

---

SCC0216 - Modelagem Computacional em Grafos

# Ordenação Topológica e Componentes Fortemente Conectados

---

Prof.: Rosane Minghim ([rminghim@icmc.usp.br](mailto:rminghim@icmc.usp.br))

1º sem. 2014

PAE: Rafael Martins ([rmartins@icmc.usp.br](mailto:rmartins@icmc.usp.br))  
Bilzã Marques ([bmarques@icmc.usp.br](mailto:bmarques@icmc.usp.br))

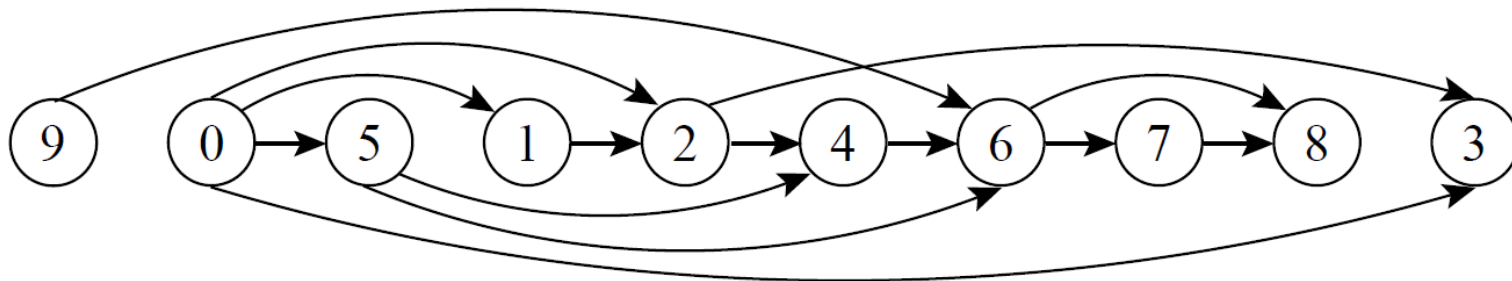
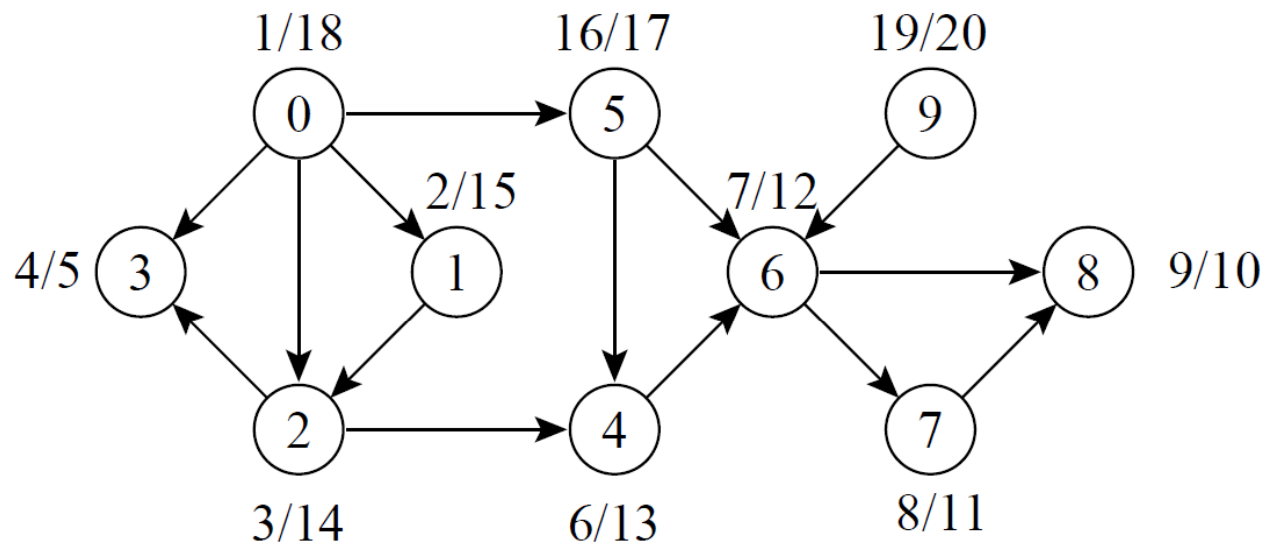
Baseado no material de aula original: Prof<sup>a</sup>. Josiane M. Bueno

---

# Ordenação Topológica

- Define-se Ordenação Topológica para Grafos orientados acíclicos.
- O objetivo da ordenação topológica é alinhar todos os vértices de um grafo em sequência, de forma que se a aresta  $(u, v)$  pertence a  $V$ , então  $u$  está antes de  $v$  na sequência

# Ordenação Topológica – Exemplo (Ziviani 2004)



---

# Ordenação Topológica (algoritmo)

1. Chame DFS para todos os vértices do grafo  $G$  (isto é, enquanto existirem vértices 'brancos').
2. A cada vértice que é terminado (isto é, que se torna 'preto'), insira-o na cabeça de uma lista encadeada.
3. Retorna a lista encadeada de vértices do grafo produzida no passo (2)

---

# Ordenação Topológica (algoritmo)

- A implementação da ordenação topológica se dá adicionando um comando:

```
Inserer_primeiro(u, L:lista)
```

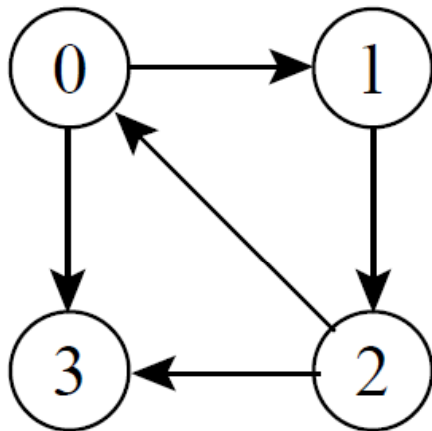
- Inserção na cabeça da lista  $L$ , na posição do algoritmo DFS logo após a determinação do tempo  $t[u]$  e da finalização do nó, isto é, após o momento em que ele se torna 'preto'.
  - Obs: naturalmente `Inicializa(L)` precisa ser chamada no início do algoritmo que chama DFS para todos os vértices 'brancos'.

