] \leftarrow \text{preto};$

▷ $f[z] \leftarrow 9;$



DFS-VISIT ($G, V, s, 1$)

$cor[s] \leftarrow \text{cinza};$

$tempo \leftarrow ++tempo;$

$d[s] \leftarrow tempo;$

Para cada $z \in Adj[s]$ **faça**

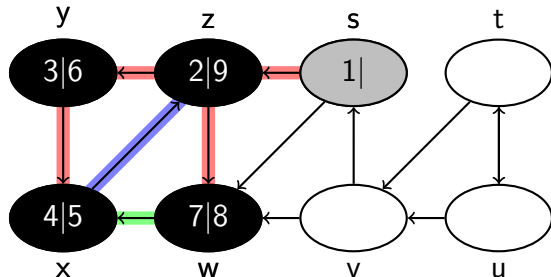
Se $cor[z] = \text{branco}$ **então**

$\pi[z] \leftarrow s;$

◁ DFS-VISIT ($G, V, z, 9$); **//retorna da recursão**

$cor[s] \leftarrow \text{preto};$

$f[s] \leftarrow ++tempo;$



DFS-VISIT ($G, V, s, 1$)

$\text{cor}[s] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[s] \leftarrow \text{tempo};$

▷ Para cada $w \in \text{Adj}[s]$ faça

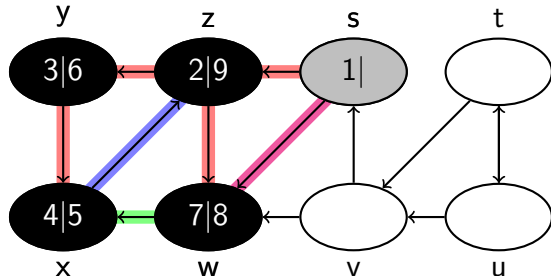
Se $\text{cor}[w] = \text{branco}$ então

$\pi[w] \leftarrow s;$

DFS-VISIT ($G, V, w, 9$);

$\text{cor}[s] \leftarrow \text{preto};$

$f[s] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, s, 1$)

$\text{cor}[s] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[s] \leftarrow \text{tempo};$

Para cada $w \in \text{Adj}[s]$ **faça**

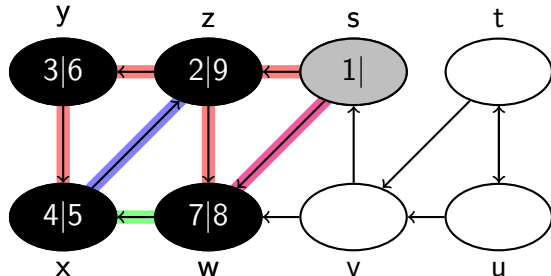
▷ **Se** $\text{cor}[w] = \text{branco}$ **então**

$\pi[w] \leftarrow s;$

DFS-VISIT ($G, V, w, 9$);

$\text{cor}[s] \leftarrow \text{preto};$

$f[s] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, s, 1$)

$\text{cor}[s] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[s] \leftarrow \text{tempo};$

Para cada $w \in \text{Adj}[s]$ **faça**

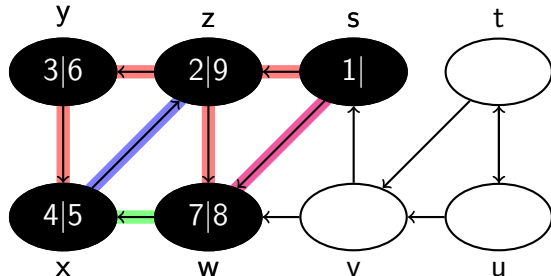
Se $\text{cor}[w] = \text{branco}$ **então**

$\pi[w] \leftarrow s;$

 DFS-VISIT ($G, V, w, 9$);

▷ $\text{cor}[s] \leftarrow \text{preto};$

$f[s] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, s, 1$)

$cor[s] \leftarrow \text{cinza};$

$tempo \leftarrow ++tempo;$

$d[s] \leftarrow tempo;$

Para cada $w \in Adj[s]$ **faça**

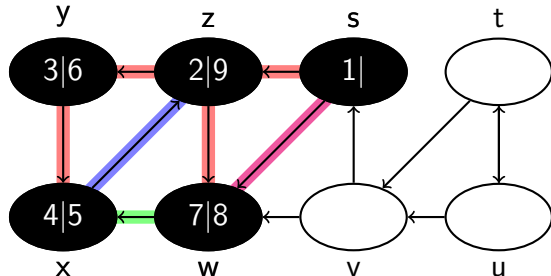
Se $cor[w] = \text{branco}$ **então**

$\pi[w] \leftarrow s;$

 DFS-VISIT ($G, V, w, 9$);

$cor[s] \leftarrow \text{preto};$

$\triangleright f[s] \leftarrow ++tempo;$



DFS-VISIT ($G, V, s, 1$)

$\text{cor}[s] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[s] \leftarrow \text{tempo};$

Para cada $w \in \text{Adj}[s]$ **faça**

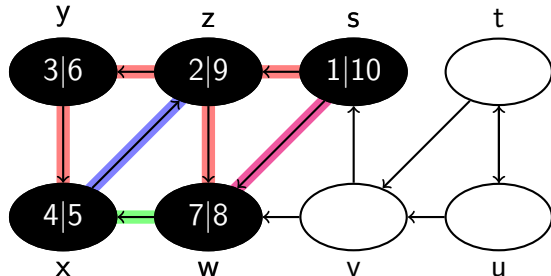
Se $\text{cor}[w] = \text{branco}$ **então**

$\pi[w] \leftarrow s;$

DFS-VISIT ($G, V, w, 9$);

$\text{cor}[s] \leftarrow \text{preto};$

$\triangleright f[s] \leftarrow 10;$



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ **faça**

$\text{cor}[u] \leftarrow \text{branco};$

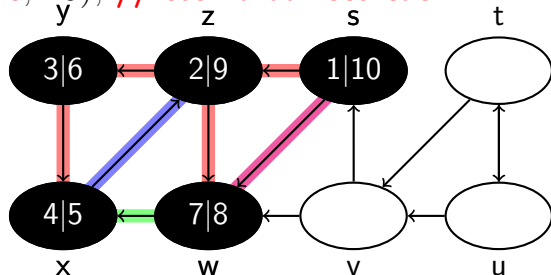
$\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0;$

Para cada $s \in V[G]$ **faça**

Se $\text{cor}[s] = \text{branco}$ **então**

\triangleleft DFS-VISIT($G, V, s, 10$); **//retorna da recursão**



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ **faça**

$\text{cor}[u] \leftarrow \text{branco};$

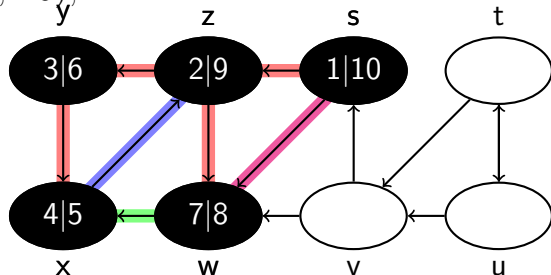
$\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0;$

▷ **Para cada** $t \in V[G]$ **faça**

Se $\text{cor}[t] = \text{branco}$ **então**

 DFS-VISIT($G, V, t, 10$);



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ **faça**

$\text{cor}[u] \leftarrow \text{branco};$

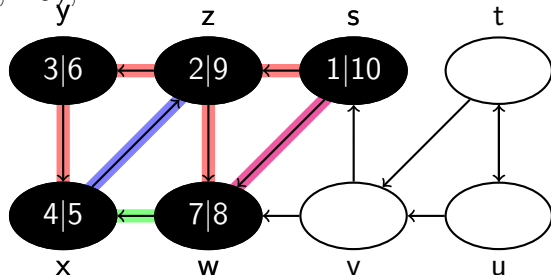
$\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0;$

