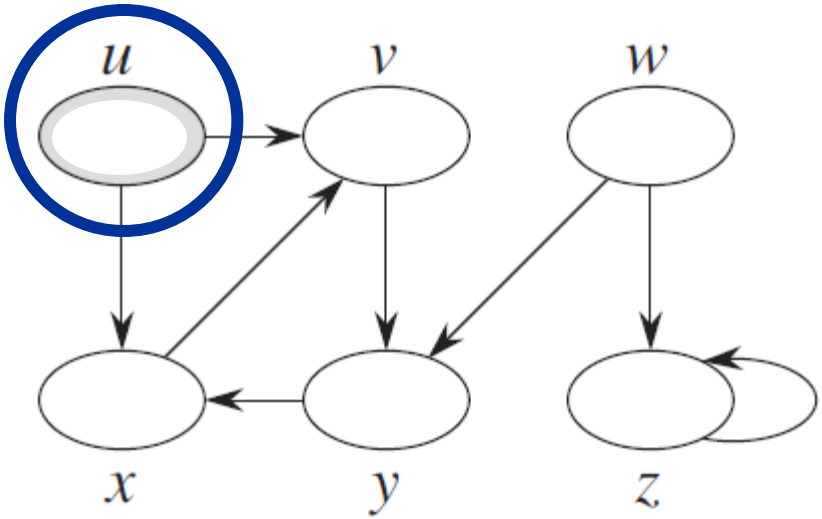
# Introdução à Teoria dos Grafos

Prof. Alexandre Noma

# DFS (profundidade)

- Ideia: "descoberta" e "finalização"

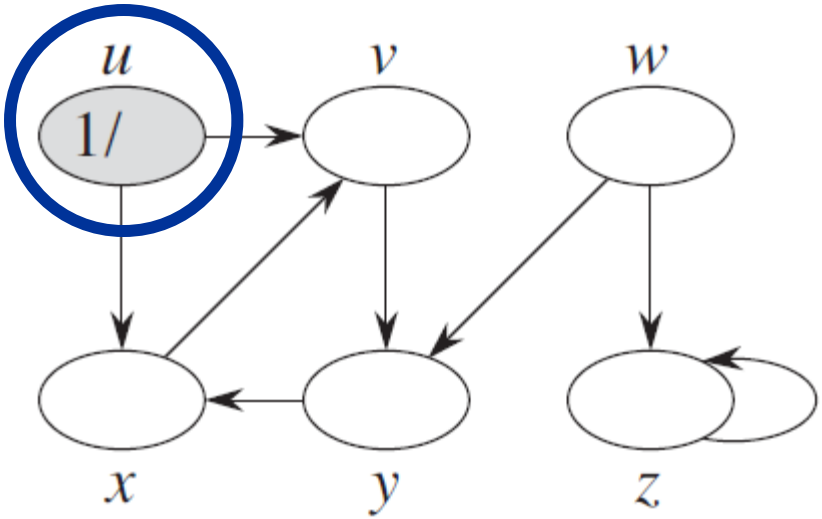



(Pilha)

**S** =

# DFS (profundidade)

- Ideia: "descoberta" e "finalização"

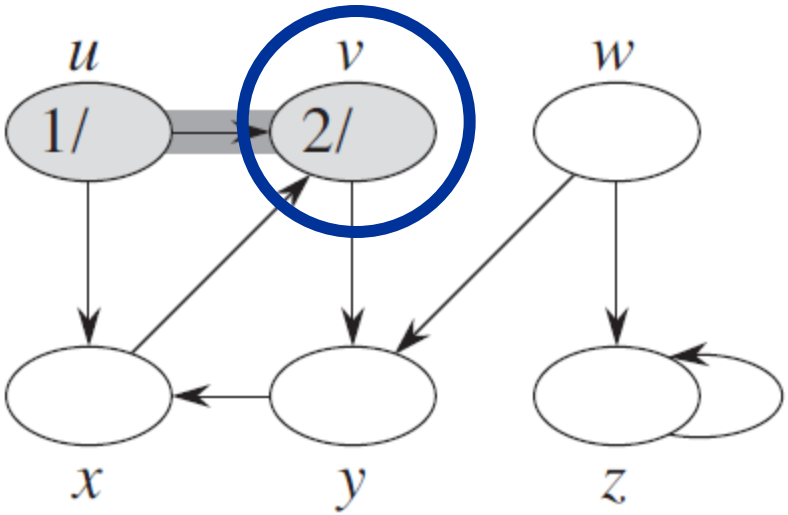



(Pilha)

**S** = u

# DFS (profundidade)

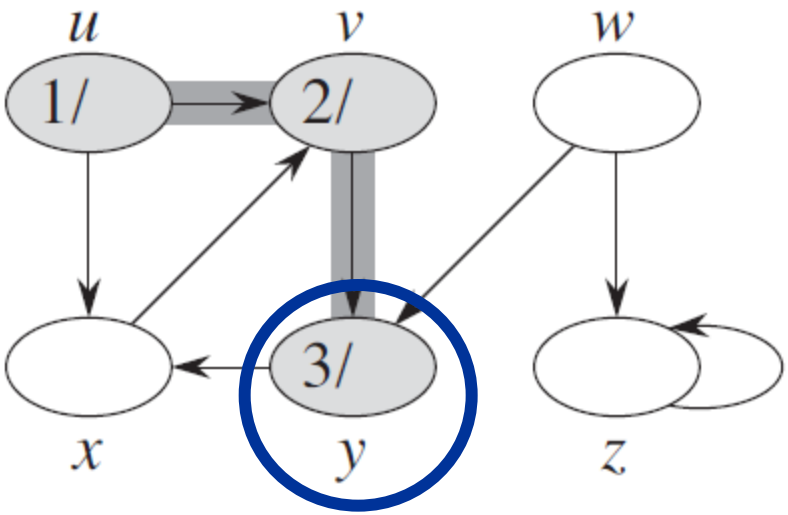
- Ideia: "descoberta" e "finalização"




**S** = u, v

# DFS (profundidade)

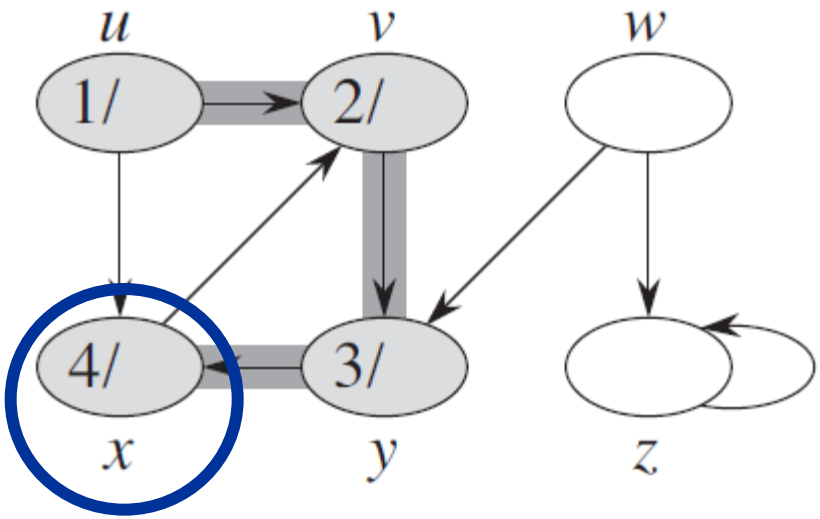
- Ideia: "descoberta" e "finalização"




**S** = u, v, y

# DFS (profundidade)

- Ideia: "descoberta" e "finalização"

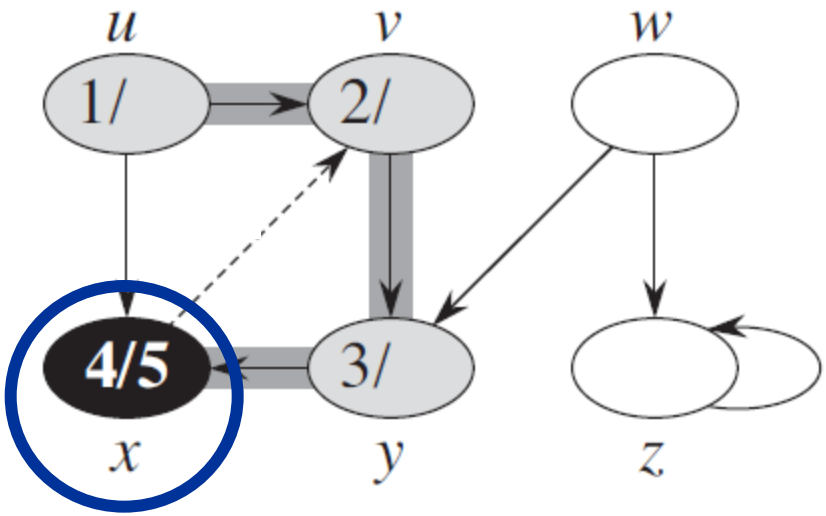



Topo

**S** = u, v, y, x

# DFS (profundidade)

- Ideia: "descoberta" e "finalização"

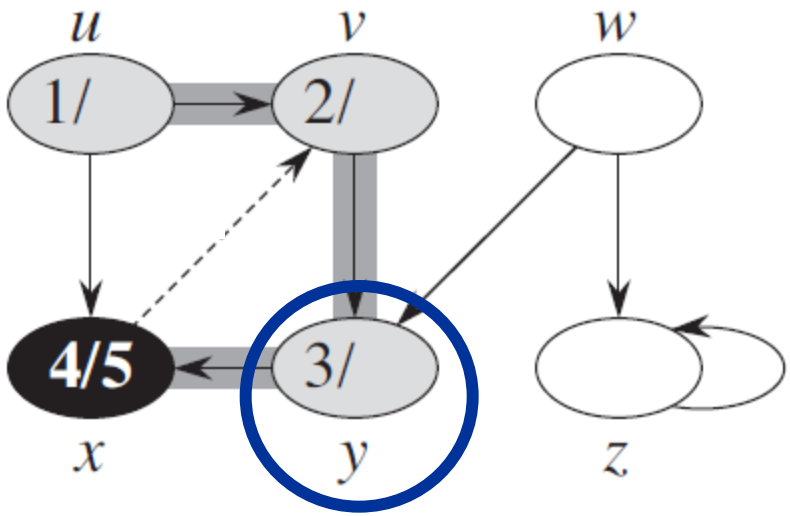



Topo

**S** = u, v, y, ~~x~~

# DFS (profundidade)

- Ideia: "descoberta" e "finalização"

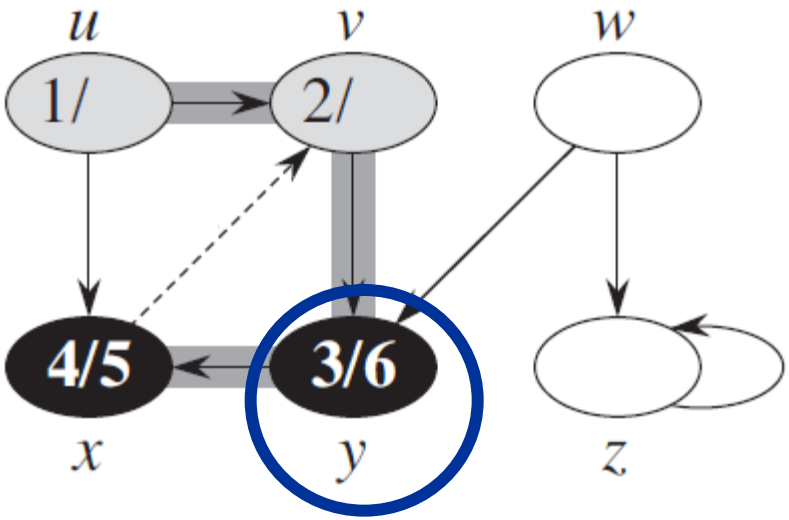



Topo

**S** = u, v, y

# DFS (profundidade)

- Ideia: "descoberta" e "finalização"

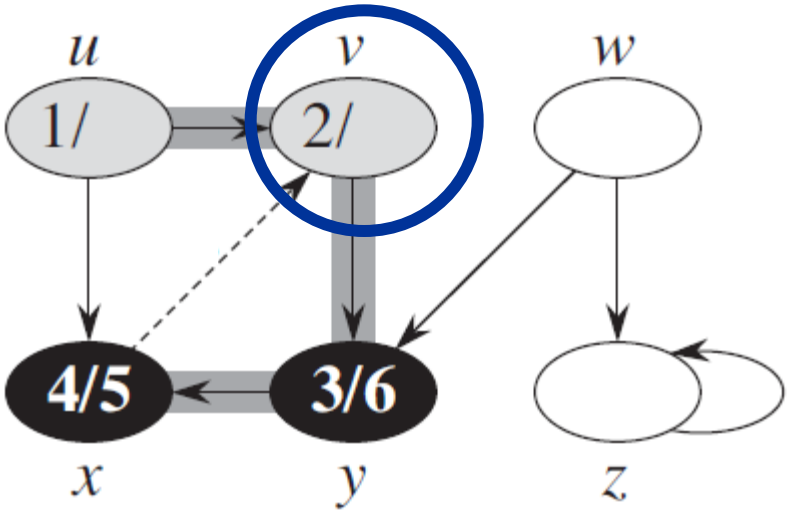



Topo

**S** = u, v, ~~y~~

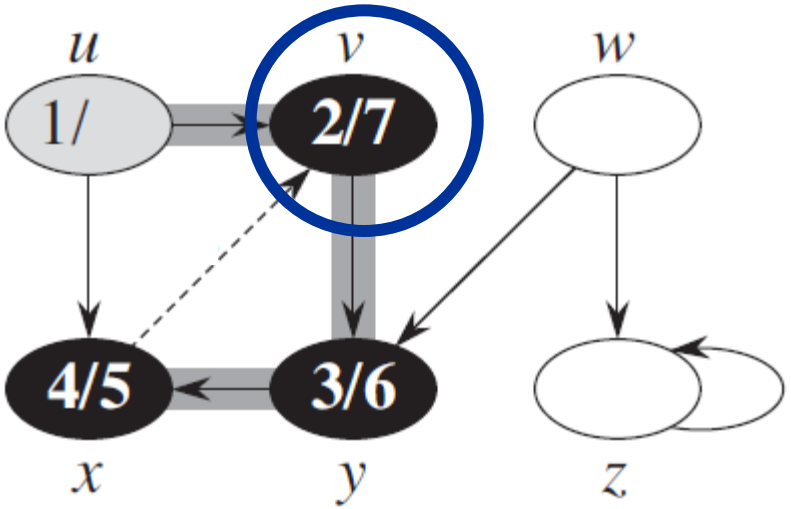
# DFS (profundidade)