---

# Componentes Fortemente Conectados

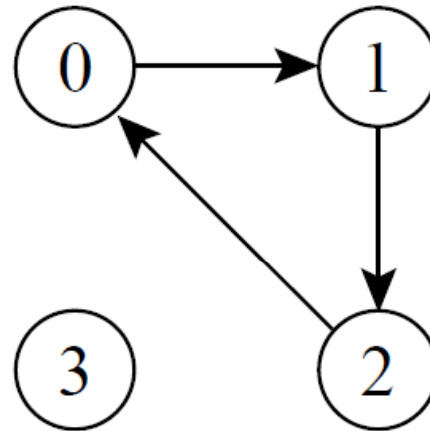
- Define-se *componentes fortemente conectados* para um grafo orientado.
- Um *Componente Fortemente Conectado* (ou *Fortemente Conexo*)  $C$  de um grafo  $G$  é um conjunto de vértices maximal de  $G$  de forma que para todos os vértices  $u$  e  $v$  em  $C$ ,  $u$  é alcançável a partir de  $v$  e  $v$  é alcançável a partir de  $u$ .

# Componentes Fortemente Conectados (Exemplo – Ziviani 2004)



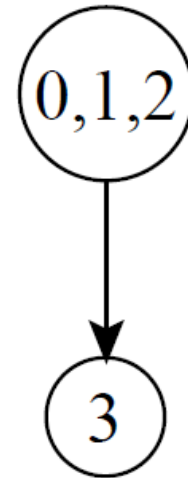
(a)

(a) Grafo original



(b)

(b) Componentes Conexas



(c)

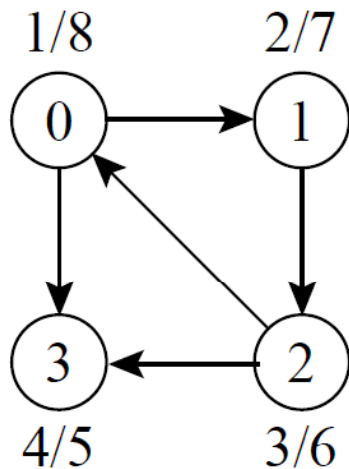
(c) Colapso dos  
vértices das componentes

# Componentes Fortemente Conectados (algoritmo)

1. Chama  $\text{DFS}(G)$  para obter os tempos de término  $t[u]$  para todos os vértices de  $G$ , isto é, enquanto existirem vértices ‘brancos’ em  $G$ .
2. Obtém  $G^T$ .
3. Chama  $\text{DFS}(G^T)$  em ordem decrescente de  $t[u]$  obtido no passo (1), enquanto existirem vértices  $u$  ‘brancos’ em  $G^T$ .
4. Retorne todas as árvores obtidas no passo (3).

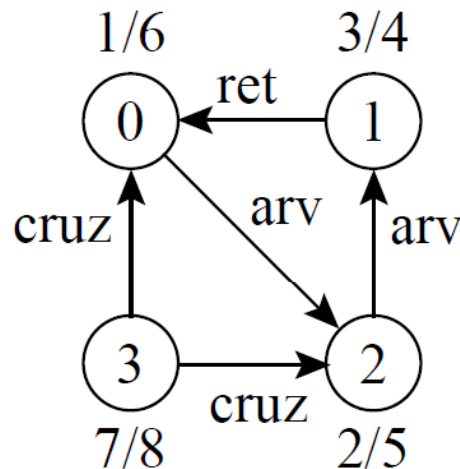


# Componentes Fortemente Conectados (Exemplo – Ziviani 2004)



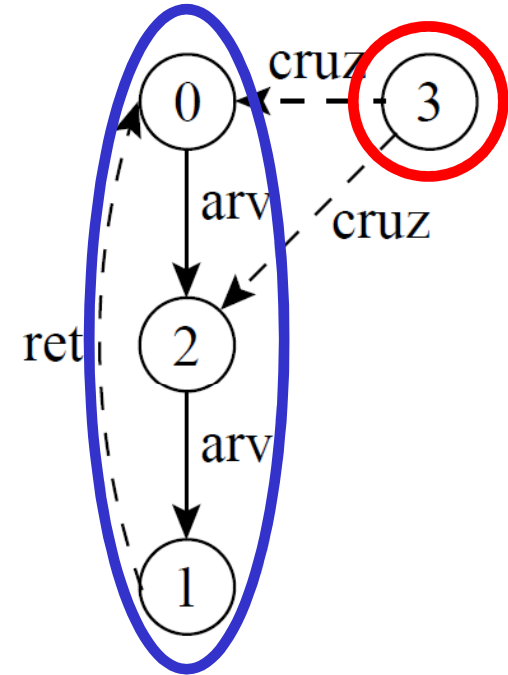
(a)

(a) Grafo original  
com resultado da DFS



(b)

(b) Grafo transposto  
com resultado da DFS



(c)

(c) Árvores encontradas  
e CFCs