



UNIVERSIDADE
FEDERAL DO CEARÁ

CAMPUS DE RUSSAS

Algoritmos em Grafos

Aula 10: Ordenação Topológica & Componentes Conexas

Professor Pablo Soares

2022.1

Sumário

1. Nomenclatura das arestas(**última aula**);
 - a. Árvore de Busca
2. Ordenação Topológica;
 - a. Precedência entre arestas/eventos.
3. Componentes Conexas.

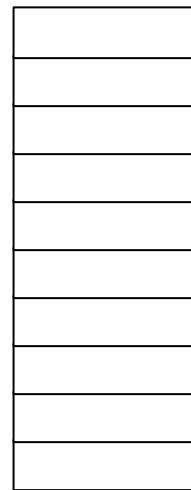
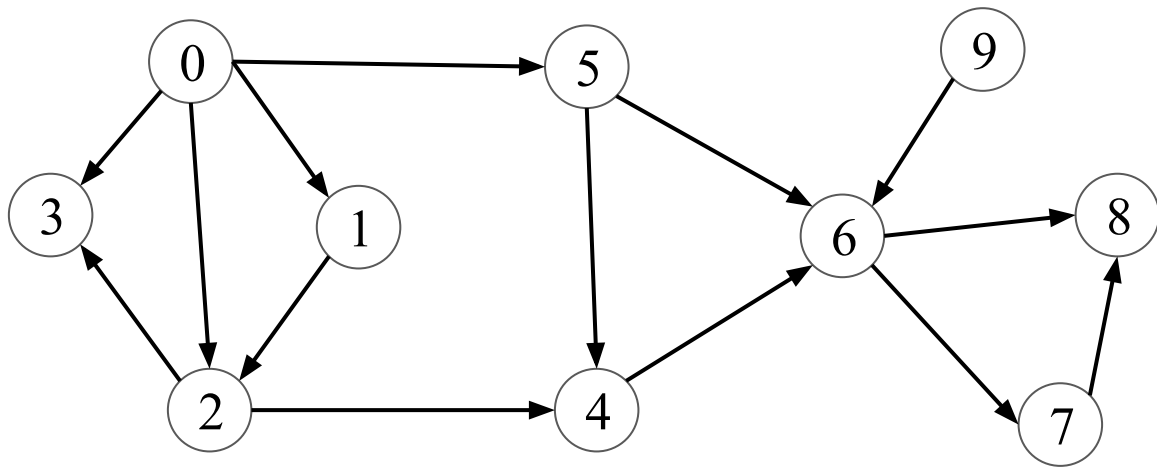
Ordenação Topológica

- *Aplicado em Grafos direcionados acíclicos;*
- *Ordenação linear de todos os vértices, tal que se G contém uma aresta (u, v) então “ u ” vem antes de “ v ”;*
- *Ordenação em linha horizontal;*
 - *Arestas direcionadas da esquerda para direita.*
- *são usados para indicar precedência entre eventos.*
 - *Hierarquia de herança entre classes em orientação a objetos;*
 - *Pré-requisitos entre disciplinas;*
 - *Restrições de cronograma entre tarefas de um projeto.*

Ordenação Topológica

➤ *Aplicar DFS + Pilha;*

- *Início no vértice com grau de entrada 0*

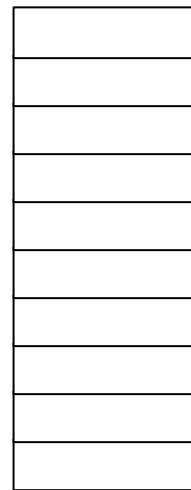
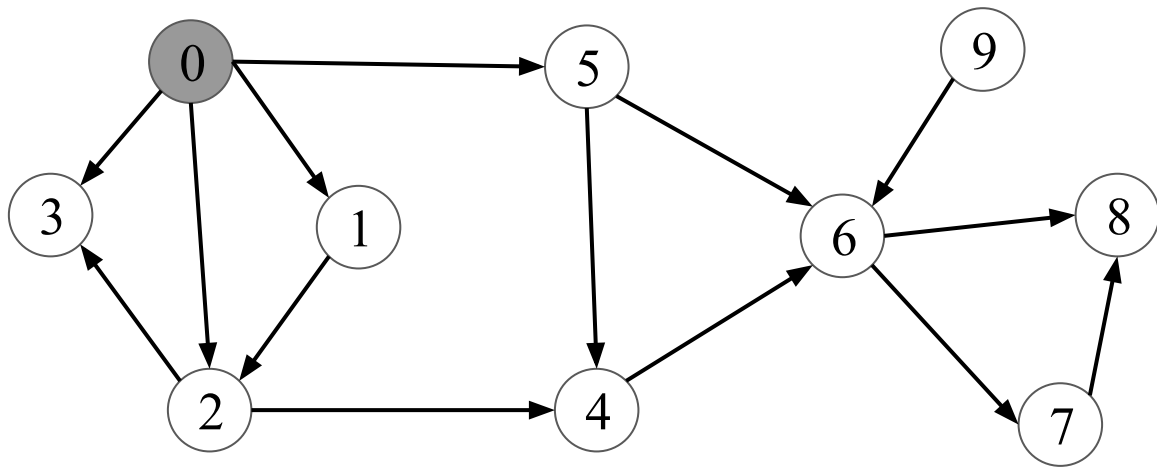