Para cada $t \in V[G]$ **faça**

\triangleright **Se** $\text{cor}[t] = \text{branco}$ **então**

 DFS-VISIT($G, V, t, 10$);



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ **faça**

$\text{cor}[u] \leftarrow \text{branco};$

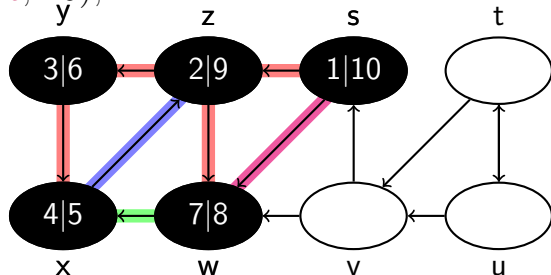
$\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0;$

Para cada $t \in V[G]$ **faça**

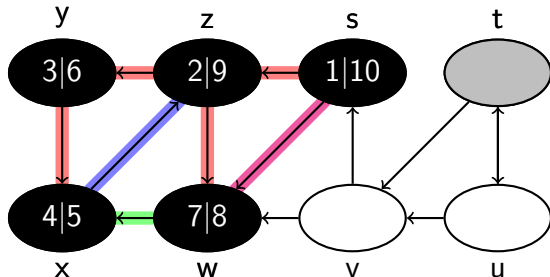
Se $\text{cor}[t] = \text{branco}$ **então**

$\triangleright \text{DFS-VISIT}(G, V, t, 10);$



DFS-VISIT ($G, V, t, 10$)

```
▷ cor[t] ← cinza;  
tempo ← ++tempo;  
d[t] ← tempo;  
Para cada  $v \in Adj[t]$  faça  
  Se cor[v] = branco então  
     $\pi[v] \leftarrow t$ ;  
    DFS-VISIT ( $G, V, v, tempo$ );  
cor[t] ← preto;  
f[t] ← ++tempo;
```



DFS-VISIT ($G, V, t, 10$)

$\text{cor}[t] \leftarrow \text{cinza};$

$\triangleright \text{tempo} \leftarrow ++\text{tempo};$

$d[t] \leftarrow \text{tempo};$

Para cada $v \in \text{Adj}[t]$ **faça**

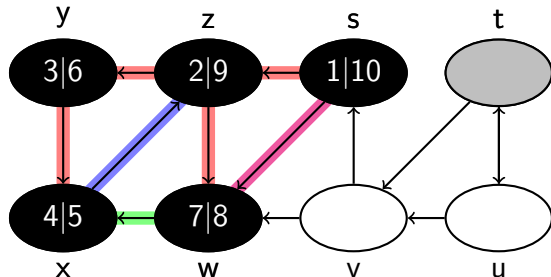
Se $\text{cor}[v] = \text{branco}$ **então**

$\pi[v] \leftarrow t;$

DFS-VISIT (G, V, v, tempo);

$\text{cor}[t] \leftarrow \text{preto};$

$f[t] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, t, 10$)

$\text{cor}[t] \leftarrow \text{cinza};$

$\triangleright \text{tempo} \leftarrow 11;$

$d[t] \leftarrow \text{tempo};$

Para cada $v \in \text{Adj}[t]$ **faça**

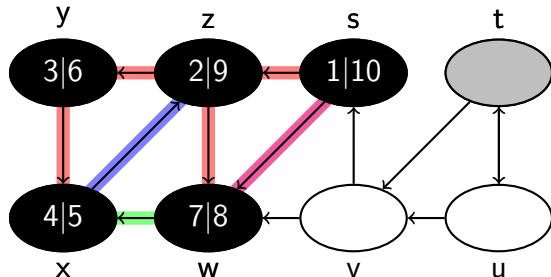
Se $\text{cor}[v] = \text{branco}$ **então**

$\pi[v] \leftarrow t;$

 DFS-VISIT (G, V, v, tempo);

$\text{cor}[t] \leftarrow \text{preto};$

$f[t] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, t, 10$)

$\text{cor}[t] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 11;$

$\triangleright \text{d}[t] \leftarrow \text{tempo};$

Para cada $v \in \text{Adj}[t]$ **faça**

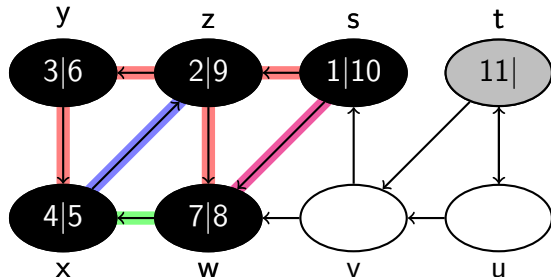
Se $\text{cor}[v] = \text{branco}$ **então**

$\pi[v] \leftarrow t;$

DFS-VISIT (G, V, v, tempo);

$\text{cor}[t] \leftarrow \text{preto};$

$f[t] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, t, 10$)

$\text{cor}[t] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 11;$

$d[t] \leftarrow \text{tempo};$

▷ **Para cada** $v \in \text{Adj}[t]$ **faça**

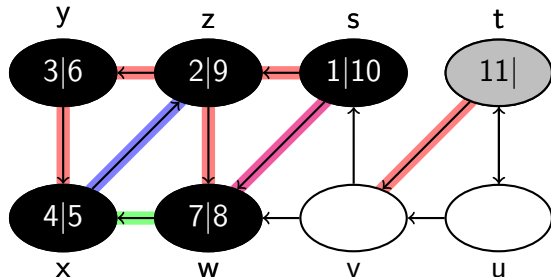
Se $\text{cor}[v] = \text{branco}$ **então**

$\pi[v] \leftarrow t;$

 DFS-VISIT (G, V, v, tempo);

$\text{cor}[t] \leftarrow \text{preto};$

$f[t] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, t, 10$)

$\text{cor}[t] \leftarrow \text{cinza};$

tempo $\leftarrow 11;$

$d[t] \leftarrow \text{tempo};$

Para cada $v \in \text{Adj}[t]$ **faça**

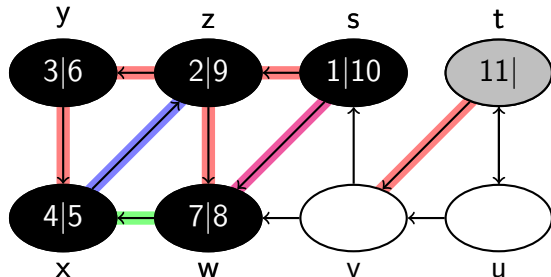
▷ **Se** $\text{cor}[v] = \text{branco}$ **então**

$\pi[v] \leftarrow t;$

DFS-VISIT (G, V, v, tempo);

$\text{cor}[t] \leftarrow \text{preto};$

$f[t] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, t, 10$)

$\text{cor}[t] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 11;$

$d[t] \leftarrow \text{tempo};$

Para cada $v \in \text{Adj}[t]$ **faça**

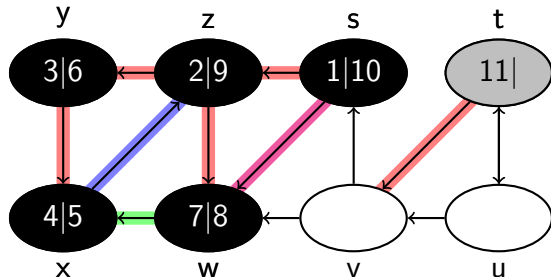
Se $\text{cor}[v] = \text{branco}$ **então**

▷ $\pi[v] \leftarrow t;$

DFS-VISIT (G, V, v, tempo);

$\text{cor}[t] \leftarrow \text{preto};$

$f[t] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, t, 10$)

