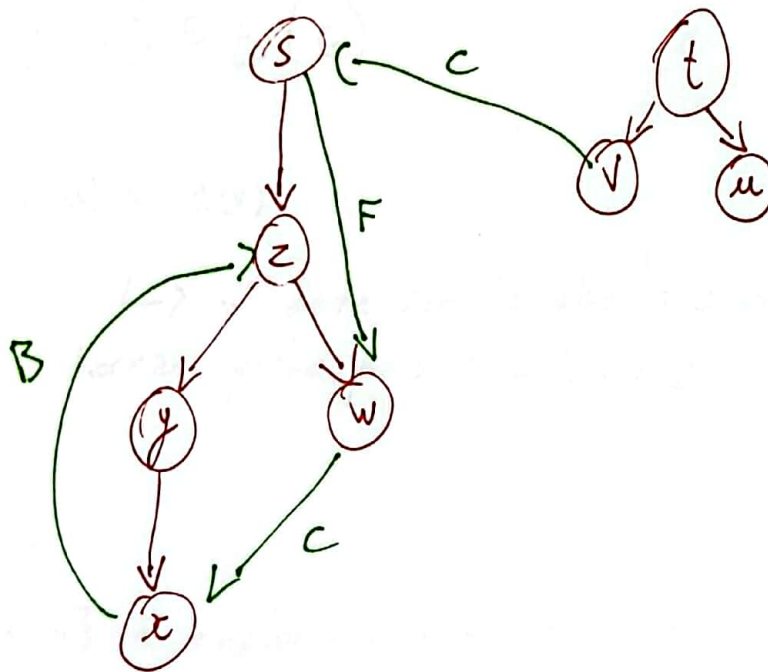


# Classificação das Arestas na DFS

- Arestas de Árvore
- Back-edges
- Forward-edges
- Cross-edges



Exercício: Modificar a DFS para classificar as arestas quando são exploradas

$(u, v)$

$v$  branco  $\rightarrow$  aresta de árvore

$v$  cinza  $\rightarrow$  back

$v$  preto  $\left\{ \begin{array}{l} d(u) < d(v) \text{ forward} \\ d(u) > d(v) \text{ cross} \end{array} \right.$

Exercícios 22.3.4

22.3.5, 22.3.7, 22.3.8

22.3.10

## Teorema 22.10

Em uma DFS em  $G = (V, E)$  toda aresta de  $G$  é ou de árvore ou uma back-edge

## Demonstração

$$\{u, v\} \in E(G)$$

$$d(u) < d(v)$$

$\hookrightarrow v$  deve ser descoberto e explorado antes de  $u$  se tornar preto, pois  $v \in \text{Adj}(u)$

1º caso

$\{u, v\}$  é explorado 1º no sentido  $u \rightarrow v$

$v$  é branco

$u$  é cinza

$\{u, v\}$  é de árvore

2º caso

$\{u, v\}$  é explorado 1º no sentido  $v \rightarrow u$

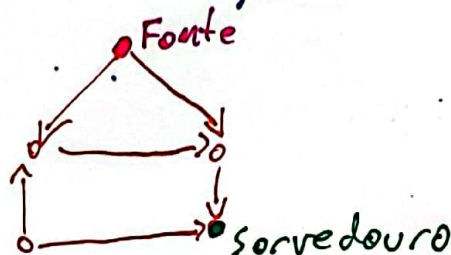
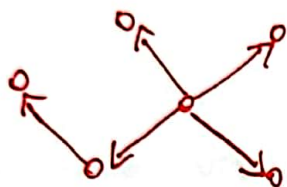
$u$  é cinza

$v$  é cinza

$\{u, v\}$  é back

# Ordenação Topológica

DAG  $\rightarrow$  Directed Acyclic Graph

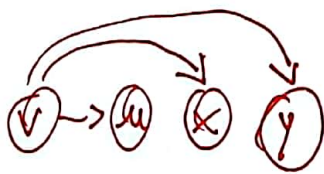
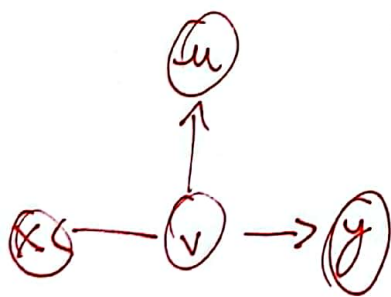


Fonte: Vértice que possui grau de entrada zero

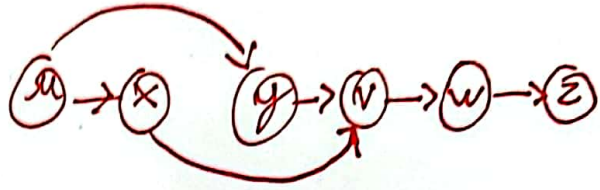
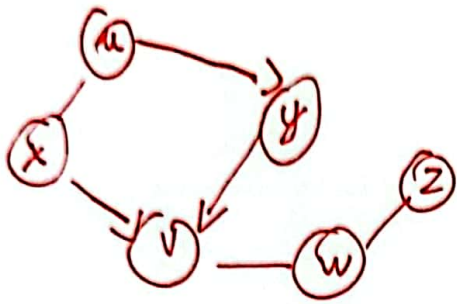
Sorvedouro: Vértice que possui grau de saída zero

Toda DAG tem pelo menos uma fonte ~~e~~ sorvedouro e um

Uma ordenação topológica de um dag  $G=(V,E)$  é uma ordenação linear dos seus vértices de maneira que se existe  $(u,v)$ , então  $(u)$  aparece antes de  $(v)$  na ordem



$u, x$  e  $y$   
podem inverter  
no exemplo  
acima, apenas  
 $v$  possui ordem dependente



### Algoritmo via Fonte

1. Escolha uma fonte  $v$
2. Enfileire  $v$
3. Remova  $v$  de  $G$
4. Se  $G \neq \emptyset$ , volte ao passo 1
5. Devolva a fila

### Algoritmo via Sorvedouro

1. Escolha um sorvedouro  $v$
2. Enfileire  $v$
3. Remova  $v$  de  $G$
4. Se  $V \neq \emptyset$ , volte ao passo 1
5. Devolva o reverso da fila