Pilha

Ordenação Topológica

➤ *Aplicar DFS + Pilha;*

- *Início no vértice com grau de entrada 0*

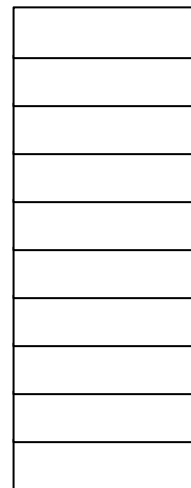
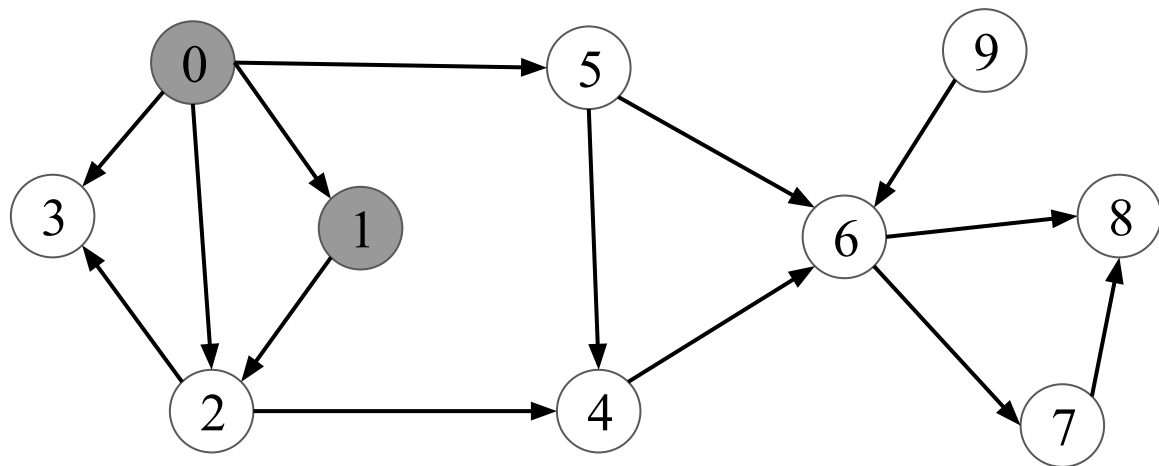


Pilha

Ordenação Topológica

➤ *Aplicar DFS + Pilha;*

- *Início no vértice com grau de entrada 0*

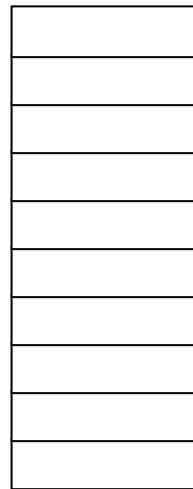
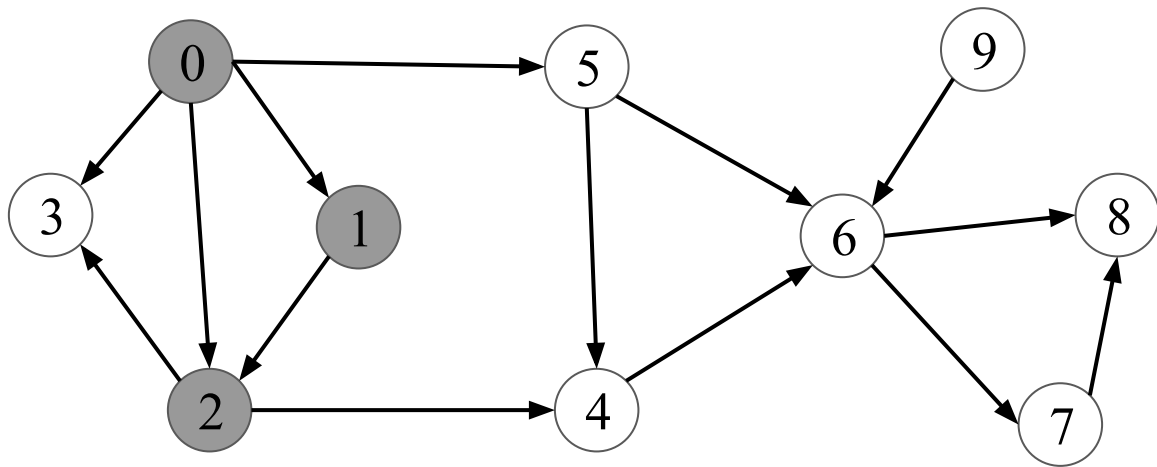


Pilha

Ordenação Topológica

➤ *Aplicar DFS + Pilha;*

- *Início no vértice com grau de entrada 0*

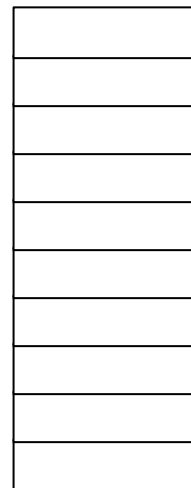
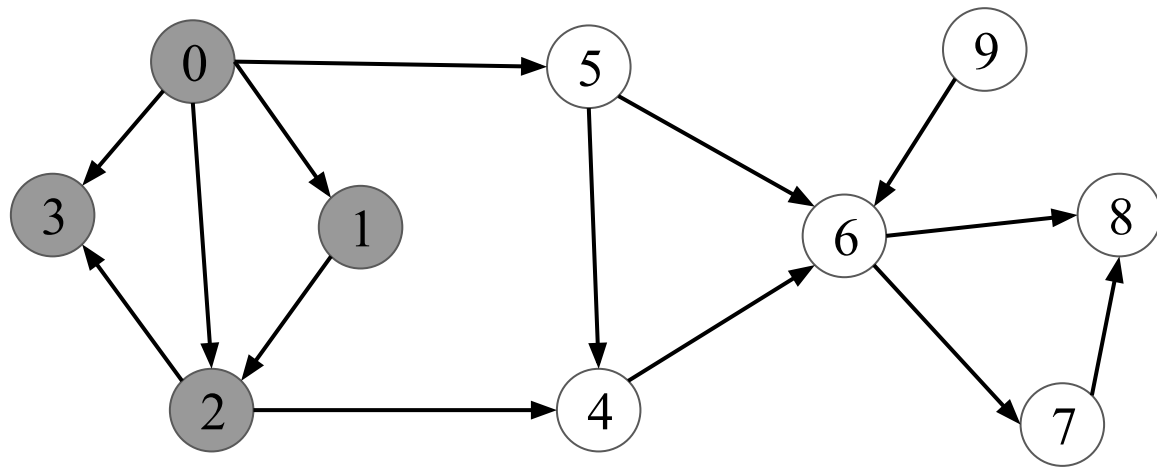