- Ideia: "descoberta" e "finalização"



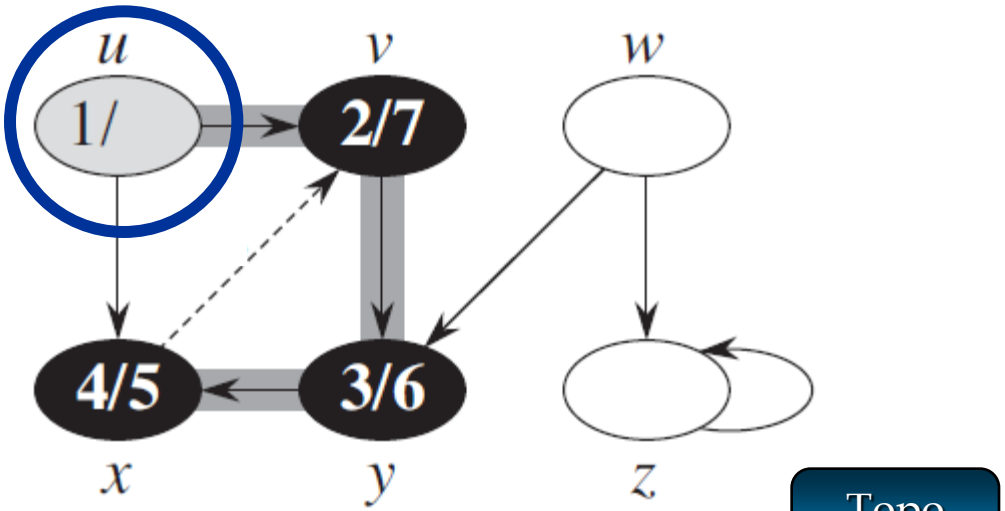
# DFS (profundidade)

- Ideia: "descoberta" e "finalização"



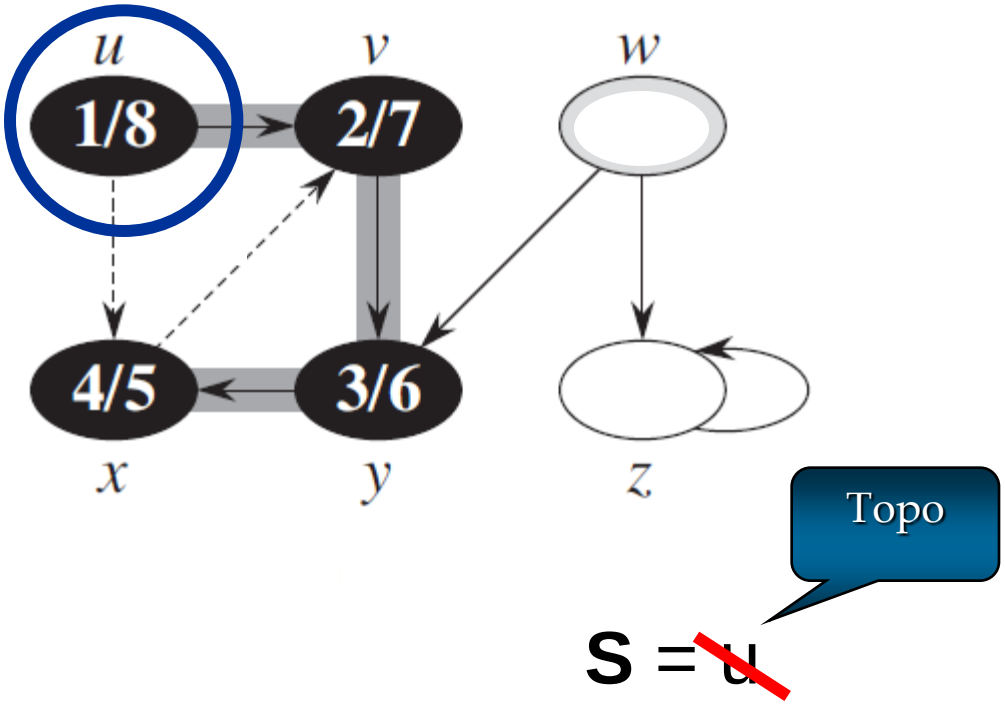
# DFS (profundidade)

- Ideia: "descoberta" e "finalização"



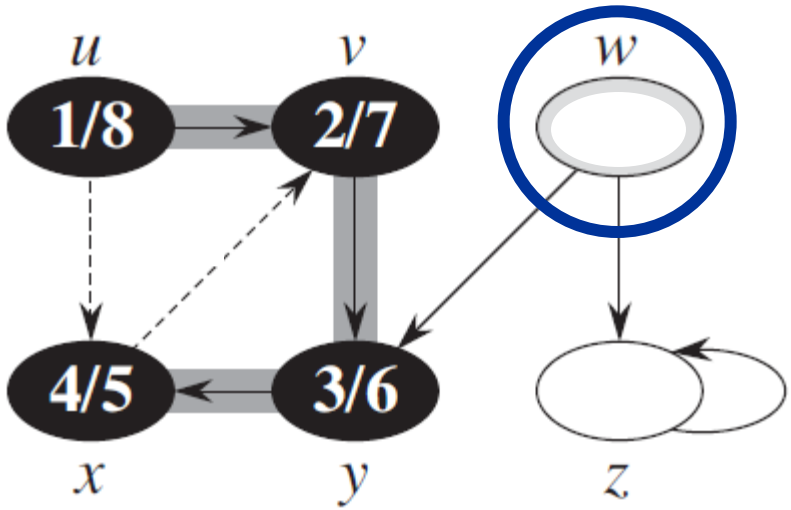
# DFS (profundidade)

- Ideia: "descoberta" e "finalização"



# DFS (profundidade)

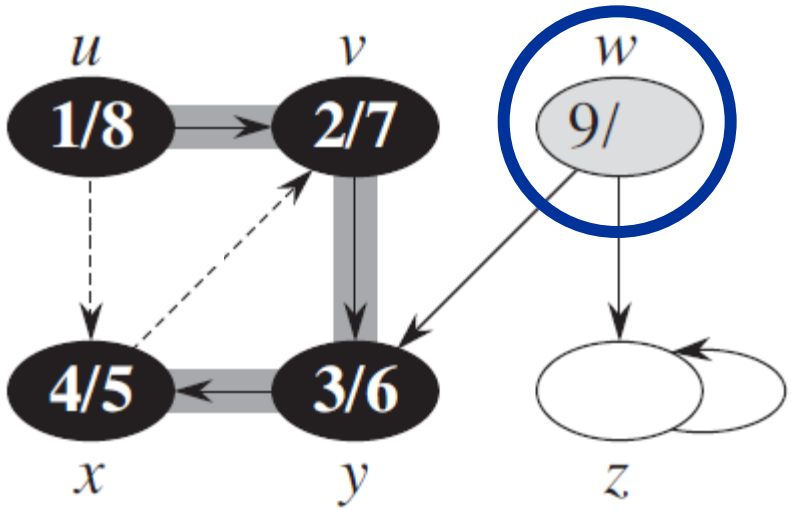
- Ideia: "descoberta" e "finalização"




**S** =

# DFS (profundidade)

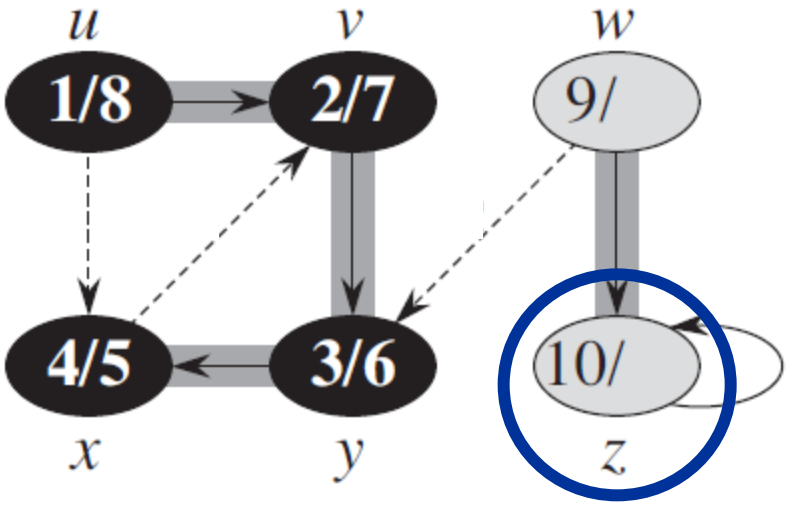
- Ideia: "descoberta" e "finalização"




**S** = w

# DFS (profundidade)

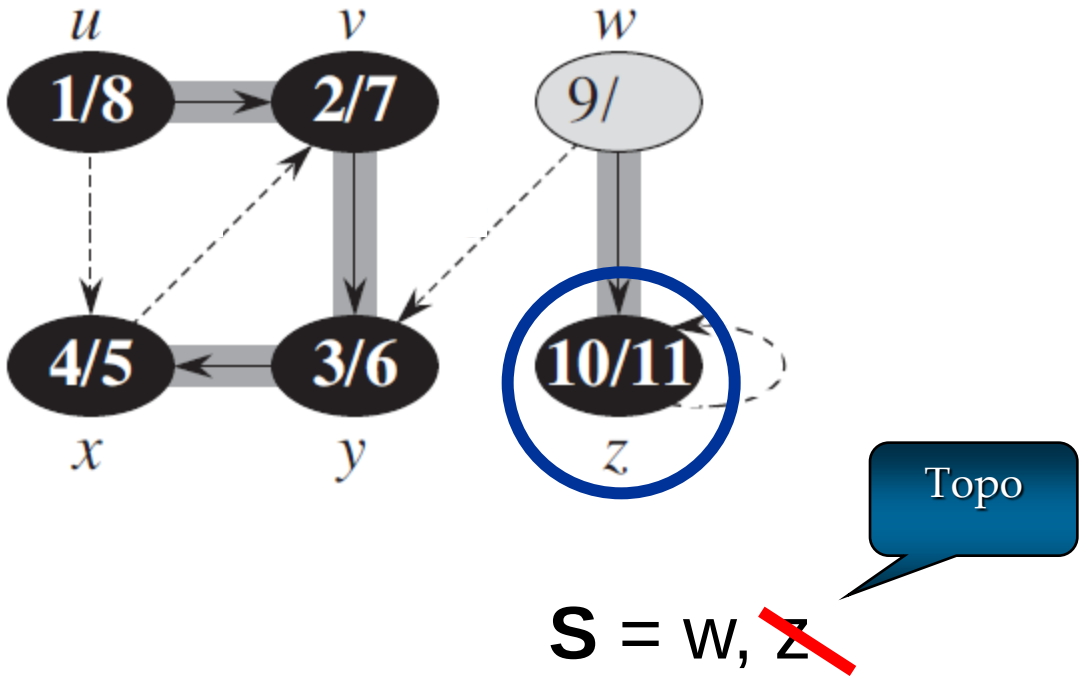
- Ideia: "descoberta" e "finalização"




**S** = w, z

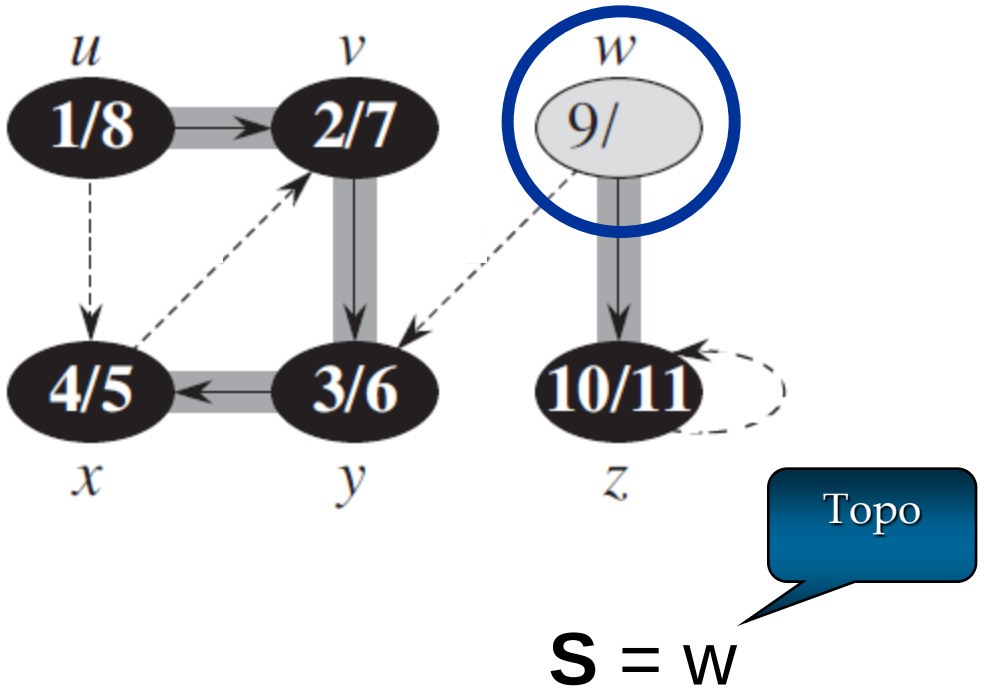
# DFS (profundidade)

- Ideia: "descoberta" e "finalização"