$\text{cor}[t] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 11;$

$d[t] \leftarrow \text{tempo};$

Para cada $v \in \text{Adj}[t]$ **faça**

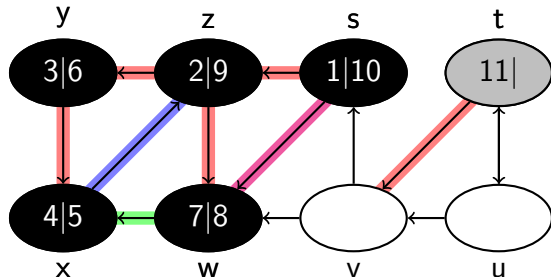
Se $\text{cor}[v] = \text{branco}$ **então**

$\pi[v] \leftarrow t;$

▷ DFS-VISIT (G, V, v, tempo);

$\text{cor}[t] \leftarrow \text{preto};$

$f[t] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, t, 10$)

$\text{cor}[t] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 11;$

$d[t] \leftarrow \text{tempo};$

Para cada $v \in \text{Adj}[t]$ **faça**

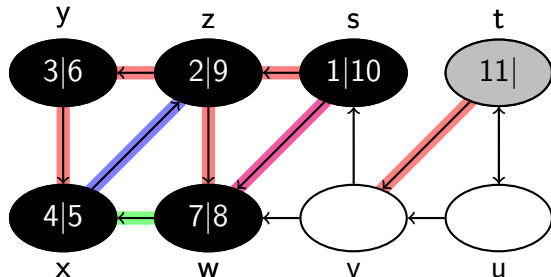
Se $\text{cor}[v] = \text{branco}$ **então**

$\pi[v] \leftarrow t;$

▷ DFS-VISIT ($G, V, v, 11$);

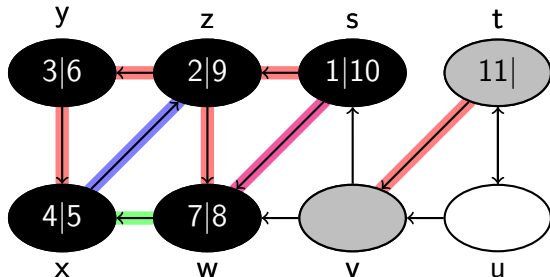
$\text{cor}[t] \leftarrow \text{preto};$

$f[t] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, v, 11$)

```
▷ cor[v] ← cinza;  
tempo ← ++tempo;  
d[v] ← tempo;  
Para cada  $w \in Adj[v]$  faça  
  Se cor[w] = branco então  
     $\pi[w] \leftarrow v$ ;  
    DFS-VISIT ( $G, V, w, tempo$ );  
cor[v] ← preto;  
f[v] ← ++tempo;
```



DFS-VISIT ($G, V, v, 11$)

$\text{cor}[v] \leftarrow \text{cinza};$

$\triangleright \text{tempo} \leftarrow ++\text{tempo};$

$d[v] \leftarrow \text{tempo};$

Para cada $w \in \text{Adj}[v]$ **faça**

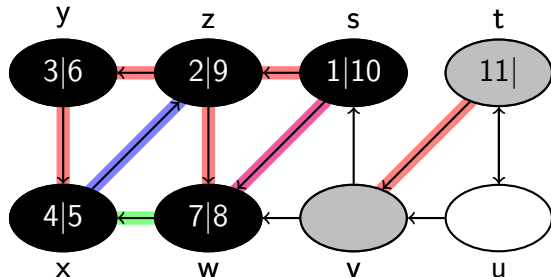
Se $\text{cor}[w] = \text{branco}$ **então**

$\pi[w] \leftarrow v;$

DFS-VISIT (G, V, w, tempo);

$\text{cor}[v] \leftarrow \text{preto};$

$f[v] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, v, 11$)

$\text{cor}[v] \leftarrow \text{cinza};$

$\triangleright \text{tempo} \leftarrow 12;$

$d[v] \leftarrow \text{tempo};$

Para cada $w \in \text{Adj}[v]$ **faça**

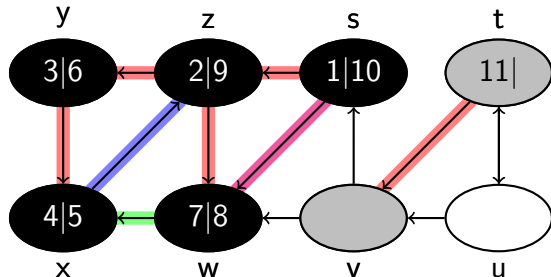
Se $\text{cor}[w] = \text{branco}$ **então**

$\pi[w] \leftarrow v;$

DFS-VISIT (G, V, w, tempo);

$\text{cor}[v] \leftarrow \text{preto};$

$f[v] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, v, 11$)

$\text{cor}[v] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 12;$

$\triangleright d[v] \leftarrow \text{tempo};$

Para cada $w \in \text{Adj}[v]$ **faça**

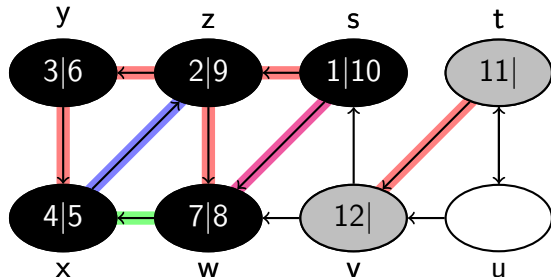
Se $\text{cor}[w] = \text{branco}$ **então**

$\pi[w] \leftarrow v;$

DFS-VISIT (G, V, w, tempo);

$\text{cor}[v] \leftarrow \text{preto};$

$f[v] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, v, 11$)

$\text{cor}[v] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 12;$

$d[v] \leftarrow \text{tempo};$

▷ **Para cada** $w \in \text{Adj}[v]$ **faça**

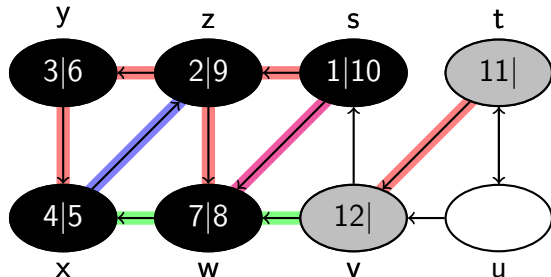
Se $\text{cor}[w] = \text{branco}$ **então**

$\pi[w] \leftarrow v;$

 DFS-VISIT (G, V, w, tempo);

$\text{cor}[v] \leftarrow \text{preto};$

$f[v] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, v, 11$)

$\text{cor}[v] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 12;$

$d[v] \leftarrow \text{tempo};$

Para cada $w \in \text{Adj}[v]$ **faça**

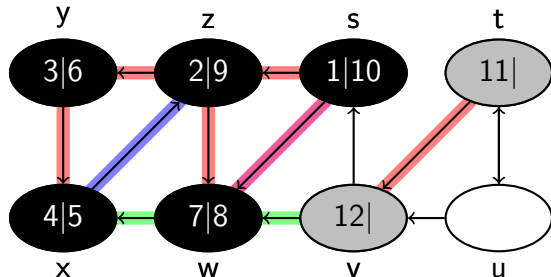
▷ **Se** $\text{cor}[w] = \text{branco}$ **então**

$\pi[w] \leftarrow v;$

DFS-VISIT (G, V, w, tempo);

$\text{cor}[v] \leftarrow \text{preto};$

$f[v] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, v, 11$)