Pilha

Ordenação Topológica

➤ *Aplicar DFS + Pilha;*

- *Início no vértice com grau de entrada 0*

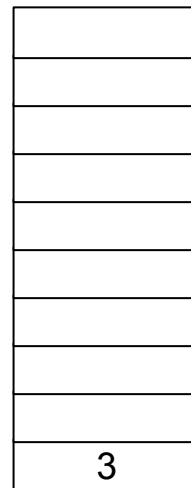
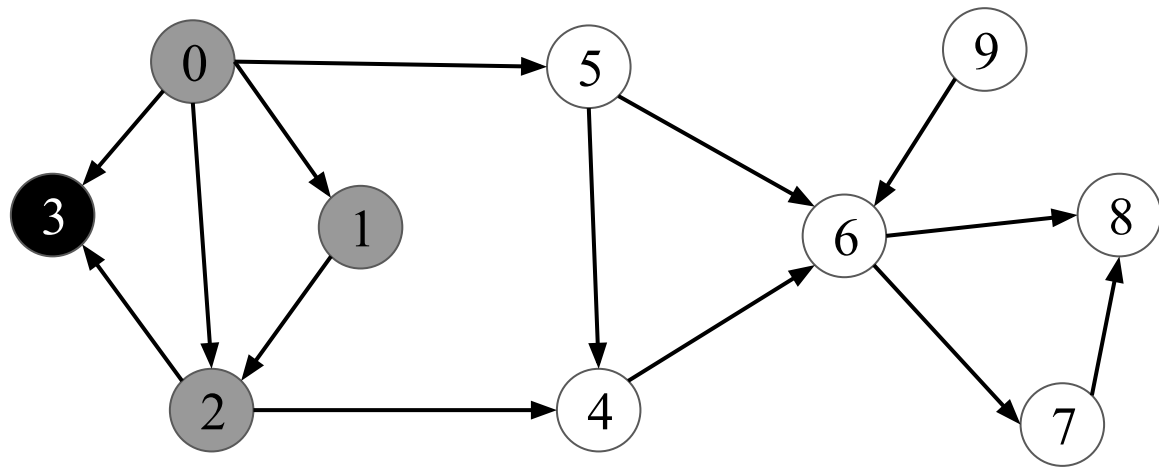


Pilha

Ordenação Topológica

➤ *Aplicar DFS + Pilha;*

- *Início no vértice com grau de entrada 0*

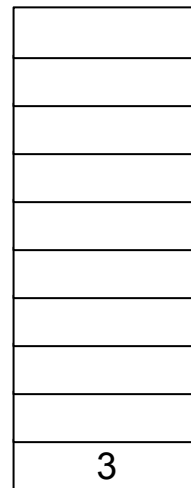
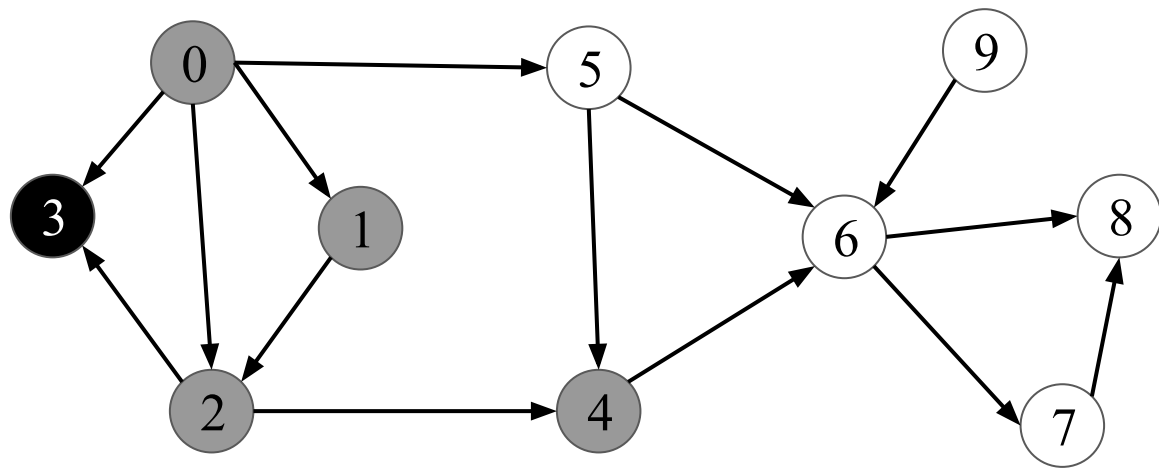