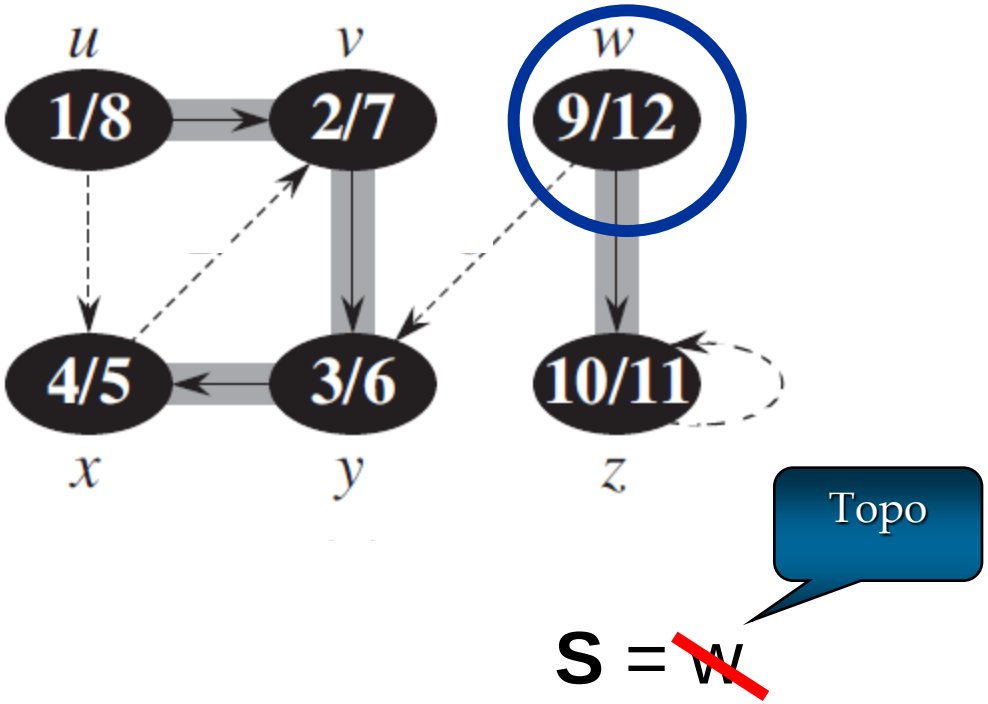
# DFS (profundidade)

- Ideia: "descoberta" e "finalização"



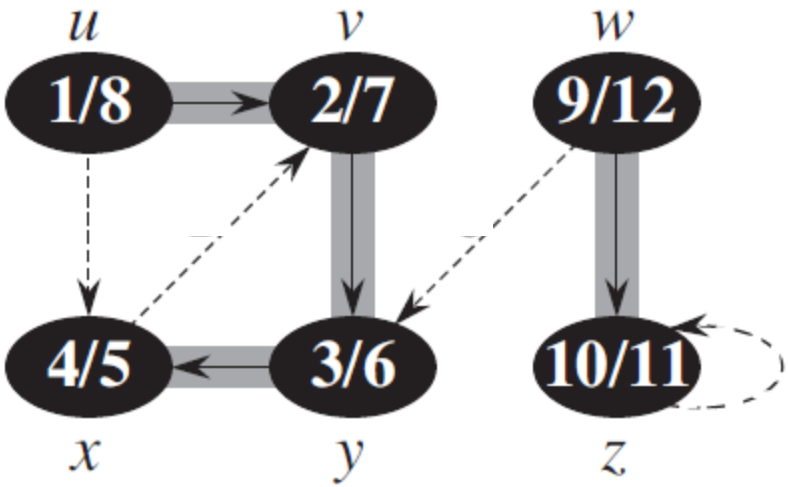
# DFS (profundidade)

- Ideia: "descoberta" e "finalização"



# DFS (profundidade)

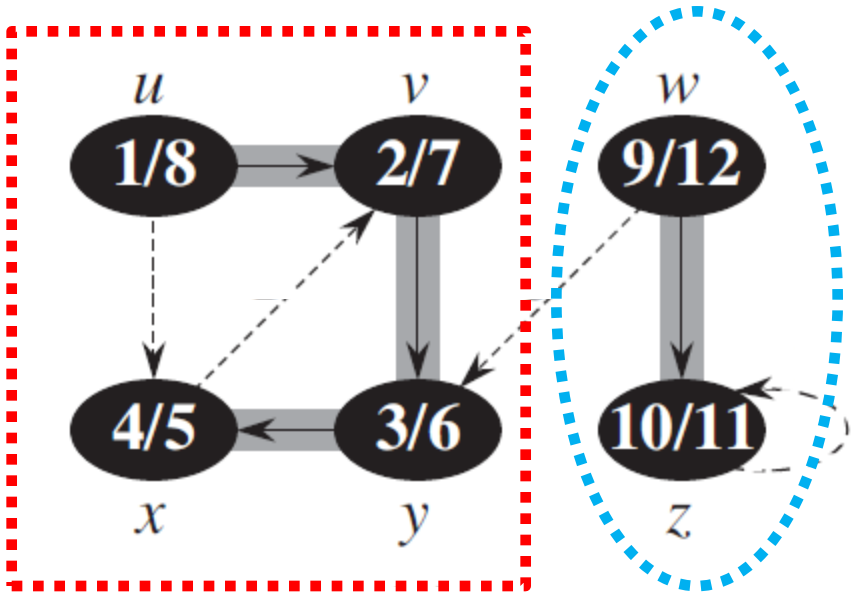
- Expressão com parênteses:




1 2 3 4 5 6 7 8 9 10 11 12

**(u (v ( y (x x) y) v) u) (w (z z) w)**

- Expressão com parênteses:

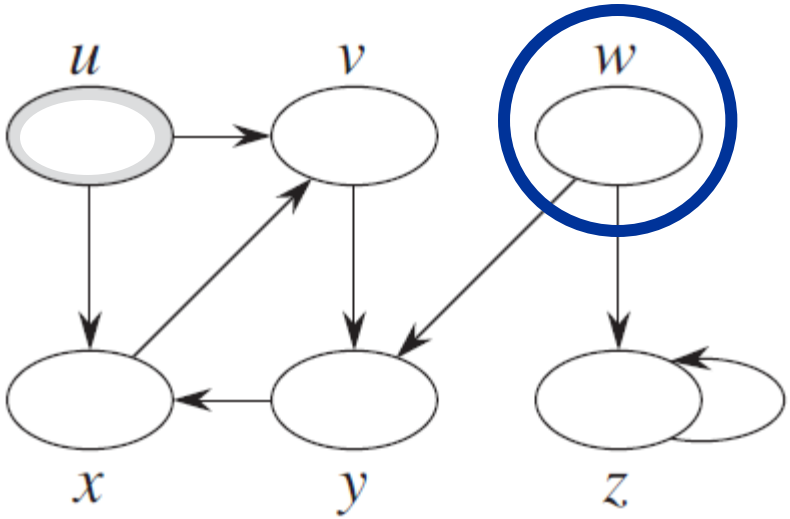



1 2 3 4 5 6 7 8 9 10 11 12

**(u (v ( y (x x) y) v) u) (w (z z) w)**

# DFS (profundidade)

- O que acontece se começarmos por outro vértice?

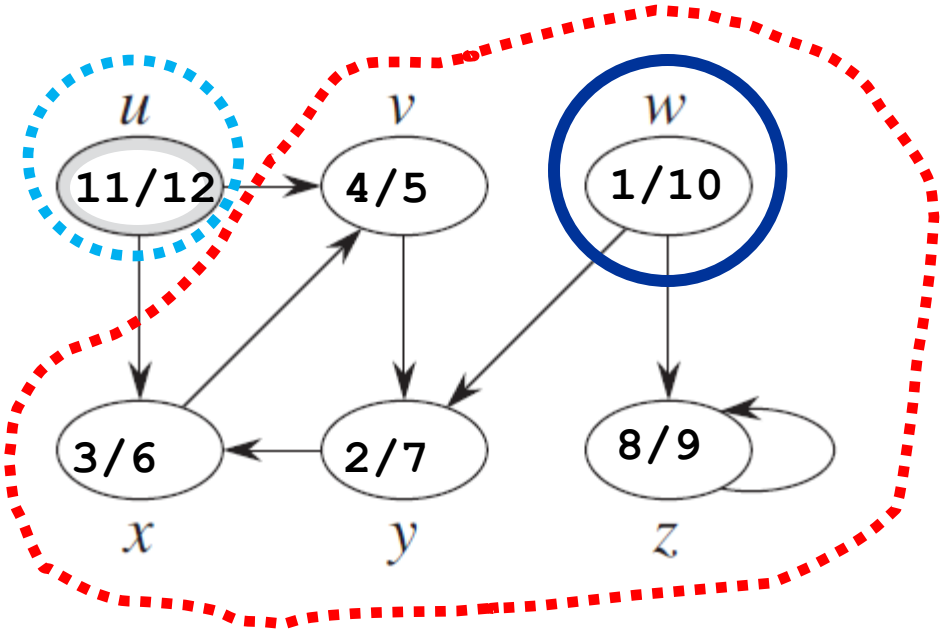



(Pilha)

**S** =

# DFS (profundidade)

- O que acontece se começarmos por outro vértice?



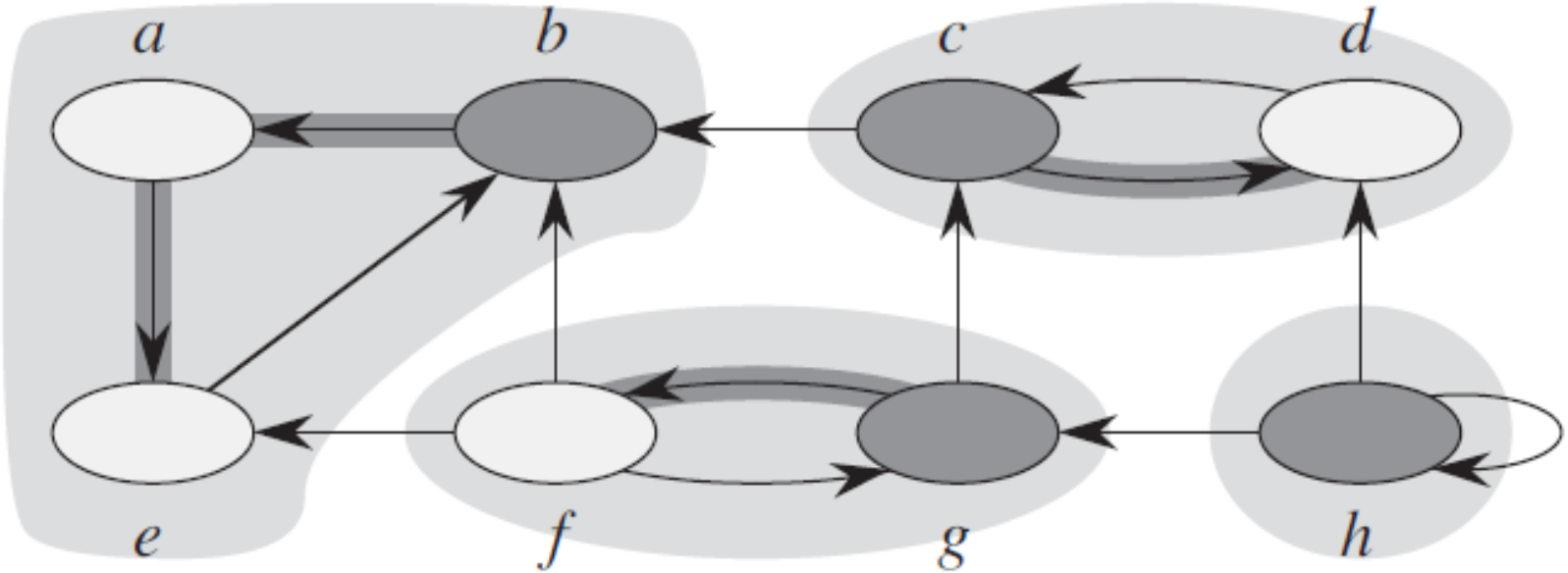

1 2 3 4 5 6 7 8 9 10 11 12

**(w (y (x (v v) x) y) (z z) w) (u u)**

# Hoje

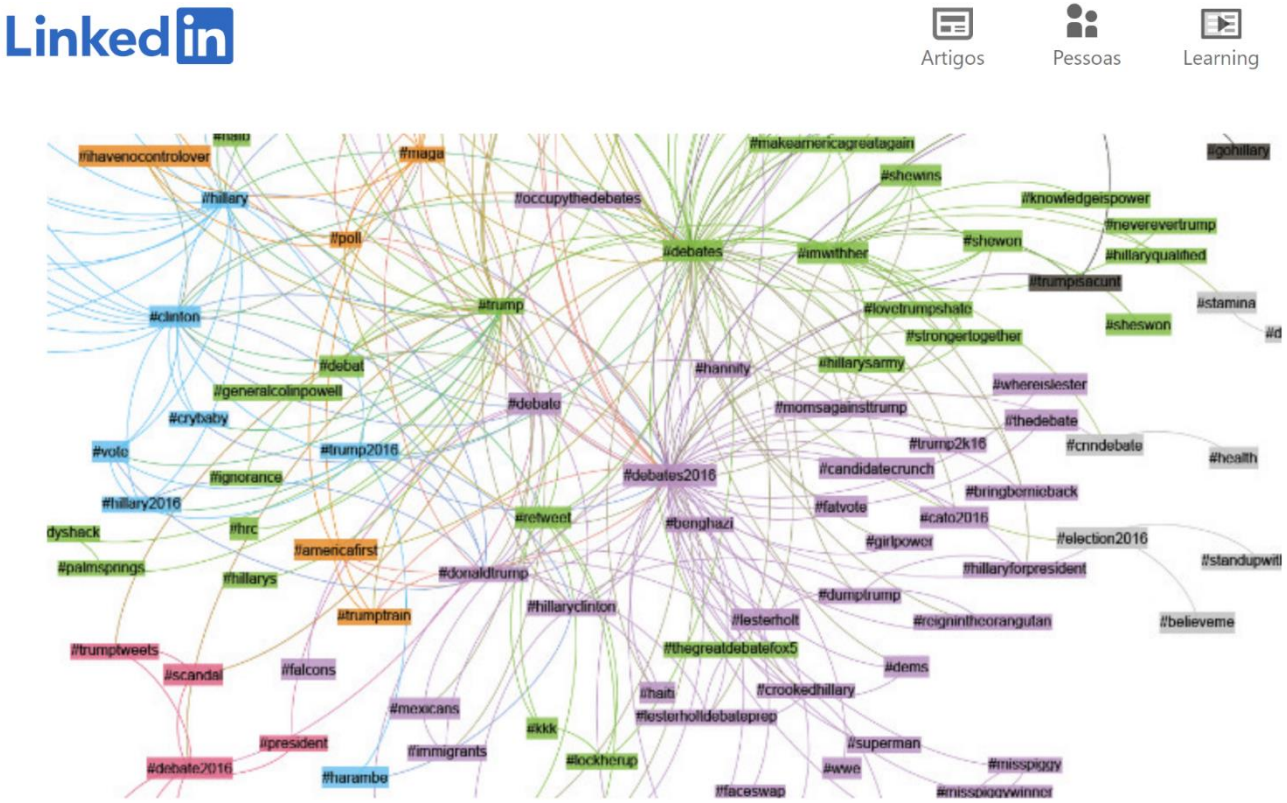
- Cálculo dos componentes fortemente conexos.