$\text{cor}[v] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 12;$

$d[v] \leftarrow \text{tempo};$

▷ **Para cada** $s \in \text{Adj}[v]$ **faça**

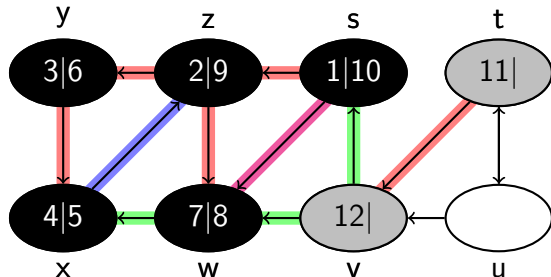
Se $\text{cor}[s] = \text{branco}$ **então**

$\pi[s] \leftarrow v;$

 DFS-VISIT (G, V, s, tempo);

$\text{cor}[v] \leftarrow \text{preto};$

$f[v] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, v, 11$)

$\text{cor}[v] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 12;$

$d[v] \leftarrow \text{tempo};$

Para cada $s \in \text{Adj}[v]$ **faça**

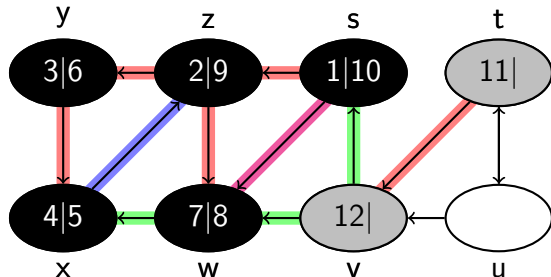
▷ **Se** $\text{cor}[s] = \text{branco}$ **então**

$\pi[s] \leftarrow v;$

DFS-VISIT (G, V, s, tempo);

$\text{cor}[v] \leftarrow \text{preto};$

$f[v] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, v, 11$)

$\text{cor}[v] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 12;$

$d[v] \leftarrow \text{tempo};$

Para cada $s \in \text{Adj}[v]$ **faça**

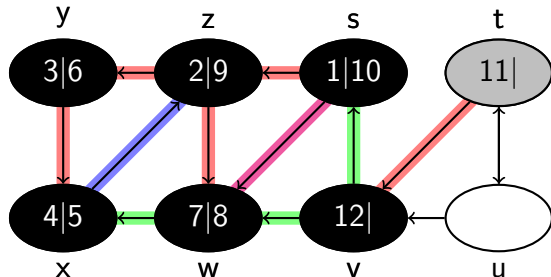
Se $\text{cor}[s] = \text{branco}$ **então**

$\pi[s] \leftarrow v;$

DFS-VISIT (G, V, s, tempo);

$\triangleright \text{cor}[v] \leftarrow \text{preto};$

$f[v] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, v, 11$)

$\text{cor}[v] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 12;$

$d[v] \leftarrow \text{tempo};$

Para cada $s \in \text{Adj}[v]$ **faça**

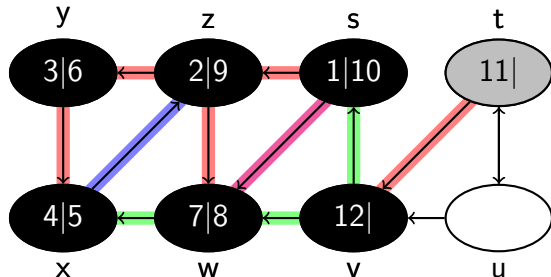
Se $\text{cor}[s] = \text{branco}$ **então**

$\pi[s] \leftarrow v;$

DFS-VISIT (G, V, s, tempo);

$\text{cor}[v] \leftarrow \text{preto};$

$\triangleright f[v] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, v, 11$)

$\text{cor}[v] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 12;$

$d[v] \leftarrow \text{tempo};$

Para cada $s \in \text{Adj}[v]$ **faça**

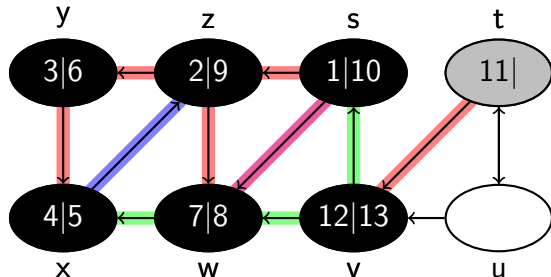
Se $\text{cor}[s] = \text{branco}$ **então**

$\pi[s] \leftarrow v;$

DFS-VISIT (G, V, s, tempo);

$\text{cor}[v] \leftarrow \text{preto};$

$\triangleright f[v] \leftarrow 13;$



DFS-VISIT ($G, V, t, 10$)

$\text{cor}[t] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[t] \leftarrow \text{tempo};$

Para cada $v \in \text{Adj}[t]$ **faça**

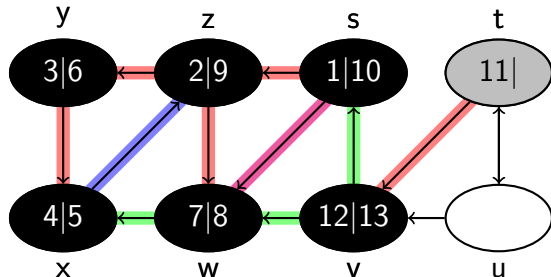
Se $\text{cor}[v] = \text{branco}$ **então**

$\pi[v] \leftarrow t;$

◁ DFS-VISIT ($G, V, v, 13$); // **retorna da recursão**

$\text{cor}[t] \leftarrow \text{preto};$

$f[t] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, t, 10$)

$\text{cor}[t] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[t] \leftarrow \text{tempo};$

▷ **Para cada** $u \in \text{Adj}[t]$ **faça**

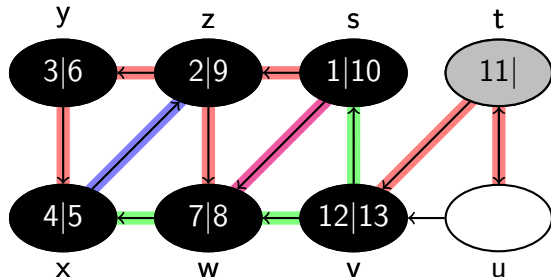
Se $\text{cor}[u] = \text{branco}$ **então**

$\pi[u] \leftarrow t;$

 DFS-VISIT ($G, V, u, 13$);

$\text{cor}[t] \leftarrow \text{preto};$

$f[t] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, t, 10$)

$\text{cor}[t] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[t] \leftarrow \text{tempo};$

Para cada $u \in \text{Adj}[t]$ **faça**

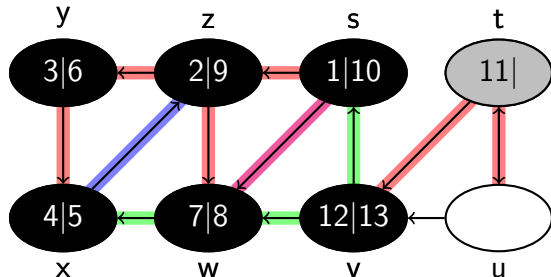
▷ **Se** $\text{cor}[u] = \text{branco}$ **então**

$\pi[u] \leftarrow t;$

DFS-VISIT ($G, V, u, 13$);

$\text{cor}[t] \leftarrow \text{preto};$

$f[t] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, t, 10$)

$\text{cor}[t] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[t] \leftarrow \text{tempo};$

Para cada $u \in \text{Adj}[t]$ **faça**

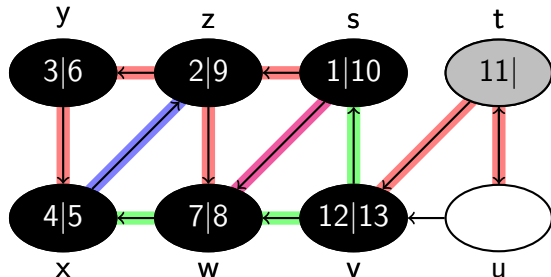
Se $\text{cor}[u] = \text{branco}$ **então**

▷ $\pi[u] \leftarrow t;$

DFS-VISIT ($G, V, u, 13$);