Pilha

Ordenação Topológica

➤ *Aplicar DFS + Pilha;*

- *Início no vértice com grau de entrada 0*

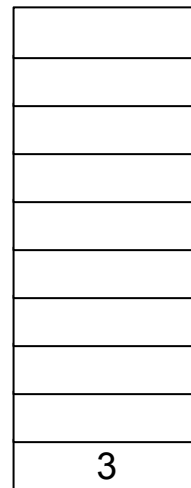
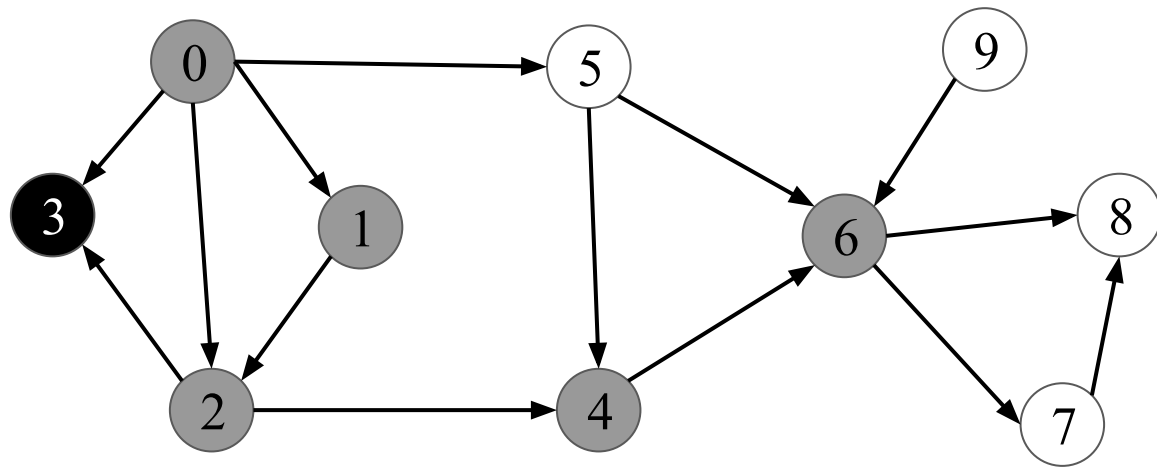


Pilha

Ordenação Topológica

➤ *Aplicar DFS + Pilha;*

- *Início no vértice com grau de entrada 0*

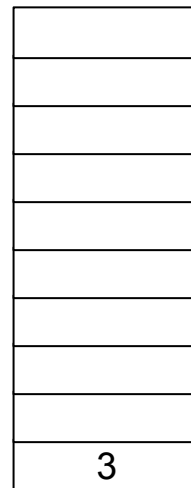
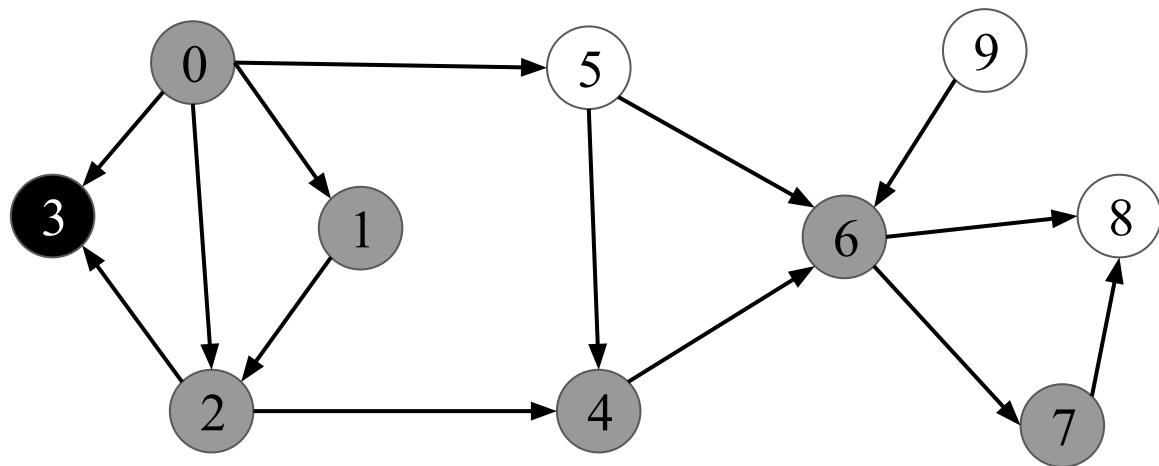


Pilha

Ordenação Topológica

➤ *Aplicar DFS + Pilha;*

- *Início no vértice com grau de entrada 0*

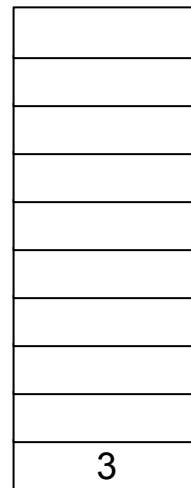
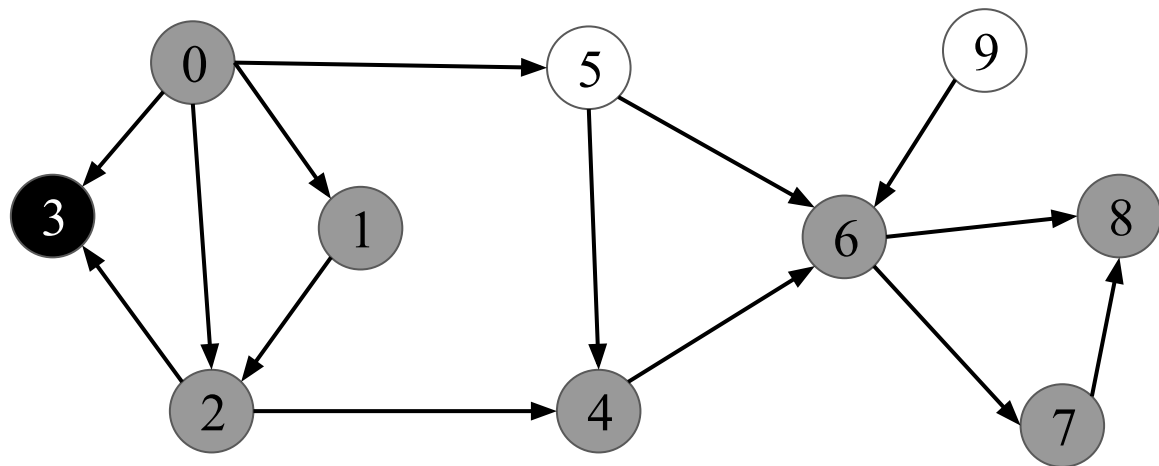