- Exemplos?



# Componentes Fortemente Conexos

- Exemplos: redes sociais, pesquisa, WEB (links)



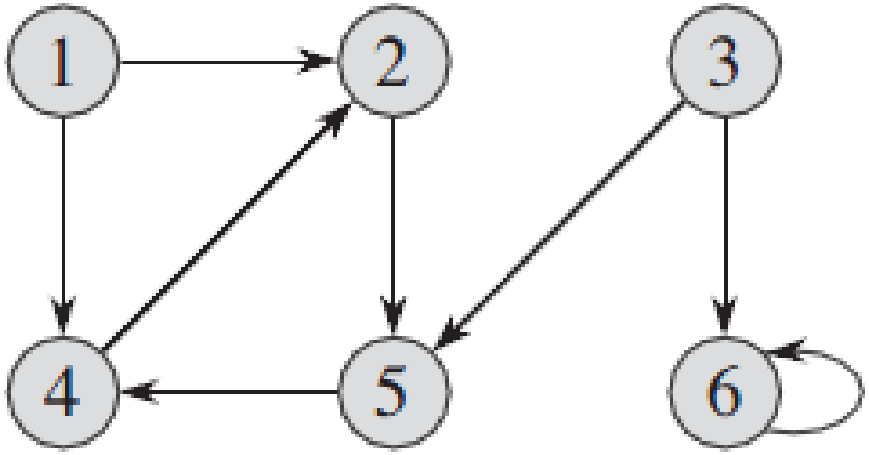
# Hoje

- Algoritmo para calcular componentes “fortes”.

- Pré-requisitos:
  - DFS, parênteses
  - Grafo transposto

# Grafo transposto

- **Exemplo**: Calcule a matriz transposta e desenhe o grafo resultante da transposta.



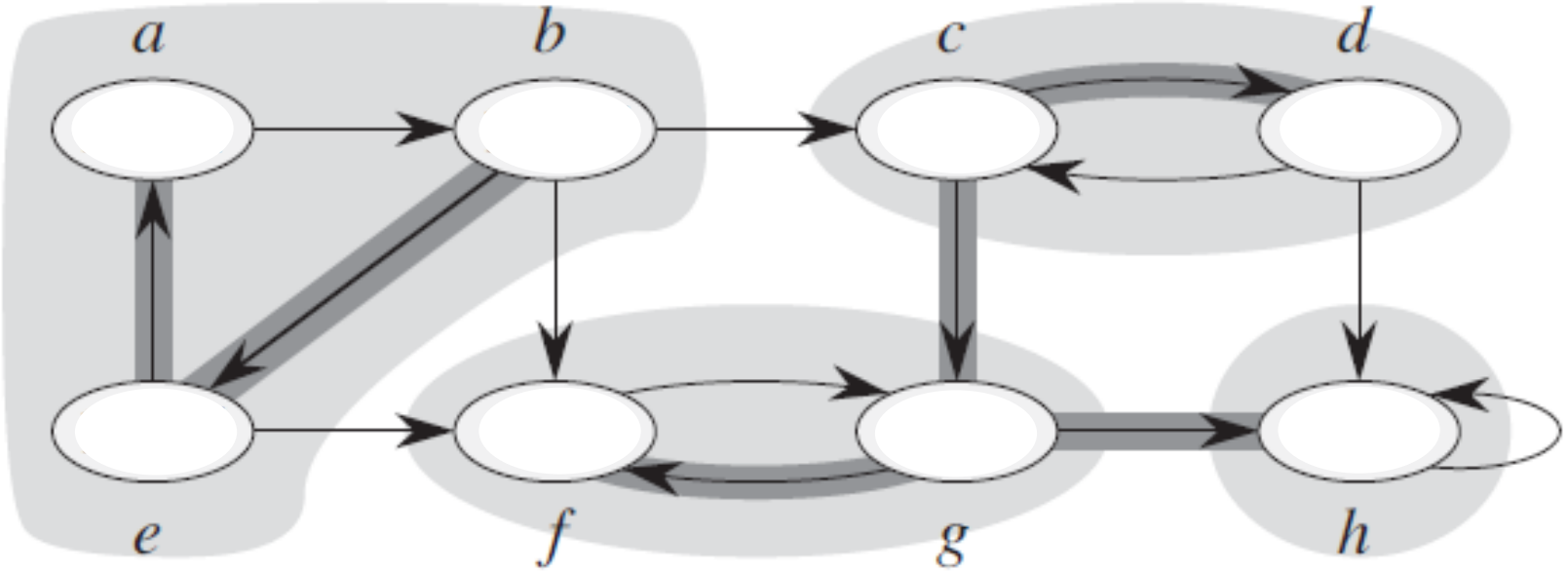

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 0 | 0 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 1 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 1 |

# Exercício Programa

- 06-grafoTransposto.py

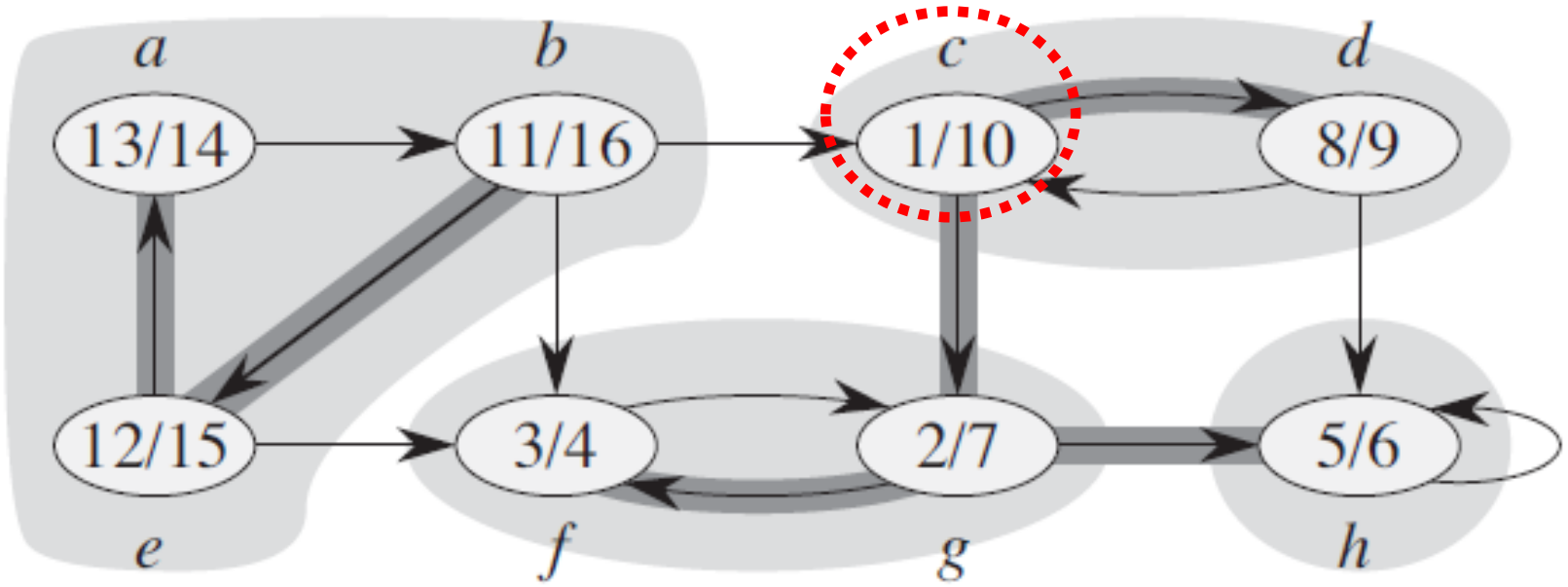
**Strongly-Connected-Components**(G)

1. executar **DFS**(**G**)
   para calcular instantes v.**f**
2. executar **DFS**(**G**$^{T}$), em ordem decrescente de v.**f** do passo 1



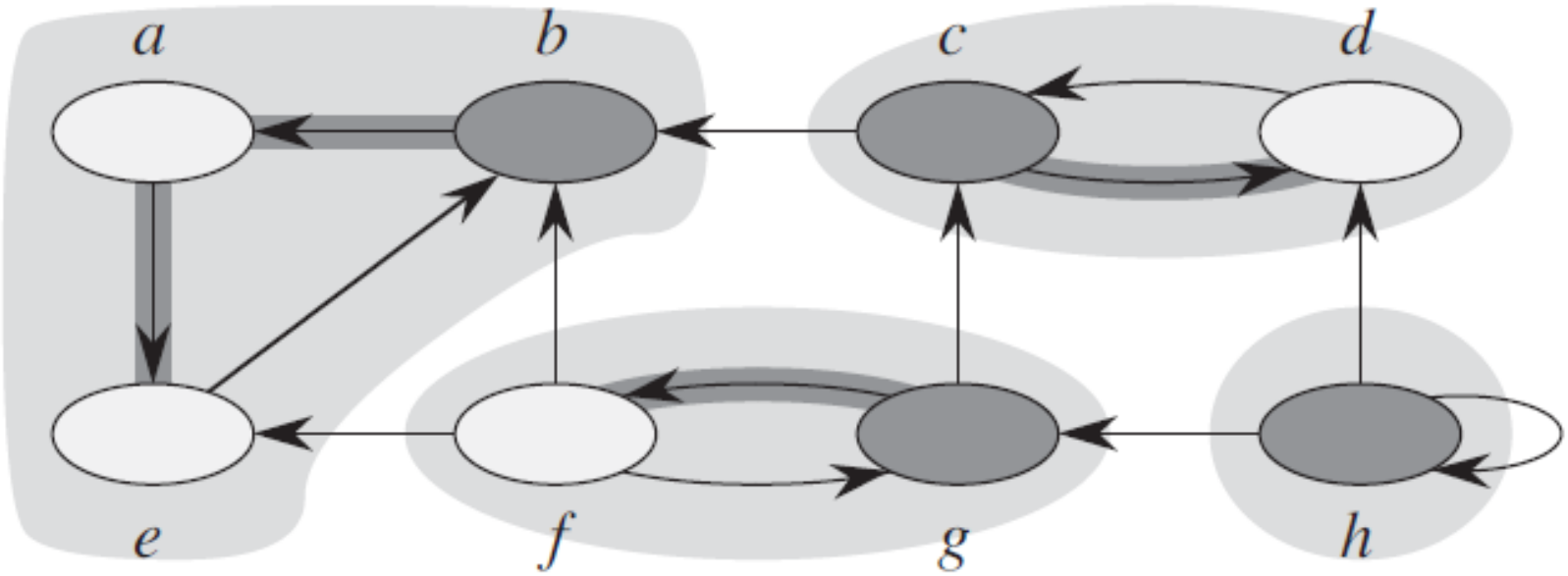
**Strongly-Connected-Components**(G)

1. executar **DFS**(**G**)
   para calcular instantes v.**f**

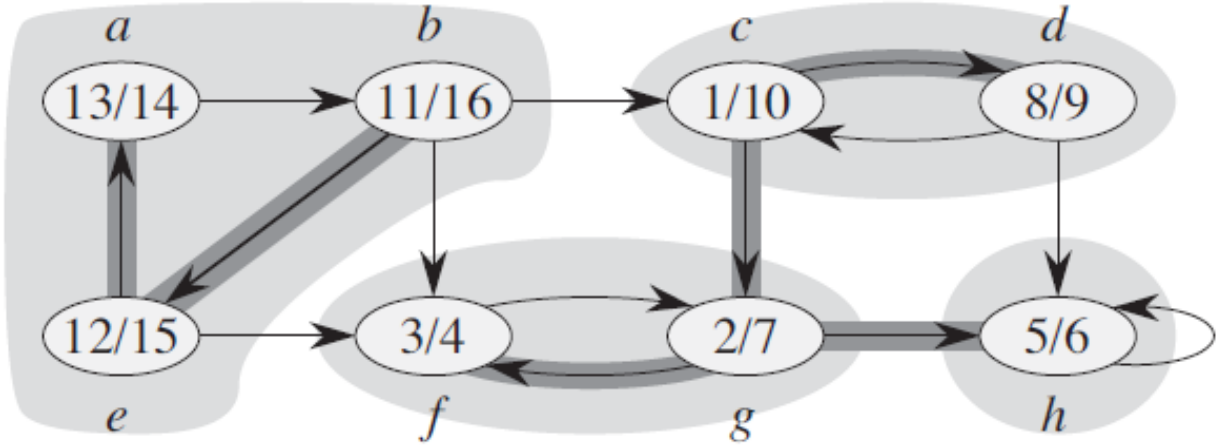


**Strongly-Connected-Components**(G)

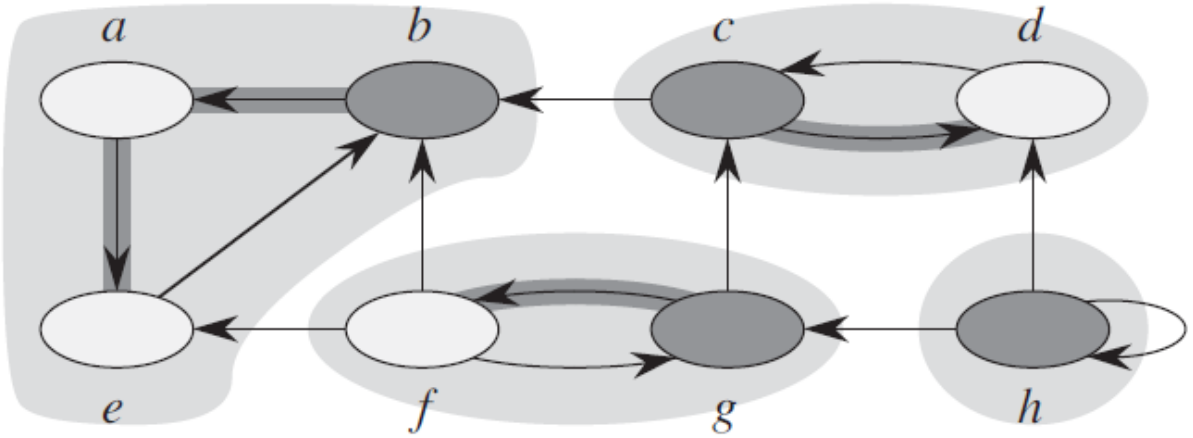
1. executar **DFS**(**G**)
   para calcular instantes v.**f**
2. executar **DFS**(**G**$^{T}$), em ordem decrescente de v.**f** do passo 1