$\text{cor}[t] \leftarrow \text{preto};$

$f[t] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, t, 10$)

$\text{cor}[t] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[t] \leftarrow \text{tempo};$

Para cada $u \in \text{Adj}[t]$ **faça**

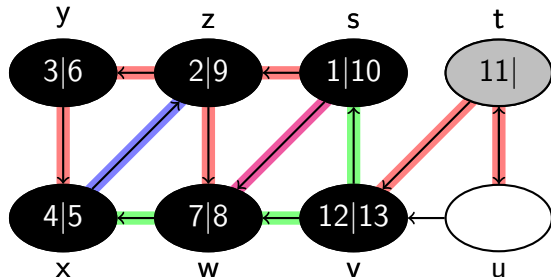
Se $\text{cor}[u] = \text{branco}$ **então**

$\pi[u] \leftarrow t;$

▷ DFS-VISIT ($G, V, u, 13$);

$\text{cor}[t] \leftarrow \text{preto};$

$f[t] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, u, 13$)

$\triangleright \text{cor}[u] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[u] \leftarrow \text{tempo};$

Para cada $v \in \text{Adj}[u]$ **faça**

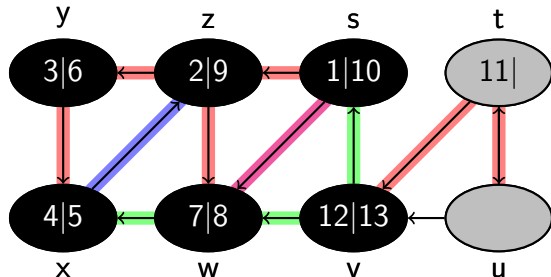
Se $\text{cor}[v] = \text{branco}$ **então**

$\pi[v] \leftarrow v;$

DFS-VISIT (G, V, v, tempo);

$\text{cor}[u] \leftarrow \text{preto};$

$f[u] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, u, 13$)

$\text{cor}[u] \leftarrow \text{cinza};$

$\triangleright \text{tempo} \leftarrow ++\text{tempo};$

$d[u] \leftarrow \text{tempo};$

Para cada $v \in \text{Adj}[u]$ **faça**

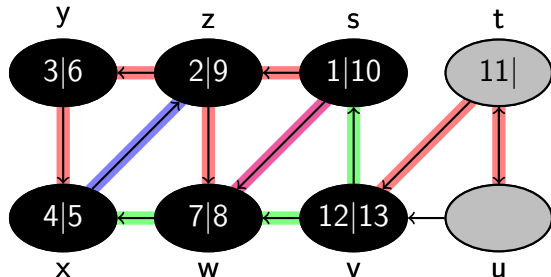
Se $\text{cor}[v] = \text{branco}$ **então**

$\pi[v] \leftarrow v;$

DFS-VISIT (G, V, v, tempo);

$\text{cor}[u] \leftarrow \text{preto};$

$f[u] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, u, 13$)

$\text{cor}[u] \leftarrow \text{cinza};$

$\triangleright \text{tempo} \leftarrow 14;$

$d[u] \leftarrow \text{tempo};$

Para cada $v \in \text{Adj}[u]$ **faça**

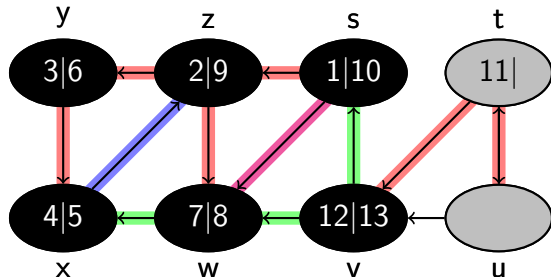
Se $\text{cor}[v] = \text{branco}$ **então**

$\pi[v] \leftarrow v;$

DFS-VISIT (G, V, v, tempo);

$\text{cor}[u] \leftarrow \text{preto};$

$f[u] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, u, 13$)

$\text{cor}[u] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 14;$

$\triangleright \text{d}[u] \leftarrow \text{tempo};$

Para cada $v \in \text{Adj}[u]$ **faça**

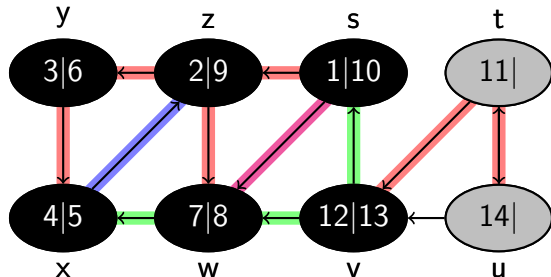
Se $\text{cor}[v] = \text{branco}$ **então**

$\pi[v] \leftarrow v;$

DFS-VISIT (G, V, v, tempo);

$\text{cor}[u] \leftarrow \text{preto};$

$f[u] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, u, 13$)

$\text{cor}[u] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 14;$

$d[u] \leftarrow \text{tempo};$

▷ **Para cada** $v \in \text{Adj}[u]$ **faça**

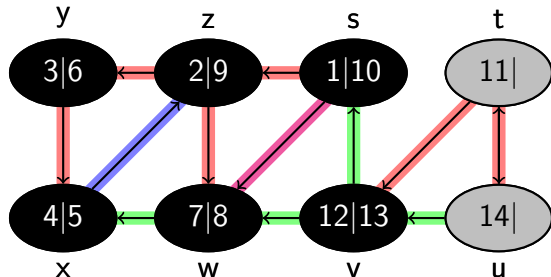
Se $\text{cor}[v] = \text{branco}$ **então**

$\pi[v] \leftarrow v;$

 DFS-VISIT (G, V, v, tempo);

$\text{cor}[u] \leftarrow \text{preto};$

$f[u] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, u, 13$)

$\text{cor}[u] \leftarrow \text{cinza};$

tempo $\leftarrow 14;$

$d[u] \leftarrow \text{tempo};$

Para cada $v \in \text{Adj}[u]$ **faça**

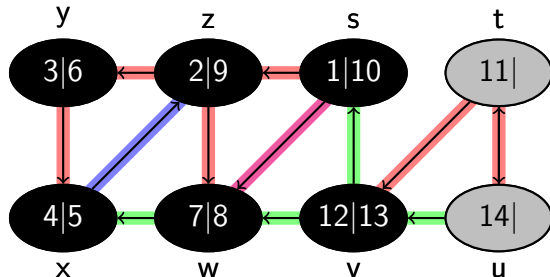
▷ **Se** $\text{cor}[v] = \text{branco}$ **então**

$\pi[v] \leftarrow v;$

DFS-VISIT (G, V, v, tempo);

$\text{cor}[u] \leftarrow \text{preto};$

$f[u] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, u, 13$)

$\text{cor}[u] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 14;$

$d[u] \leftarrow \text{tempo};$

▷ Para cada $t \in \text{Adj}[u]$ faça

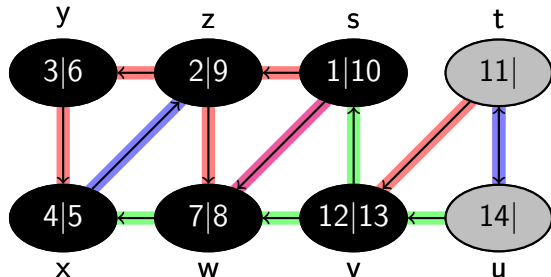
Se $\text{cor}[t] = \text{branco}$ então

$\pi[v] \leftarrow t;$

DFS-VISIT (G, V, t, tempo);

$\text{cor}[u] \leftarrow \text{preto};$

$f[u] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, u, 13$)