Pilha

Ordenação Topológica

➤ *Aplicar DFS + Pilha;*

- *Início no vértice com grau de entrada 0*

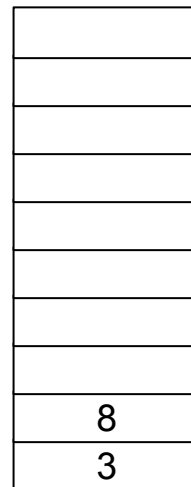
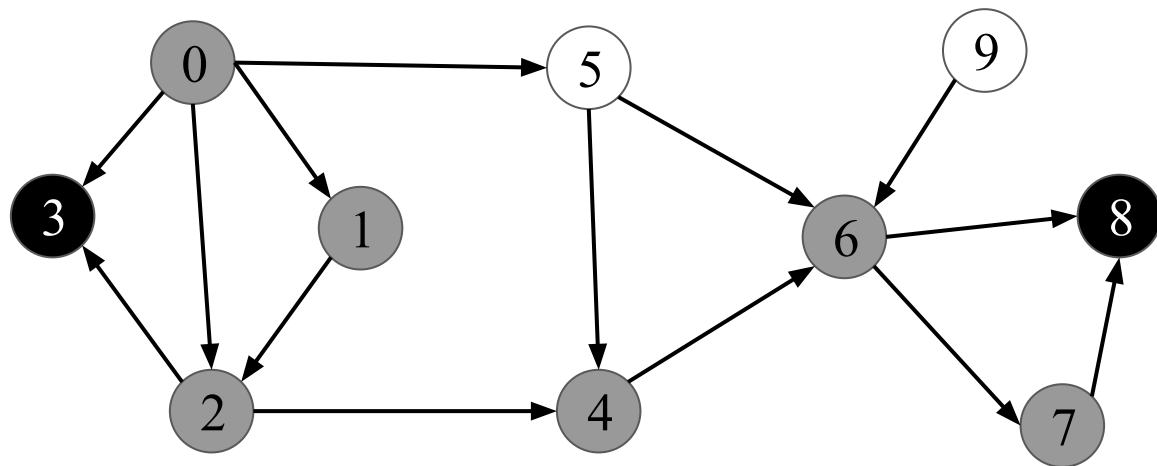