- $G$

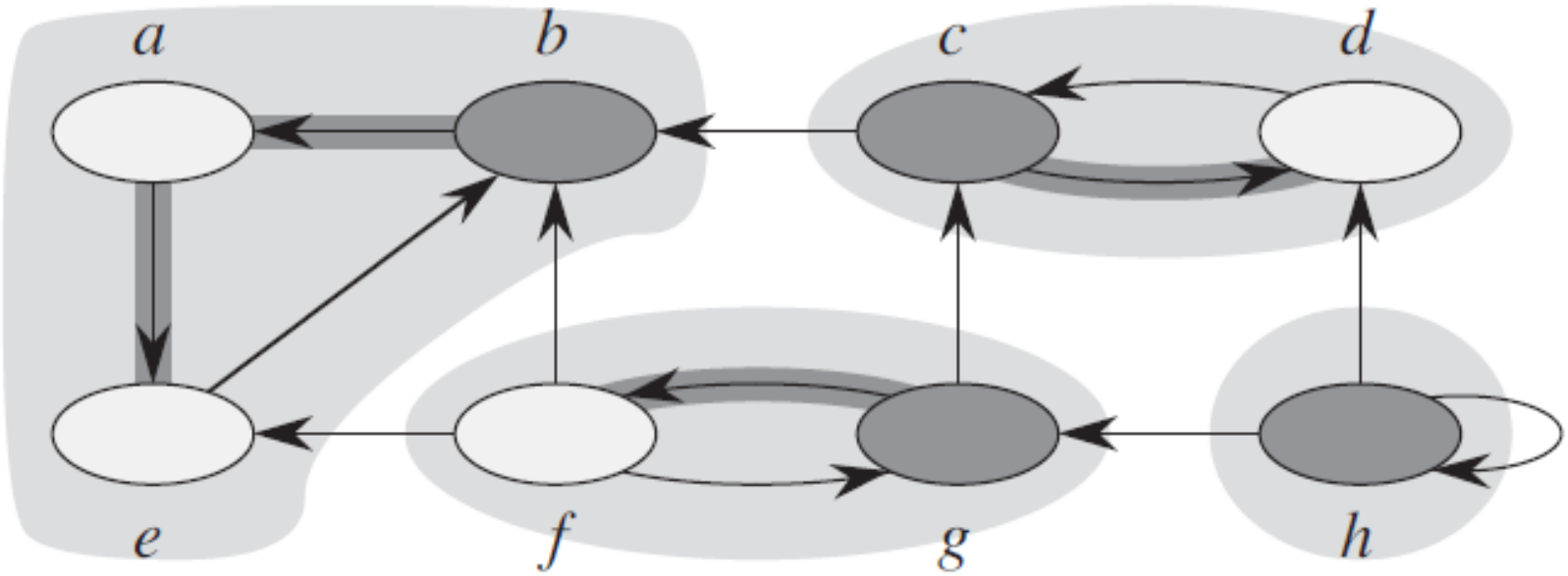



- $G^T$: grafo transposto



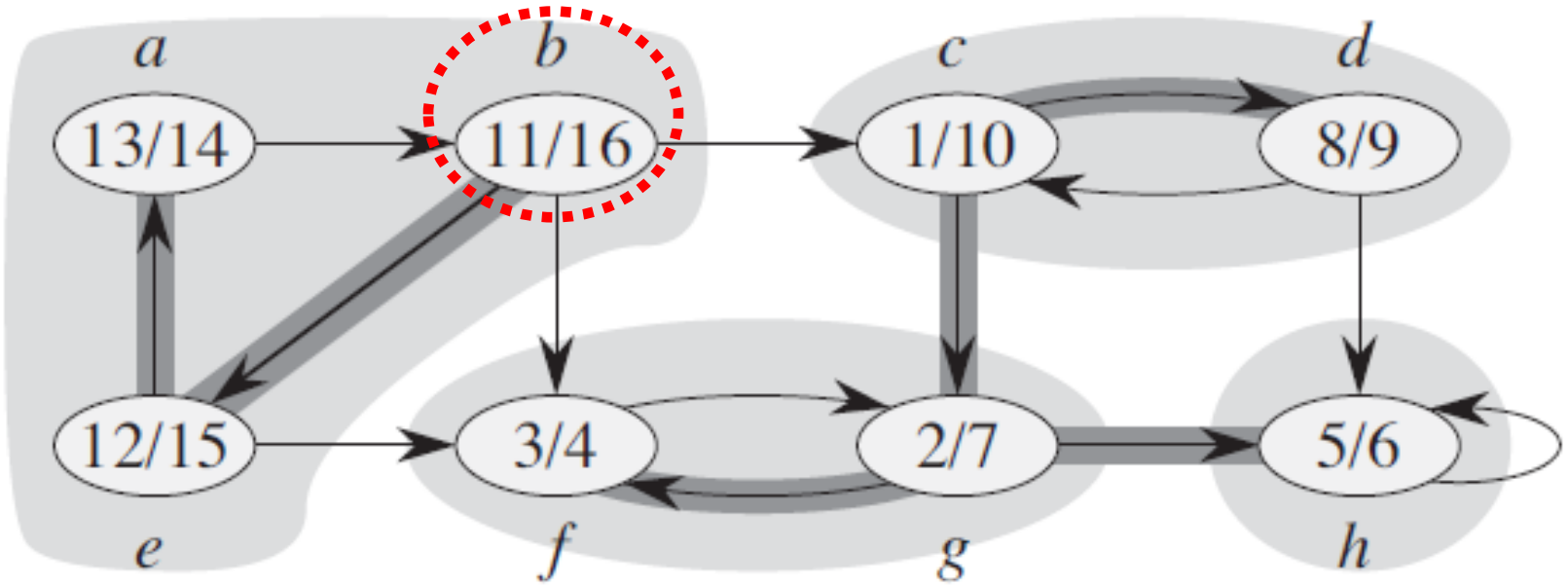
**Strongly-Connected-Components**(G)

1. executar **DFS**(**G**)
   para calcular instantes v.**f**
2. executar **DFS**($\mathbf{G^T}$), em ordem decrescente de v.**f** do passo 1



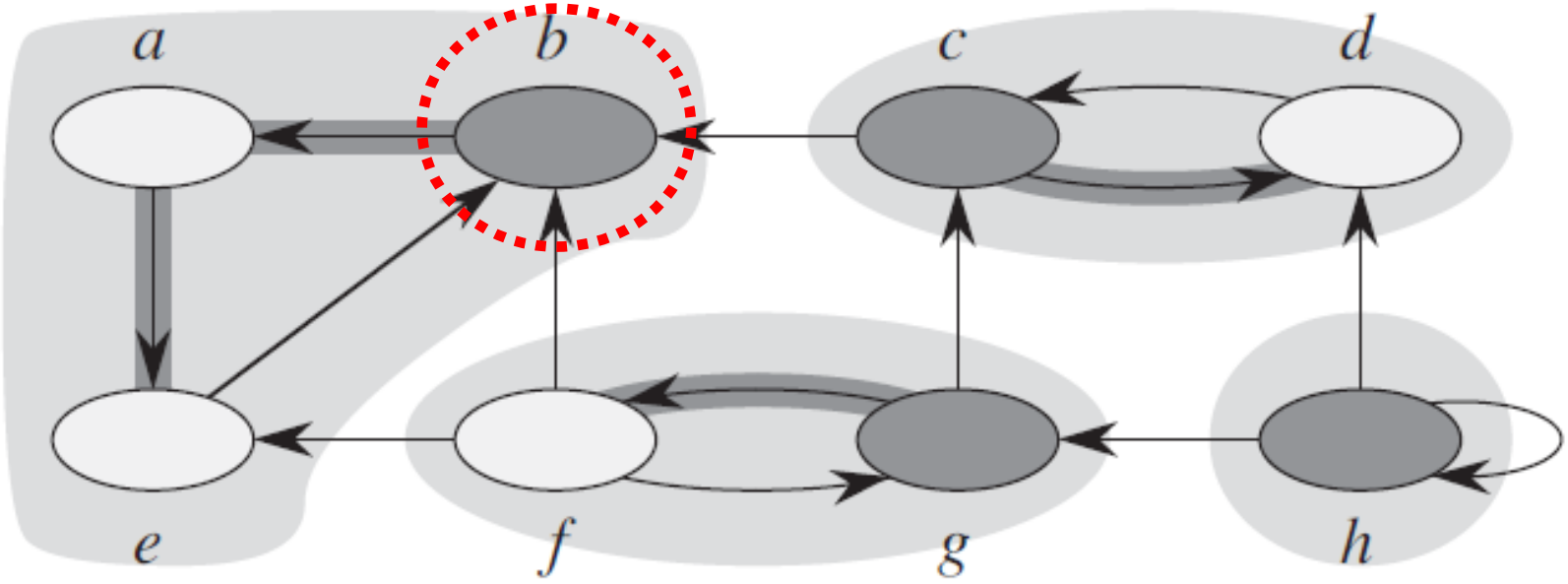
**Strongly-Connected-Components**(G)

1. executar **DFS**(**G**)
   para calcular instantes v.**f**
2. executar **DFS**(**G**$^{T}$), em ordem decrescente de v.**f** do **passo 1**



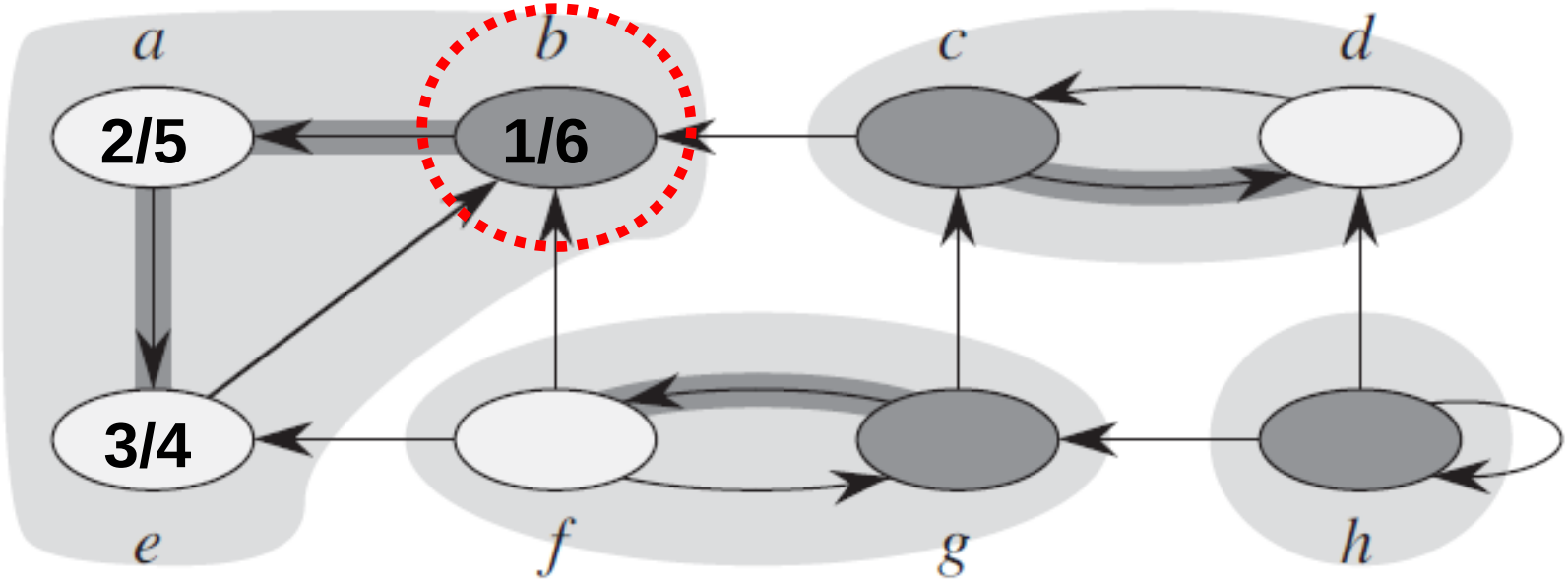
**Strongly-Connected-Components**(G)

1. executar **DFS**(**G**)
   para calcular instantes v.**f**
2. executar **DFS**(**G**$^{T}$), em ordem decrescente de v.**f** do passo 1