$\text{cor}[u] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 14;$

$d[u] \leftarrow \text{tempo};$

Para cada $t \in \text{Adj}[u]$ **faça**

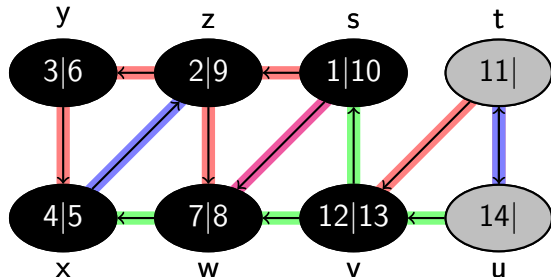
▷ **Se** $\text{cor}[t] = \text{branco}$ **então**

$\pi[v] \leftarrow t;$

DFS-VISIT (G, V, t, tempo);

$\text{cor}[u] \leftarrow \text{preto};$

$f[u] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, u, 13$)

$\text{cor}[u] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 14;$

$d[u] \leftarrow \text{tempo};$

Para cada $t \in \text{Adj}[u]$ **faça**

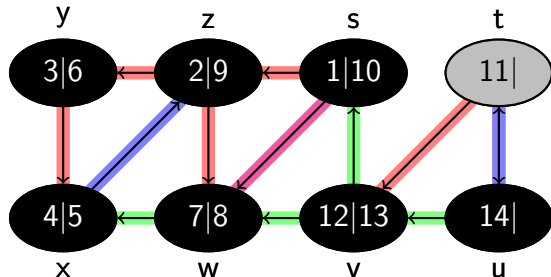
Se $\text{cor}[t] = \text{branco}$ **então**

$\pi[v] \leftarrow t;$

DFS-VISIT (G, V, t, tempo);

$\triangleright \text{cor}[u] \leftarrow \text{preto};$

$f[u] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, u, 13$)

$\text{cor}[u] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 14;$

$d[u] \leftarrow \text{tempo};$

Para cada $t \in \text{Adj}[u]$ **faça**

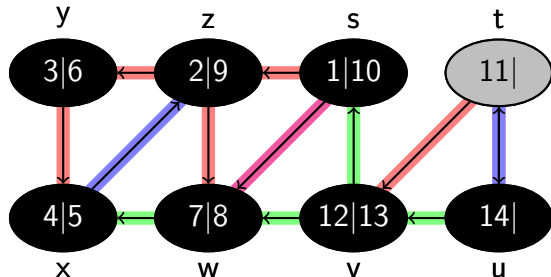
Se $\text{cor}[t] = \text{branco}$ **então**

$\pi[v] \leftarrow t;$

DFS-VISIT (G, V, t, tempo);

$\text{cor}[u] \leftarrow \text{preto};$

$\triangleright f[u] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, u, 13$)

$\text{cor}[u] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow 14;$

$d[u] \leftarrow \text{tempo};$

Para cada $t \in \text{Adj}[u]$ **faça**

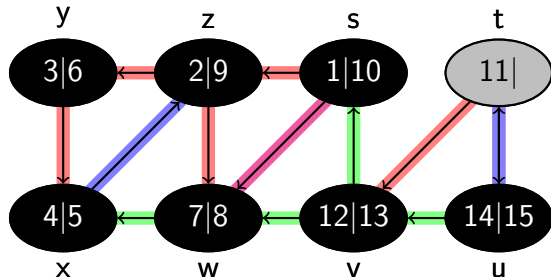
Se $\text{cor}[t] = \text{branco}$ **então**

$\pi[v] \leftarrow t;$

DFS-VISIT (G, V, t, tempo);

$\text{cor}[u] \leftarrow \text{preto};$

$\triangleright f[u] \leftarrow 15;$



DFS-VISIT ($G, V, t, 10$)

$\text{cor}[t] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[t] \leftarrow \text{tempo};$

Para cada $u \in \text{Adj}[t]$ **faça**

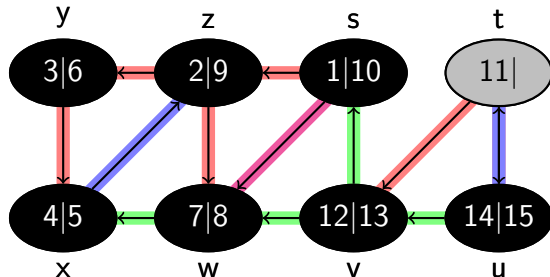
Se $\text{cor}[u] = \text{branco}$ **então**

$\pi[u] \leftarrow t;$

$\triangleleft \text{DFS-VISIT}(G, V, u, 15); //$ **retorna da recursão**

$\text{cor}[t] \leftarrow \text{preto};$

$f[t] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, t, 10$)

$\text{cor}[t] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[t] \leftarrow \text{tempo};$

Para cada $u \in \text{Adj}[t]$ **faça**

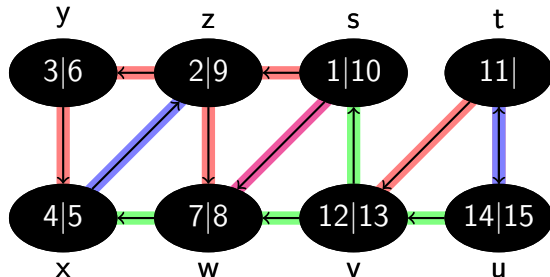
Se $\text{cor}[u] = \text{branco}$ **então**

$\pi[u] \leftarrow t;$

DFS-VISIT ($G, V, u, 15$);

$\triangleright \text{cor}[t] \leftarrow \text{preto};$

$f[t] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, t, 10$)

$\text{cor}[t] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[t] \leftarrow \text{tempo};$

Para cada $u \in \text{Adj}[t]$ **faça**

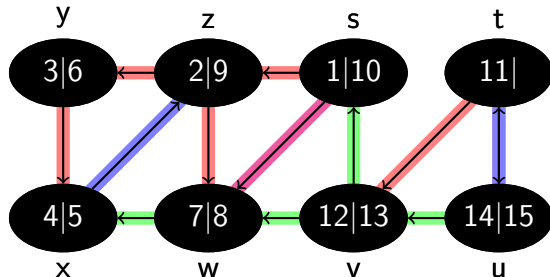
Se $\text{cor}[u] = \text{branco}$ **então**

$\pi[u] \leftarrow t;$

DFS-VISIT ($G, V, u, 15$);

$\text{cor}[t] \leftarrow \text{preto};$

$\triangleright f[t] \leftarrow ++\text{tempo};$



DFS-VISIT ($G, V, t, 10$)

$\text{cor}[t] \leftarrow \text{cinza};$

$\text{tempo} \leftarrow ++\text{tempo};$

$d[t] \leftarrow \text{tempo};$

Para cada $u \in \text{Adj}[t]$ **faça**

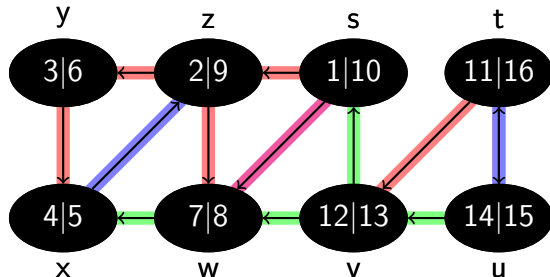
Se $\text{cor}[u] = \text{branco}$ **então**

$\pi[u] \leftarrow t;$

 DFS-VISIT ($G, V, u, 15$);

$\text{cor}[t] \leftarrow \text{preto};$

$\triangleright f[t] \leftarrow 16;$



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ **faça**

$\text{cor}[u] \leftarrow \text{branco};$

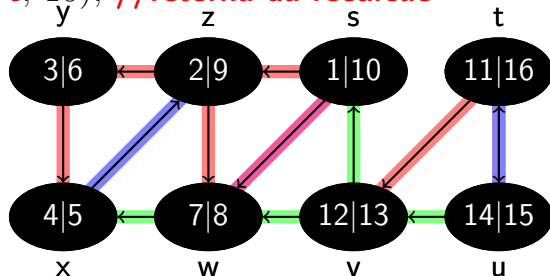
$\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0;$