Pilha

Ordenação Topológica

➤ *Aplicar DFS + Pilha;*

- *Início no vértice com grau de entrada 0*

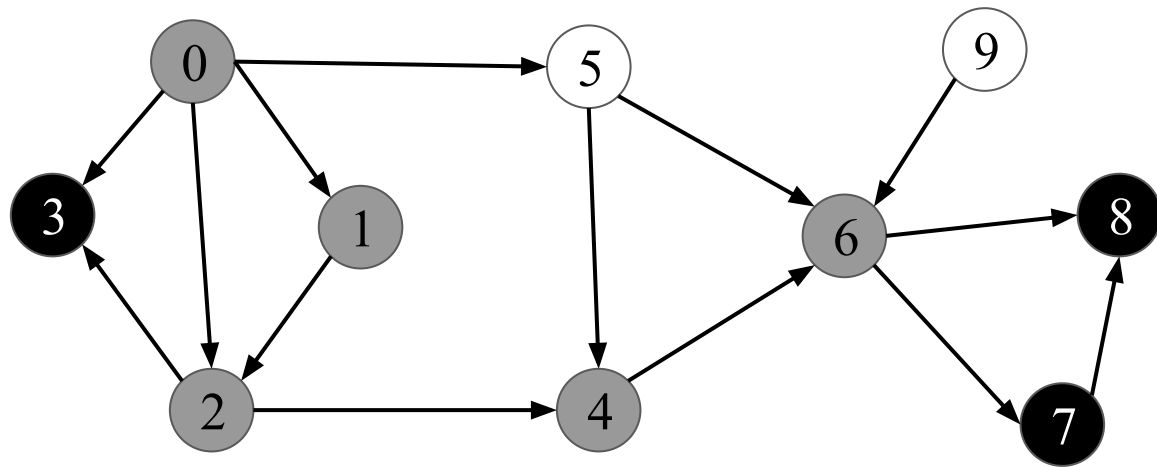


Pilha

Ordenação Topológica

➤ *Aplicar DFS + Pilha;*

- *Início no vértice com grau de entrada 0*



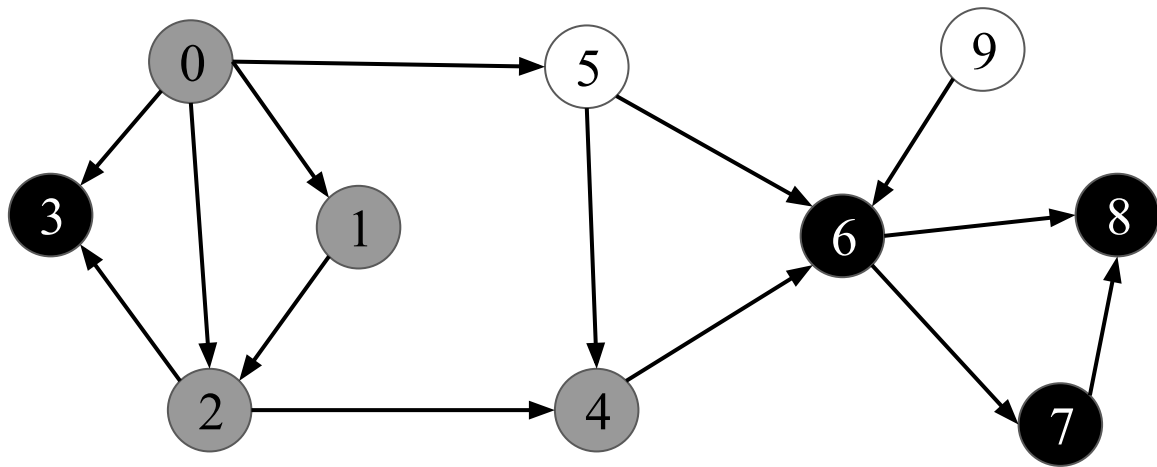
7
8
3

Pilha

Ordenação Topológica

➤ *Aplicar DFS + Pilha;*

- *Início no vértice com grau de entrada 0*



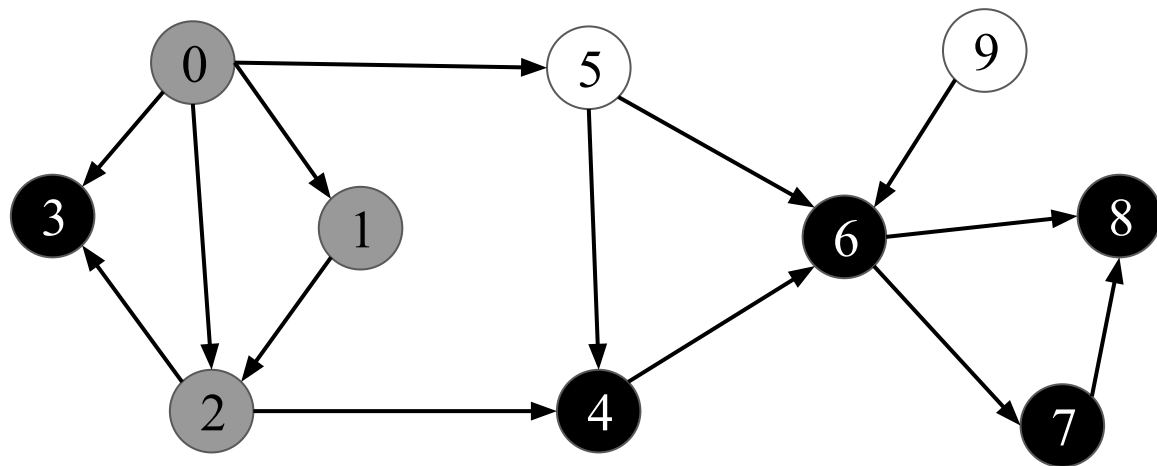
6
7
8
3

Pilha

Ordenação Topológica

➤ *Aplicar DFS + Pilha;*

- *Início no vértice com grau de entrada 0*