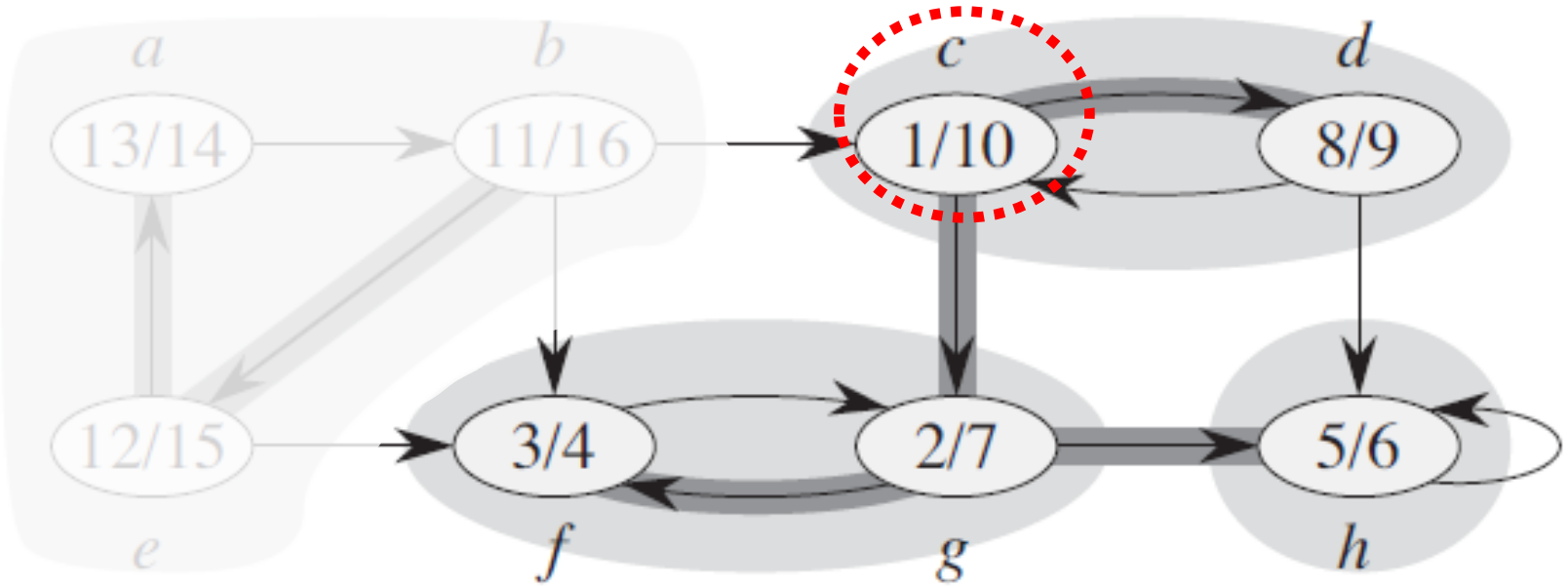
**Strongly-Connected-Components**(G)

1. executar **DFS**(**G**)
   para calcular instantes v.**f**
2. executar **DFS**(**G**$^{T}$), em ordem decrescente de v.**f** do passo 1



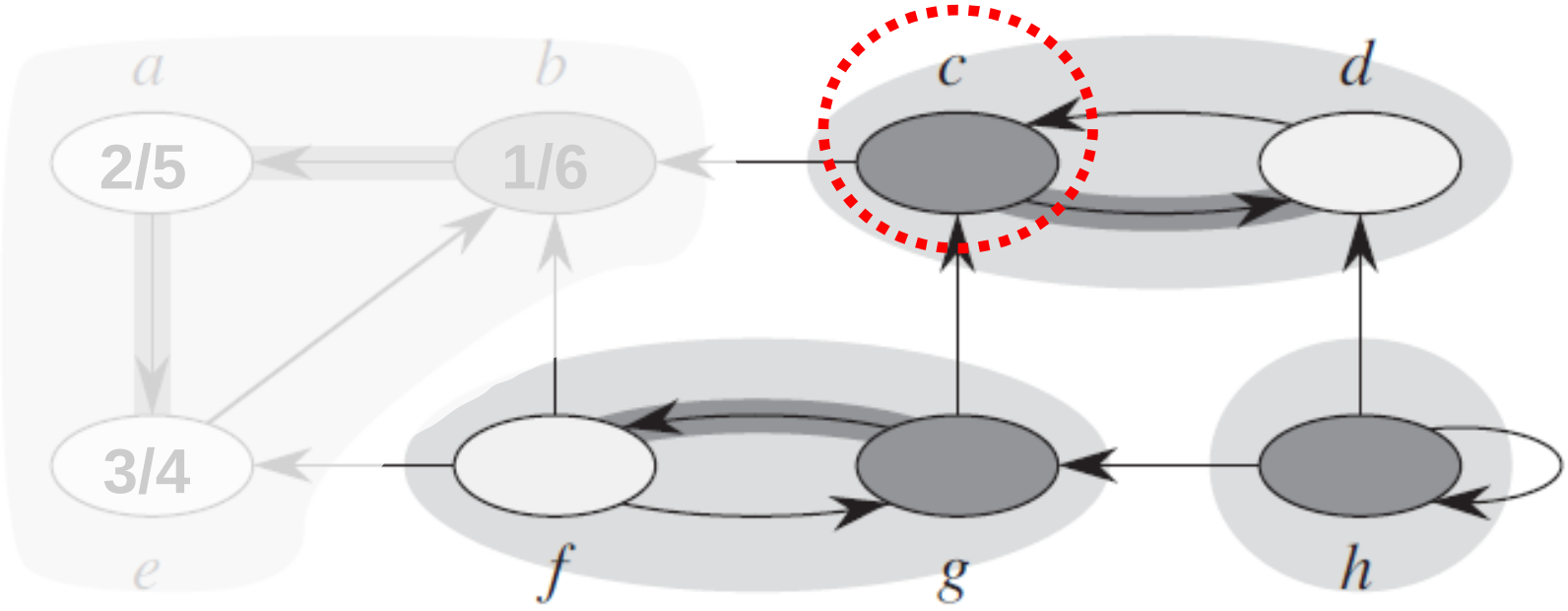
**Strongly-Connected-Components**(G)

1. executar **DFS**(**G**)
   para calcular instantes v.**f**
2. executar **DFS**(**G**$^{T}$), em ordem decrescente de v.**f** do **passo 1**



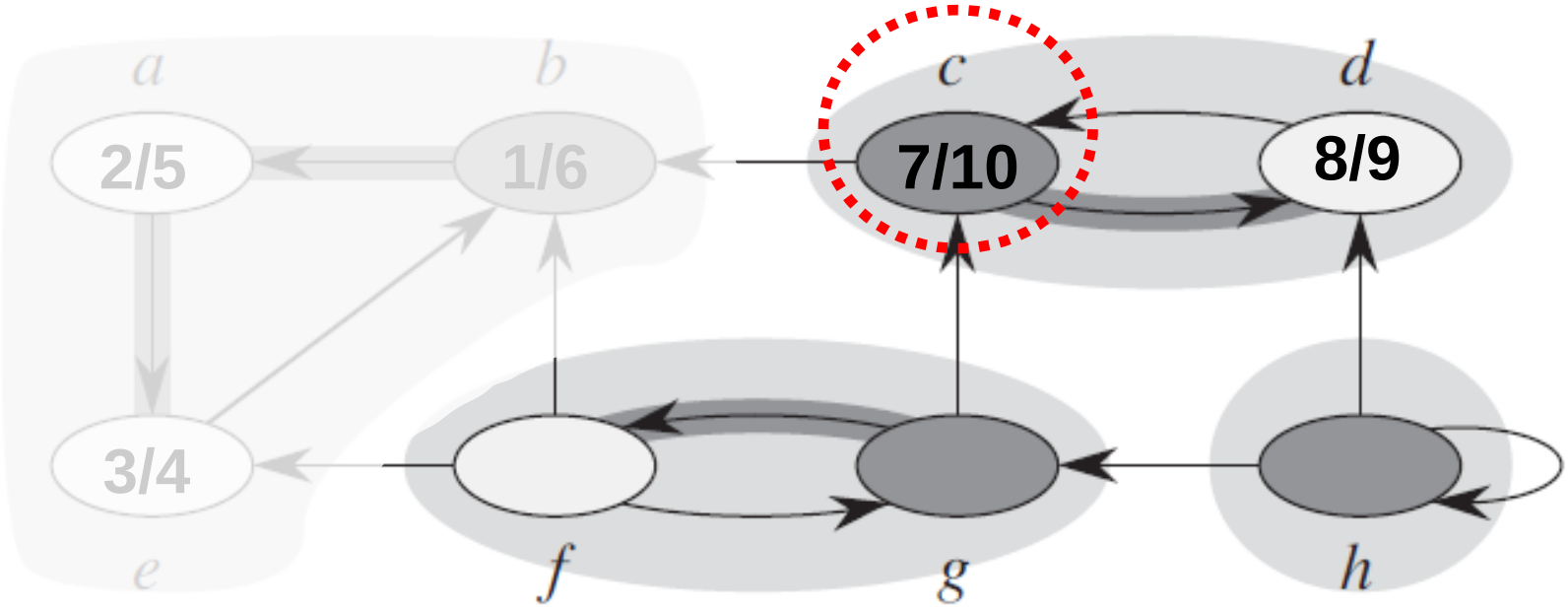
**Strongly-Connected-Components**(G)

1. executar **DFS**(**G**)
   para calcular instantes v.**f**
2. executar **DFS**(**G**$^{T}$), em ordem decrescente de v.**f** do passo 1



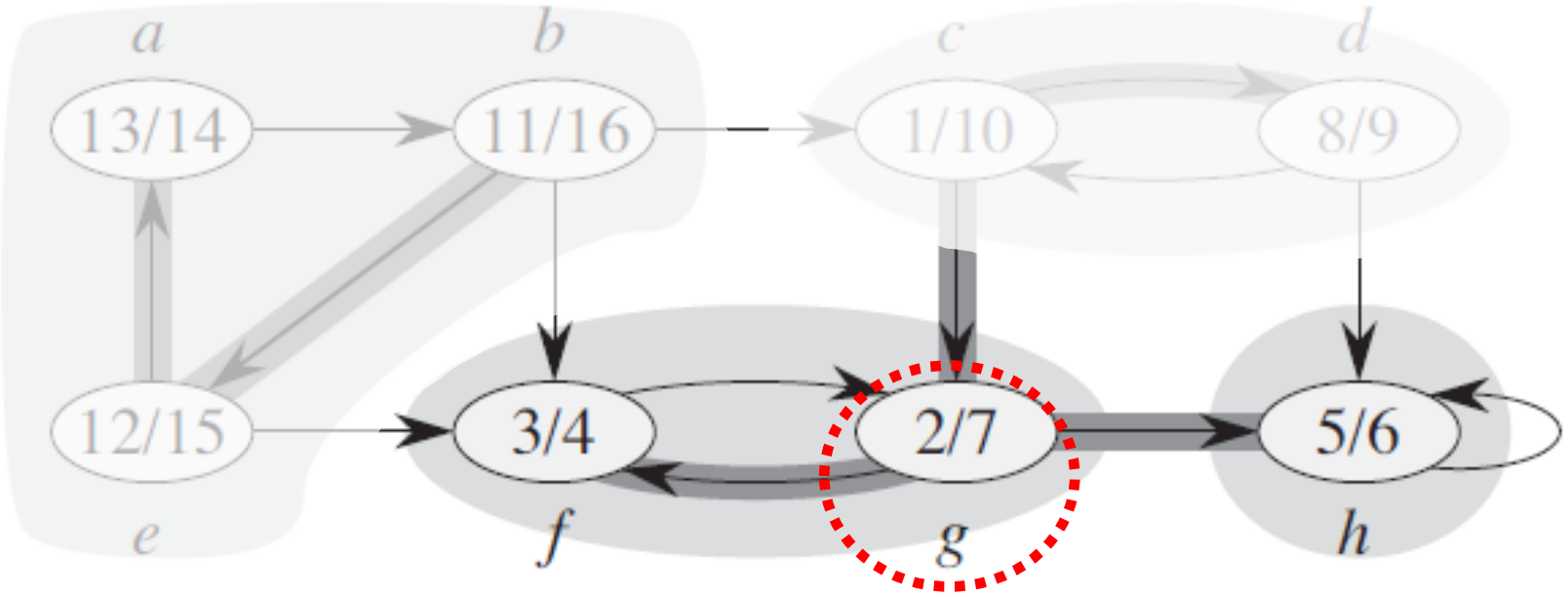
**Strongly-Connected-Components**(G)

1. executar **DFS**(**G**)
   para calcular instantes v.**f**
2. executar **DFS**(**G**$^{T}$), em ordem decrescente de v.**f** do passo 1



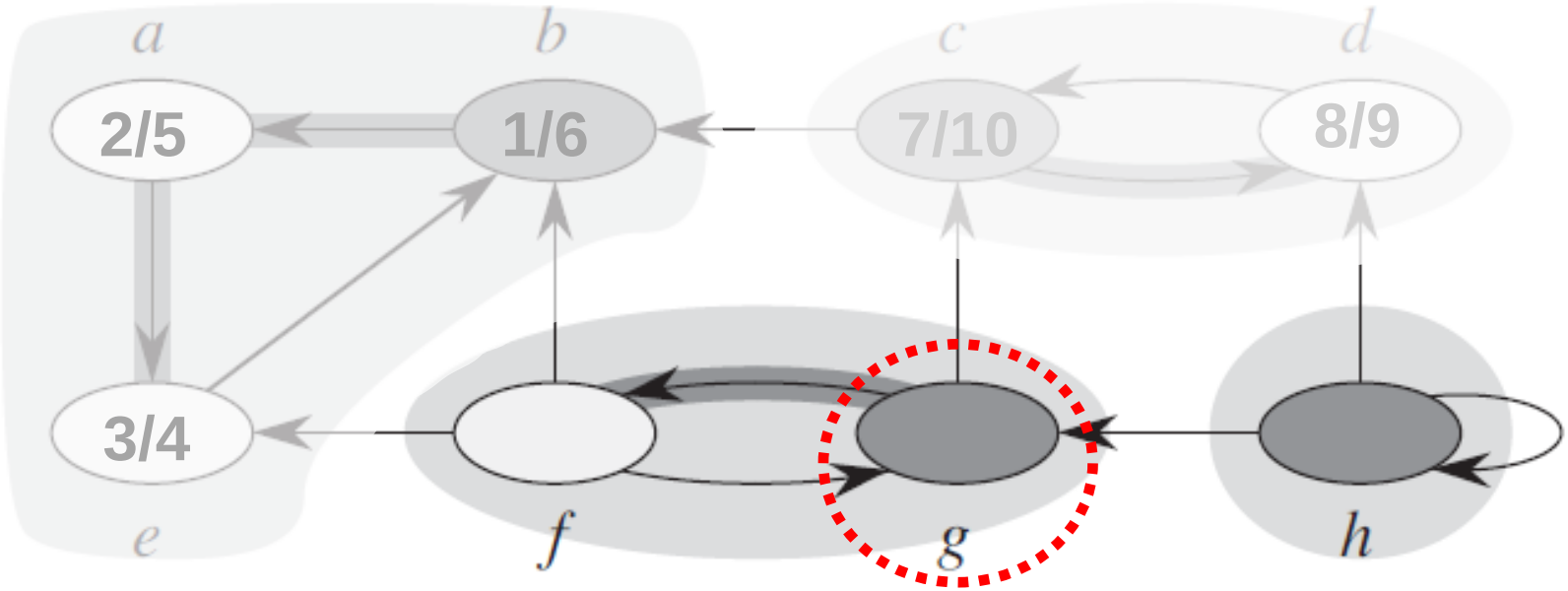
**Strongly-Connected-Components**(G)