Para cada $t \in V[G]$ **faça**

Se $\text{cor}[t] = \text{branco}$ **então**

$\triangleleft \text{DFS-VISIT}(G, V, t, 16);$ **//retorna da recursão**



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ faça

$\text{cor}[u] \leftarrow \text{branco};$

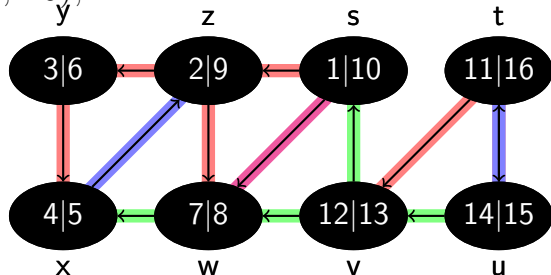
$\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0;$

▷ **Para cada $u \in V[G]$ faça**

Se $\text{cor}[u] = \text{branco}$ então

 DFS-VISIT($G, V, u, 16$);



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ faça

$\text{cor}[u] \leftarrow \text{branco};$

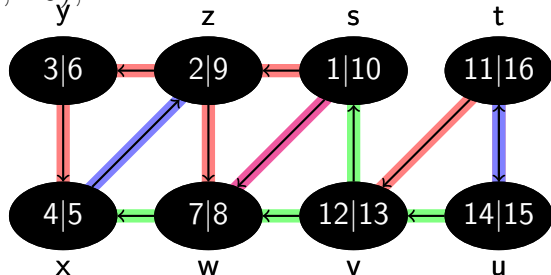
$\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0;$

Para cada $u \in V[G]$ faça

\triangleright **Se $\text{cor}[u] = \text{branco}$ então**

 DFS-VISIT($G, V, u, 16$);



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ **faça**

$\text{cor}[u] \leftarrow \text{branco};$

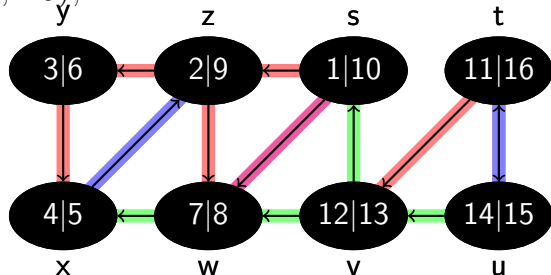
$\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0;$

▷ **Para cada** $v \in V[G]$ **faça**

Se $\text{cor}[v] = \text{branco}$ **então**

 DFS-VISIT($G, V, v, 16$);



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ faça

$\text{cor}[u] \leftarrow \text{branco};$

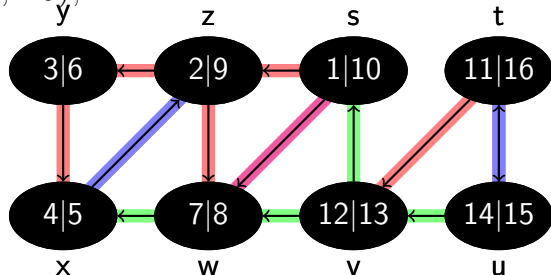
$\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0;$

Para cada $v \in V[G]$ faça

\triangleright **Se $\text{cor}[v] = \text{branco}$ então**

 DFS-VISIT($G, V, v, 16$);



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ faça

$\text{cor}[u] \leftarrow \text{branco};$

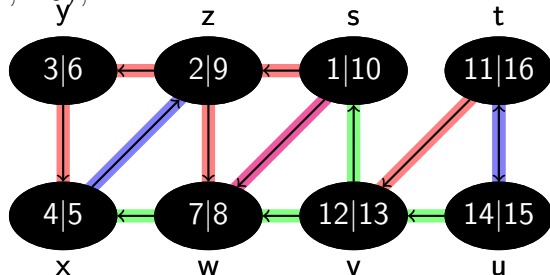
$\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0;$

▷ **Para cada $w \in V[G]$ faça**

Se $\text{cor}[w] = \text{branco}$ **então**

 DFS-VISIT($G, V, w, 16$);



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ faça

$\text{cor}[u] \leftarrow \text{branco};$

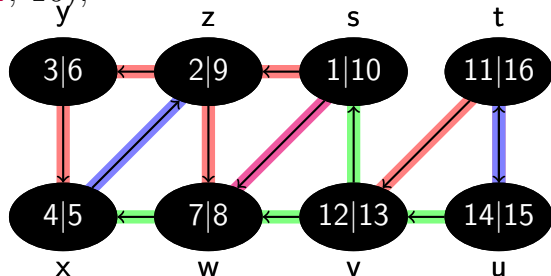
$\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0;$

Para cada $w \in V[G]$ faça

\triangleright **Se $\text{cor}[w] = \text{branco}$ então**

 DFS-VISIT($G, V, w, 16$);



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ **faça**

$\text{cor}[u] \leftarrow \text{branco};$

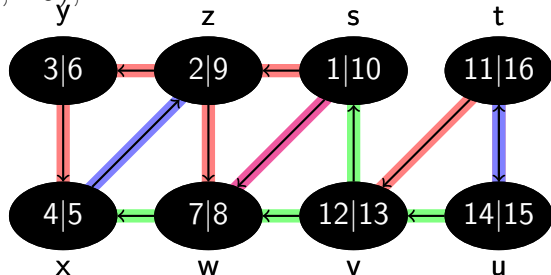
$\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0;$

▷ **Para cada** $x \in V[G]$ **faça**

Se $\text{cor}[x] = \text{branco}$ **então**

 DFS-VISIT($G, V, x, 16$);



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ **faça**

$\text{cor}[u] \leftarrow \text{branco};$

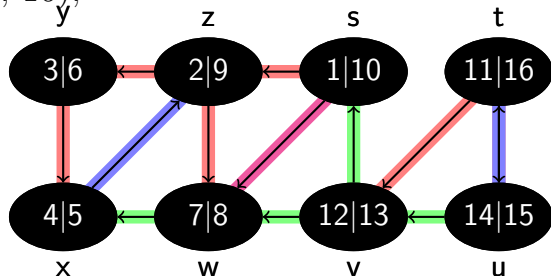
$\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0;$

Para cada $x \in V[G]$ **faça**

\triangleright **Se** $\text{cor}[x] = \text{branco}$ **então**

 DFS-VISIT($G, V, x, 16$);



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ faça

$\text{cor}[u] \leftarrow \text{branco};$

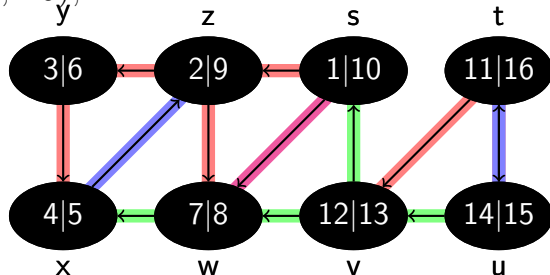
$\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0;$

▷ **Para cada $y \in V[G]$ faça**

Se $\text{cor}[y] = \text{branco}$ **então**

 DFS-VISIT($G, V, y, 16$);



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ **faça**

$\text{cor}[u] \leftarrow \text{branco};$

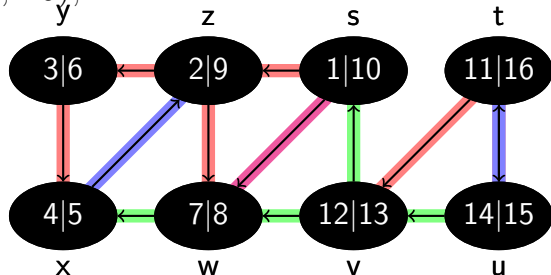
$\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0;$