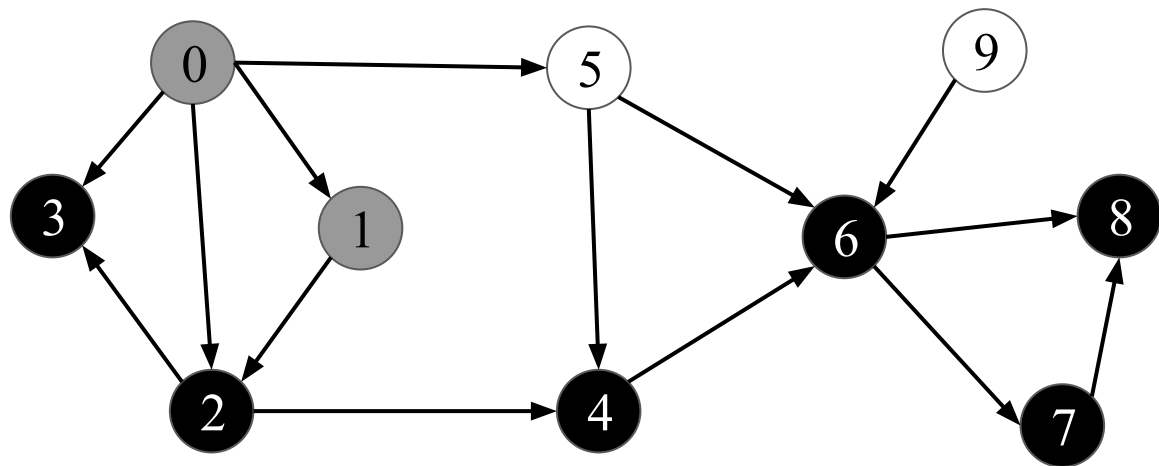
4
6
7
8
3

Pilha

Ordenação Topológica

➤ *Aplicar DFS + Pilha;*

- *Início no vértice com grau de entrada 0*



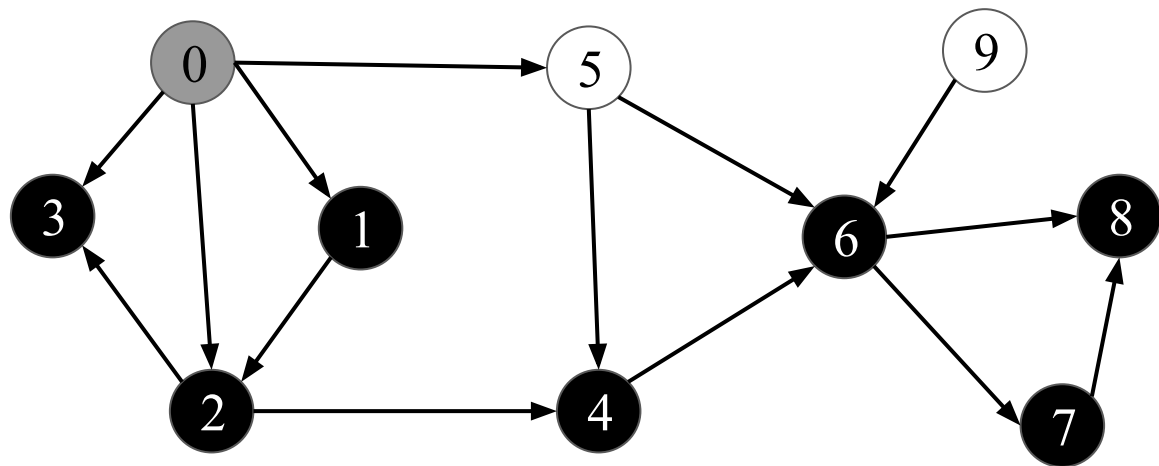
2
4
6
7
8
3

Pilha

Ordenação Topológica

➤ *Aplicar DFS + Pilha;*

- *Início no vértice com grau de entrada 0*



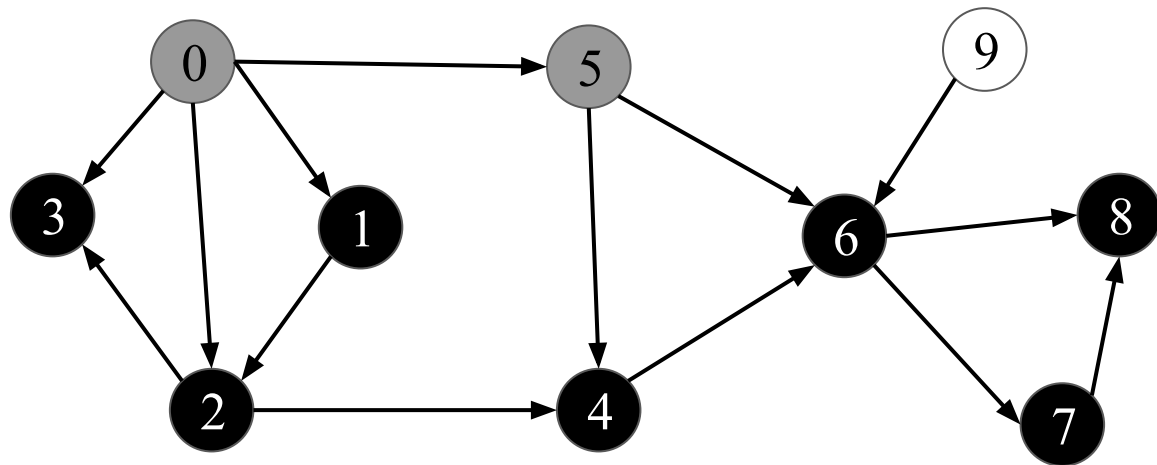
1
2
4
6
7
8
3

Pilha

Ordenação Topológica

➤ *Aplicar DFS + Pilha;*

- *Início no vértice com grau de entrada 0*



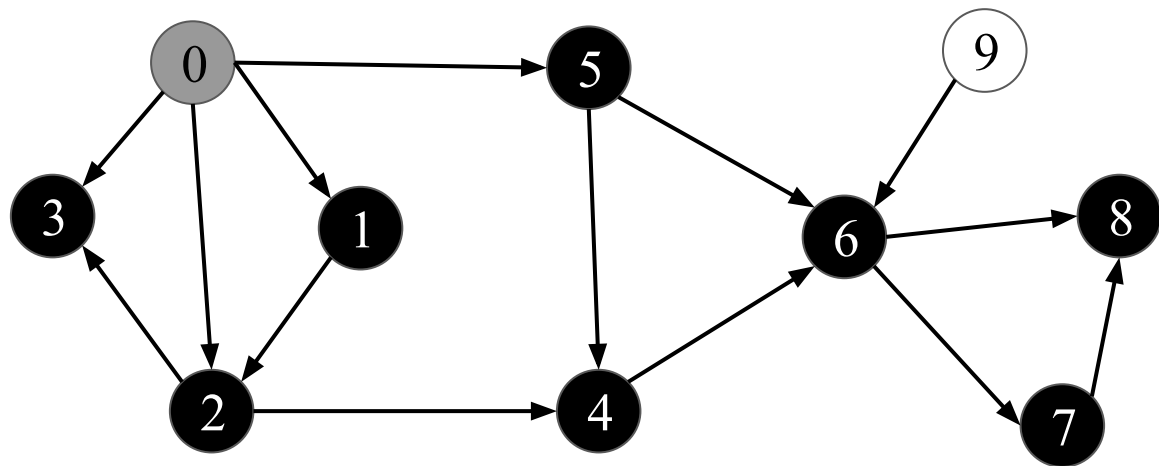
1
2
4
6
7
8
3