1. executar **DFS**(**G**)
   para calcular instantes v.**f**
2. executar **DFS**(**G**$^{T}$), em ordem decrescente de v.**f** do **passo 1**



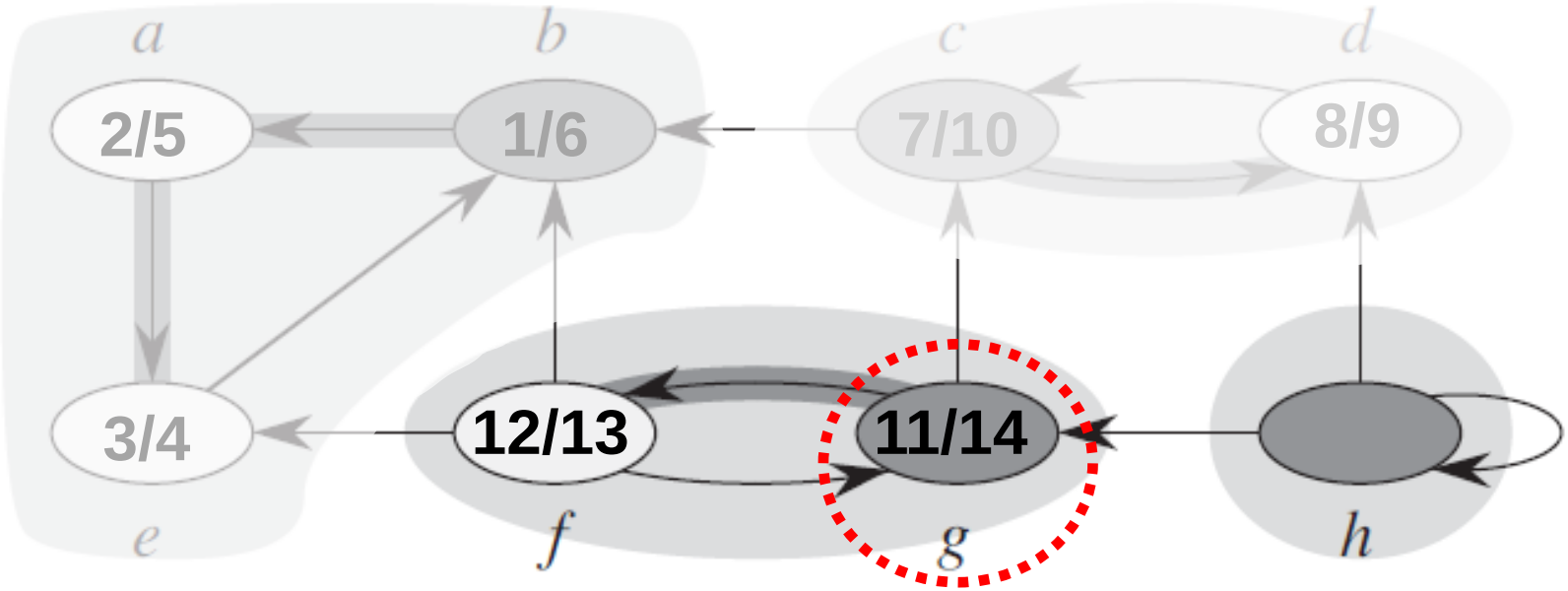
**Strongly-Connected-Components**(G)

1. executar **DFS**(**G**)
   para calcular instantes v.**f**
2. executar **DFS**(**G**$^{T}$), em ordem decrescente de v.**f** do passo 1



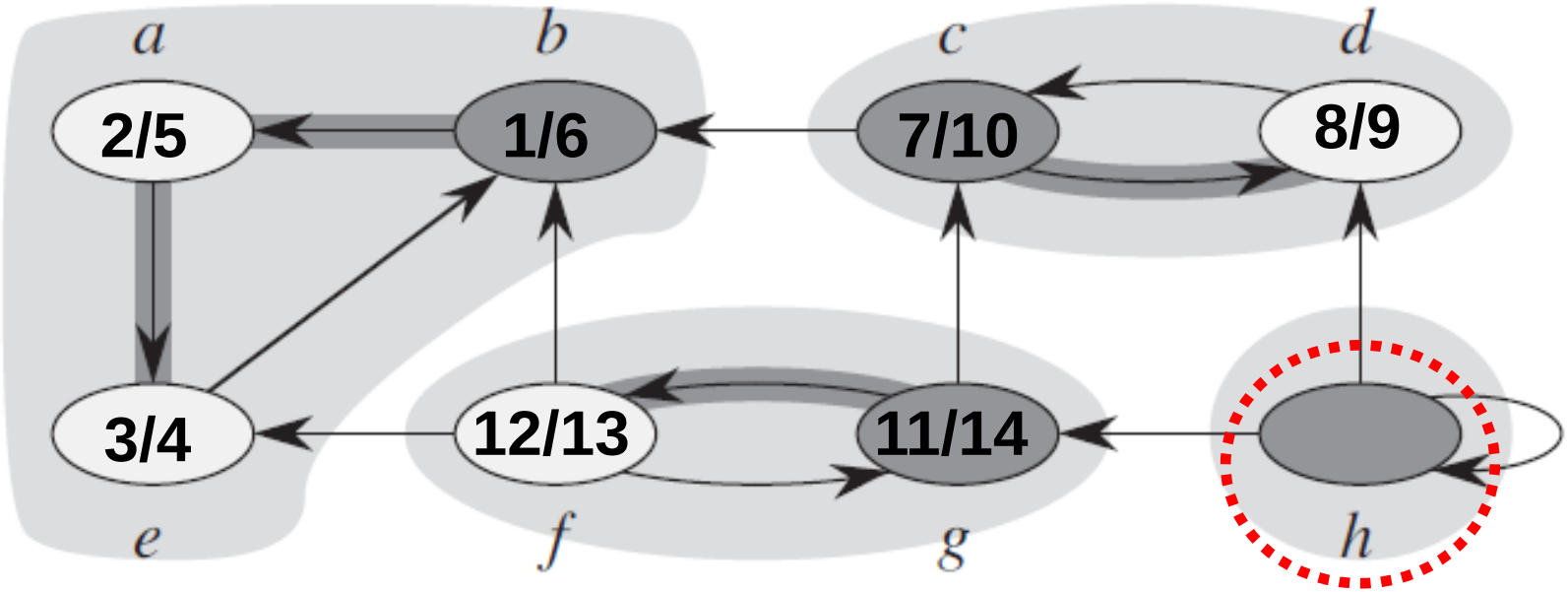
**Strongly-Connected-Components**(G)

1. executar **DFS**(**G**)
   para calcular instantes v.**f**
2. executar **DFS**(**G**$^{T}$), em ordem decrescente de v.**f** do passo 1



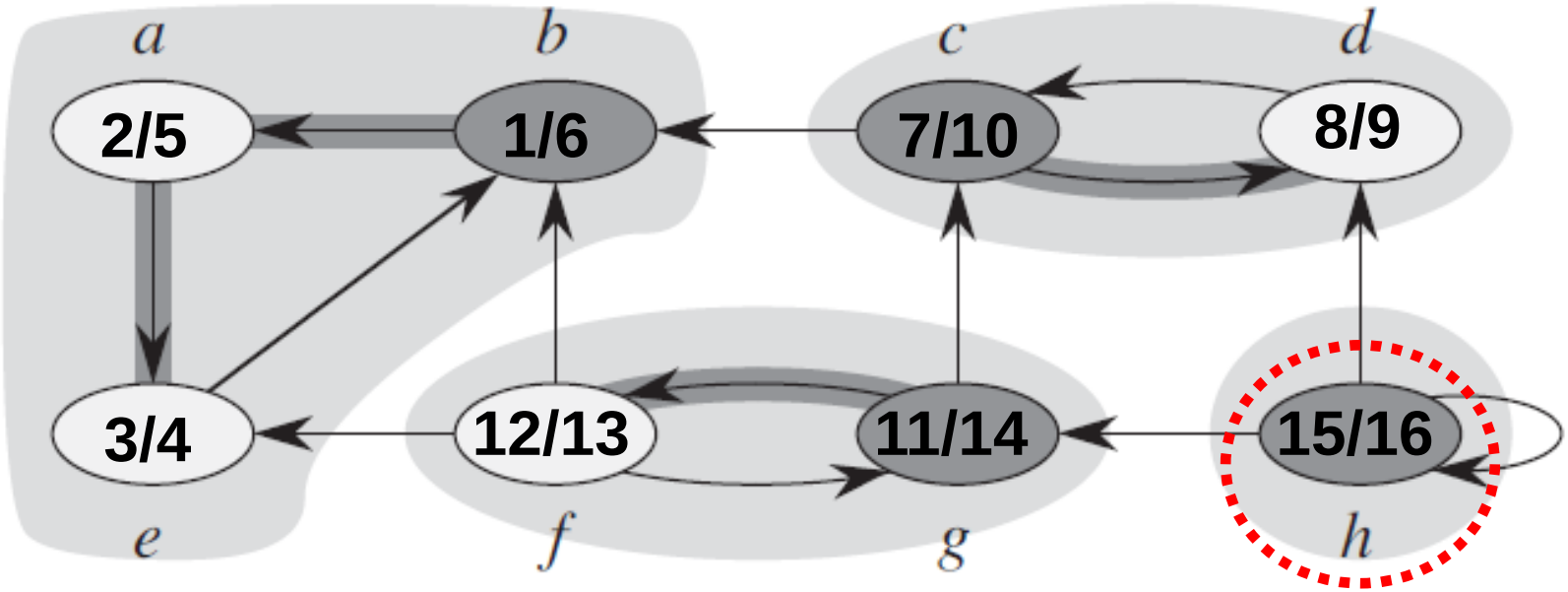
**Strongly-Connected-Components**(G)

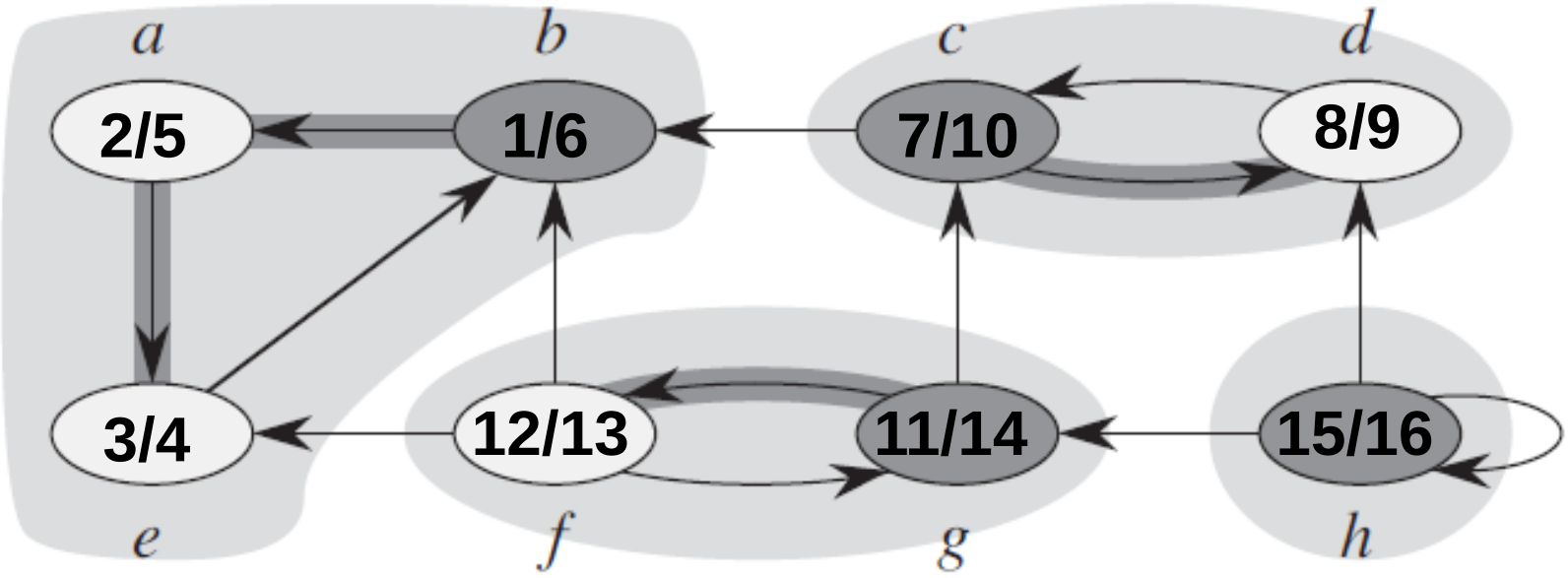
1. executar **DFS**(**G**)
   para calcular instantes v.**f**
2. executar **DFS**(**G**$^{T}$), em ordem decrescente de v.**f** do passo 1