Para cada $y \in V[G]$ **faça**

\triangleright **Se** $\text{cor}[y] = \text{branco}$ **então**

 DFS-VISIT($G, V, y, 16$);



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ **faça**

$\text{cor}[u] \leftarrow \text{branco};$

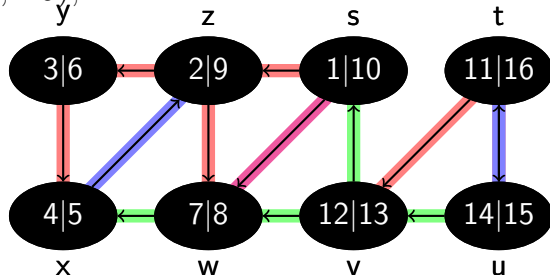
$\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0;$

▷ **Para cada** $z \in V[G]$ **faça**

Se $\text{cor}[z] = \text{branco}$ **então**

 DFS-VISIT($G, V, z, 16$);



BUSCA-EM-PROFUNDIDADE (G)

Para cada $u \in V[G]$ **faça**

$\text{cor}[u] \leftarrow \text{branco};$

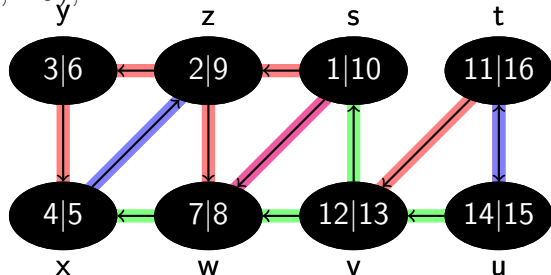
$\pi[u] \leftarrow \text{NIL};$

tempo $\leftarrow 0;$

Para cada $z \in V[G]$ **faça**

\triangleright **Se** $\text{cor}[z] = \text{branco}$ **então**

 DFS-VISIT($G, V, z, 16$);



Complexidade

BUSCA-EM-PROFUNDIDADE (G)

1. **Para cada** $u \in V[G]$ **faça**
2. $\text{cor}[u] \leftarrow \text{branco};$
3. $\pi[u] \leftarrow \text{NIL};$
4. $\text{tempo} \leftarrow 0;$
5. **Para cada** $u \in V[G]$ **faça**
6. **Se** $\text{cor}[u] = \text{branco}$ **então**
7. DFS-VISIT (G, V, u, tempo);

Linhas 1–3: $\mathcal{O}(V)$

Linhas 5–7: V chamadas a DFS-visit.

Complexidade

DFS-VISIT ($G, V, u, tempo$)

1. $cor[u] \leftarrow \text{cinza};$
2. $tempo \leftarrow ++tempo;$
3. $d[u] \leftarrow tempo;$
4. **Para cada** $v \in Adj[u]$ **faça**
5. **Se** $cor[v] = \text{branco}$ **então**
6. $\pi[v] \leftarrow u;$
7. DFS-VISIT ($G, V, v, tempo$);
8. $cor[u] \leftarrow \text{preto};$
9. $f[u] \leftarrow tempo \leftarrow ++tempo;$

DFS-visit(v) é executado exatamente uma vez para cada $v \in V$.

Complexidade

DFS-VISIT ($G, V, u, tempo$)

1. $cor[u] \leftarrow \text{cinza};$
2. $tempo \leftarrow ++tempo;$
3. $d[u] \leftarrow tempo;$
4. **Para cada** $v \in Adj[u]$ **faça**
5. **Se** $cor[v] = \text{branco}$ **então**
6. $\pi[v] \leftarrow u;$
7. DFS-VISIT ($G, V, v, tempo$);
8. $cor[u] \leftarrow \text{preto};$
9. $f[u] \leftarrow tempo \leftarrow ++tempo;$

Em uma execução de DFS-visit(v), o laço das linhas 4-7 é executado $|Adj[u]|$ vezes.

Complexidade

DFS-VISIT ($G, V, u, tempo$)

1. $cor[u] \leftarrow \text{cinza};$
2. $tempo \leftarrow ++tempo;$
3. $d[u] \leftarrow tempo;$
4. **Para cada** $v \in Adj[u]$ **faça**
5. **Se** $cor[v] = \text{branco}$ **então**
6. $\pi[v] \leftarrow u;$
7. DFS-VISIT ($G, V, v, tempo$);
8. $cor[u] \leftarrow \text{preto};$
9. $f[u] \leftarrow tempo \leftarrow ++tempo;$

Assim, o custo total de todas as chamadas é $\sum_{v \in V} |Adj(v)| = \Theta(E)$.

Complexidade

DFS-VISIT ($G, V, \mathbf{u}, tempo$)

1. $cor[\mathbf{u}] \leftarrow \text{cinza};$
2. $tempo \leftarrow ++tempo;$
3. $d[\mathbf{u}] \leftarrow tempo;$
4. **Para cada** $\mathbf{v} \in Adj[\mathbf{u}]$ **faça**
5. **Se** $cor[\mathbf{v}] = \text{branco}$ **então**
6. $\pi[\mathbf{v}] \leftarrow \mathbf{u};$
7. DFS-VISIT ($G, V, \mathbf{v}, tempo$);
8. $cor[\mathbf{u}] \leftarrow \text{preto};$
9. $f[\mathbf{u}] \leftarrow tempo \leftarrow ++tempo;$

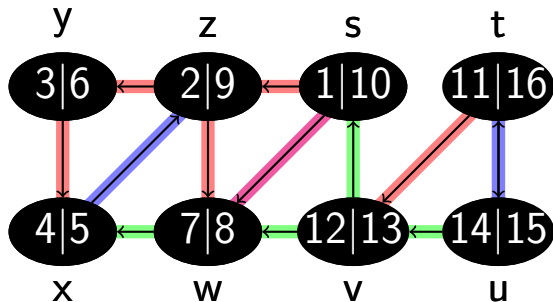
Conclusão: a complexidade de tempo de Busca-em-profundidade é $\mathcal{O}(V + E)$.

Busca em profundidade

A busca em profundidade pode ser usada para **classificar as arestas** de um grafo $G = (V, E)$. Ela classifica as arestas em quatro tipos.

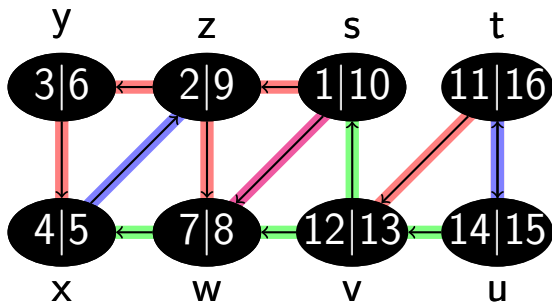
Busca em profundidade

Arestas da árvore (**vermelhas**): arestas que pertencem à floresta da busca em profundidade.



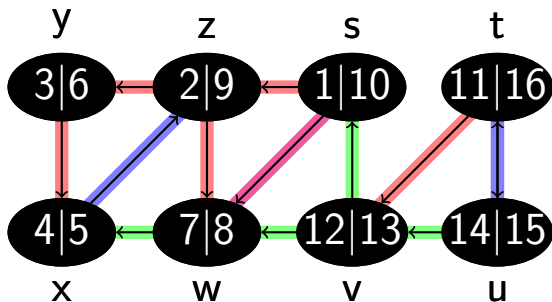
Busca em profundidade

Arestas de retorno (azuis): arestas (u, v) ligando um vértice u a um ancestral v na árvore de busca em profundidade.



Busca em profundidade

Arestas de avanço (roxas): arestas (u, v) ligando um vértice u a um descendente próprio v na árvore de busca em profundidade.



Busca em profundidade

Arestas de cruzamento (**verdes**): todas as outras arestas.