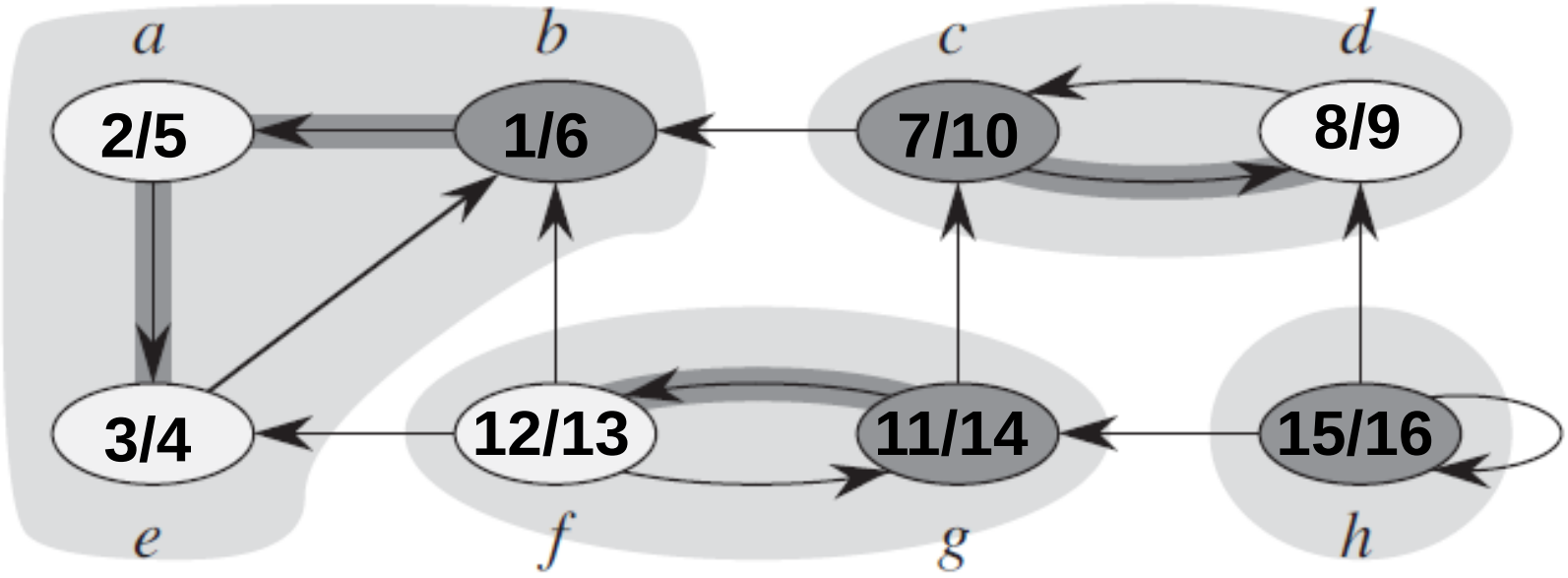
**Strongly-Connected-Components**(G)

1. executar **DFS**(**G**)
   para calcular instantes v.**f**
2. executar **DFS**(**G**$^{T}$), em ordem decrescente de v.**f** do passo 1



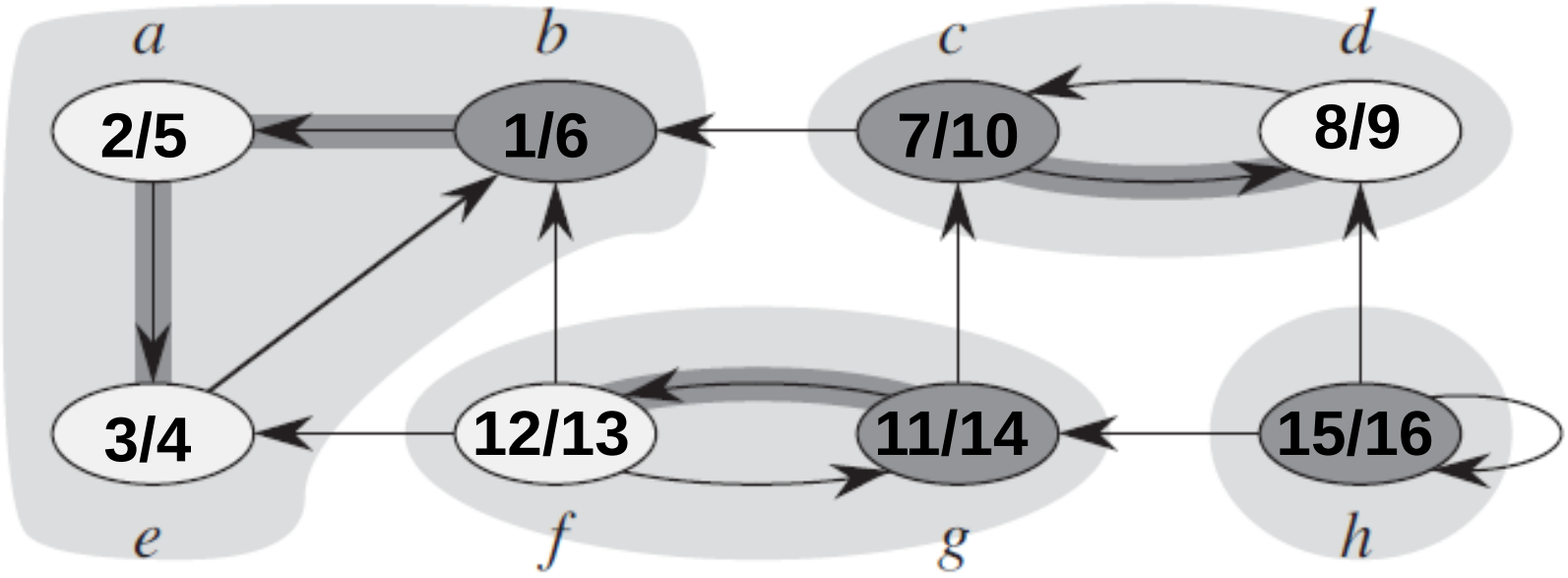



Mas e agora? Como obter os componentes fortes?




Expressão de parênteses?

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |



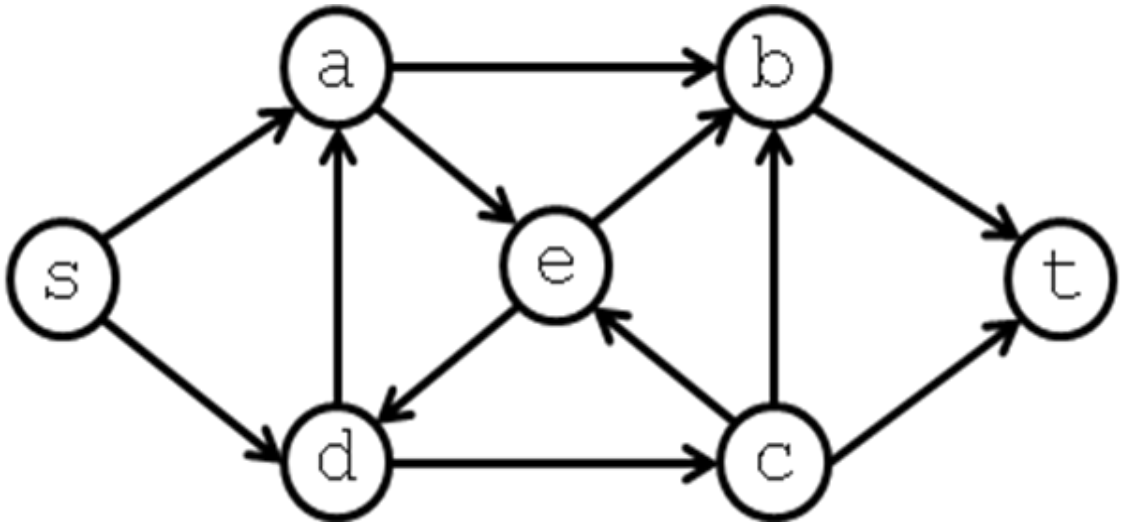

Expressão de parênteses

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| (b | (a | (e | e) | a) | b) | (c | (d | d) | c) | (g | (f | f) | g) | (h | h) |

# Exercícios

- Calcule os componentes "fortes":

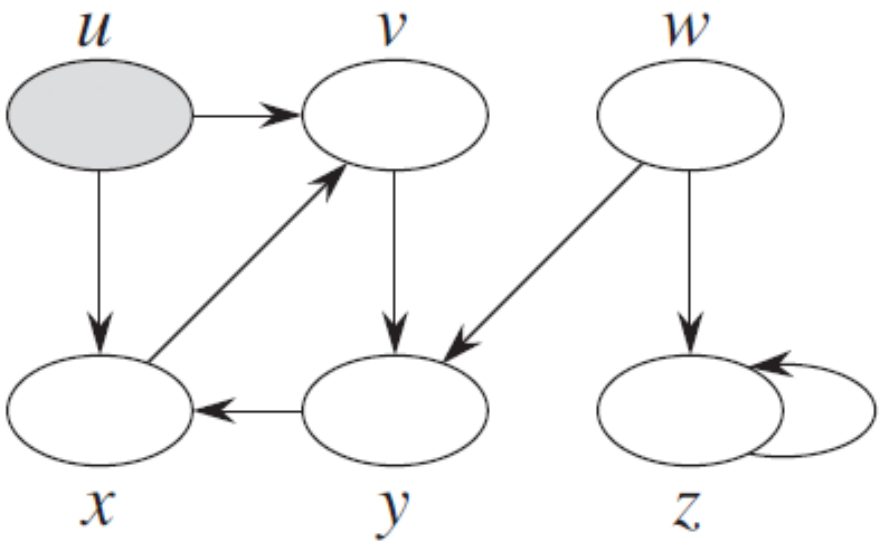
(a)



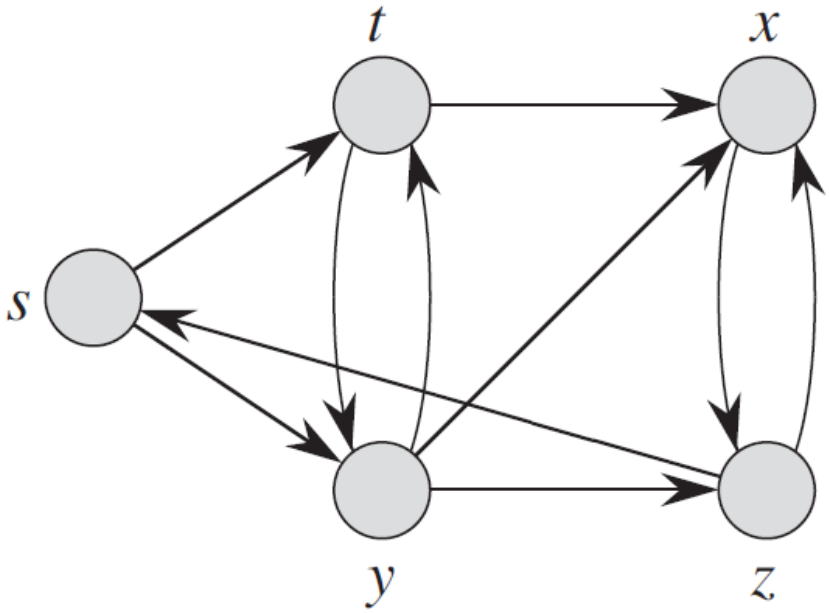
# Exercícios

- Calcule os componentes "fortes":

(b)



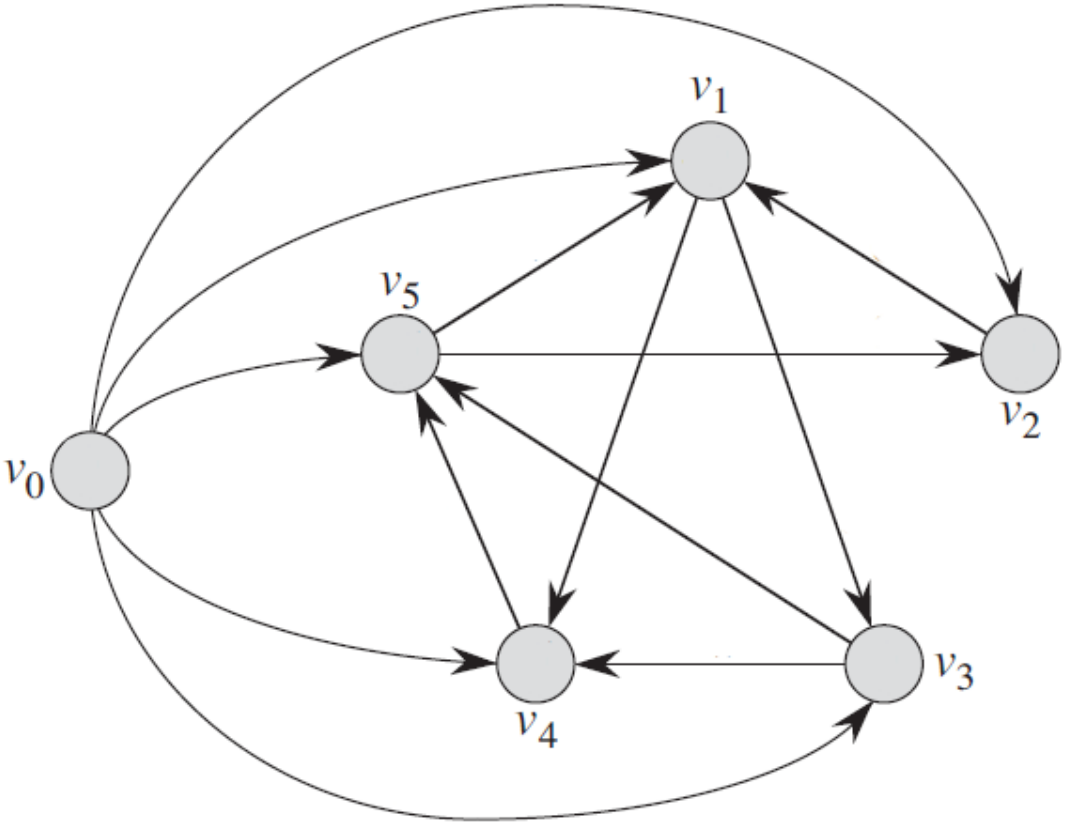

(c)



# Exercícios

- Calcule os componentes "fortes":

(d)



# Consumo de tempo?

**Consumo de tempo:**

**Strongly-Connected-Components**(G)

1. executar **DFS**(**G**) — **???**
   para calcular instantes v.**f**
2. executar **DFS**(**$G^T$**), em ordem decrescente de v.**f** do passo 1 — **???**

**Total**: **T(n,m) = ???**

# Consumo de tempo?

**Consumo de tempo:**

**Strongly-Connected-Components**(G)

1. executar **DFS**(**G**) para calcular instantes v.**f** — **O(n+m)**
2. executar **DFS**($\mathbf{G^T}$), em ordem decrescente de v.**f** do passo 1 — **O(n+m)**

**Total**: **T(n,m) = O(n+m)**

# Exercício Programa

- 07-dfs-compsFortes.py

# Exercícios

```
AlgoritmoOriginal(G)
 1 executar DFS(G) para calcular instantes v.f
 2 executar DFS(G^T), considerando ordem
decrescente de v.f do passo 1
```

- Será que o algoritmo do prof. Bacon está correto? Sim ou não? Justifique.

```
algoritmoBacon(G)
 1 executar DFS(G) para calcular instantes v.f
 2 executar DFS(G), considerando ordem crescente
de v.f do passo 1
```