Pilha

Ordenação Topológica

➤ *Aplicar DFS + Pilha;*

- *Início no vértice com grau de entrada 0*



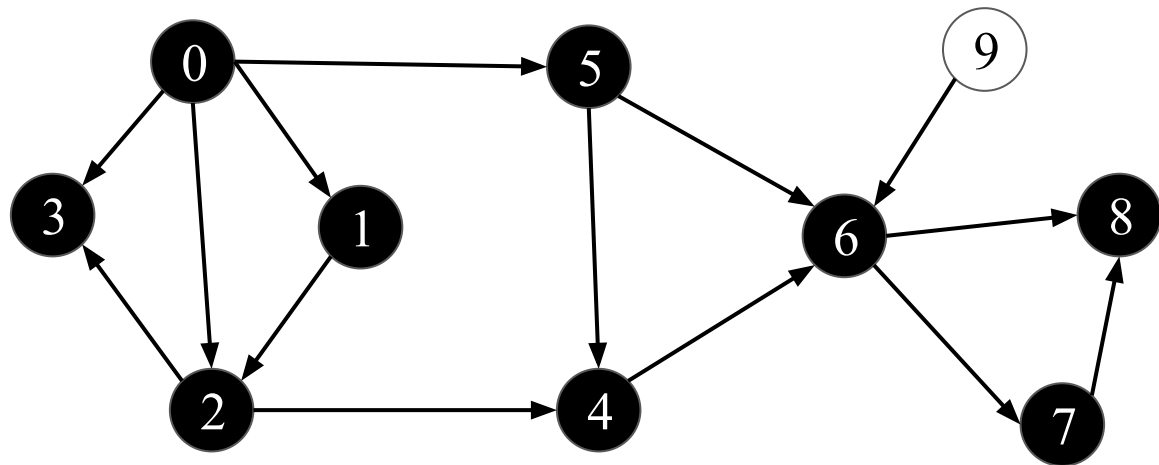
5
1
2
4
6
7
8
3

Pilha

Ordenação Topológica

➤ *Aplicar DFS + Pilha;*

- *Início no vértice com grau de entrada 0*



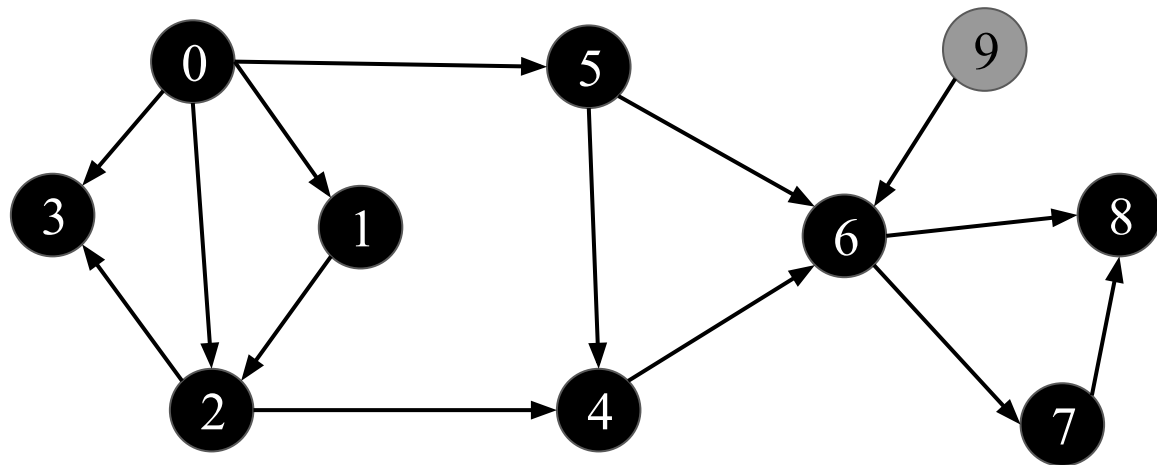
0
5
1
2
4
6
7
8
3

Pilha

Ordenação Topológica

➤ *Aplicar DFS + Pilha;*

- *Início no vértice com grau de entrada 0*



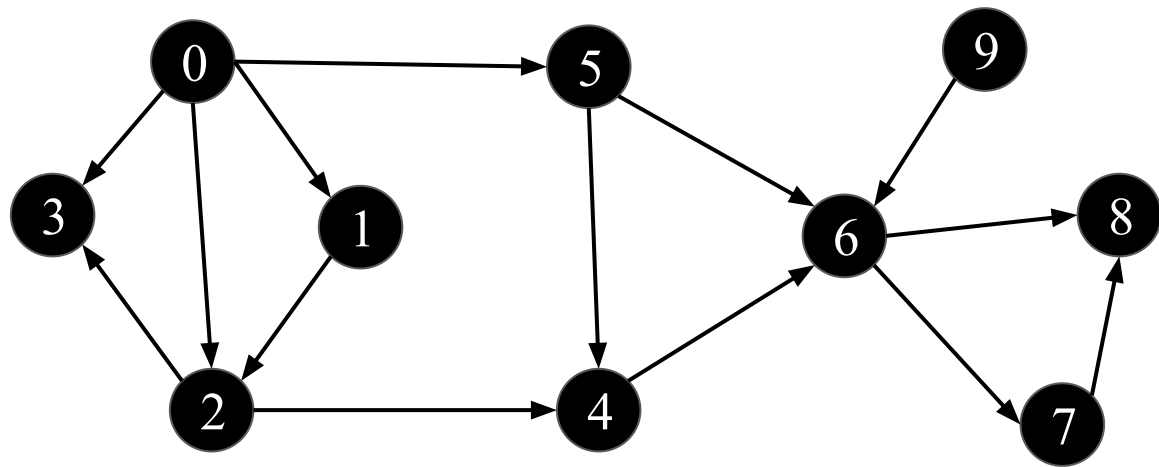
0
5
1
2
4
6
7
8
3

Pilha

Ordenação Topológica

➤ *Aplicar DFS + Pilha;*

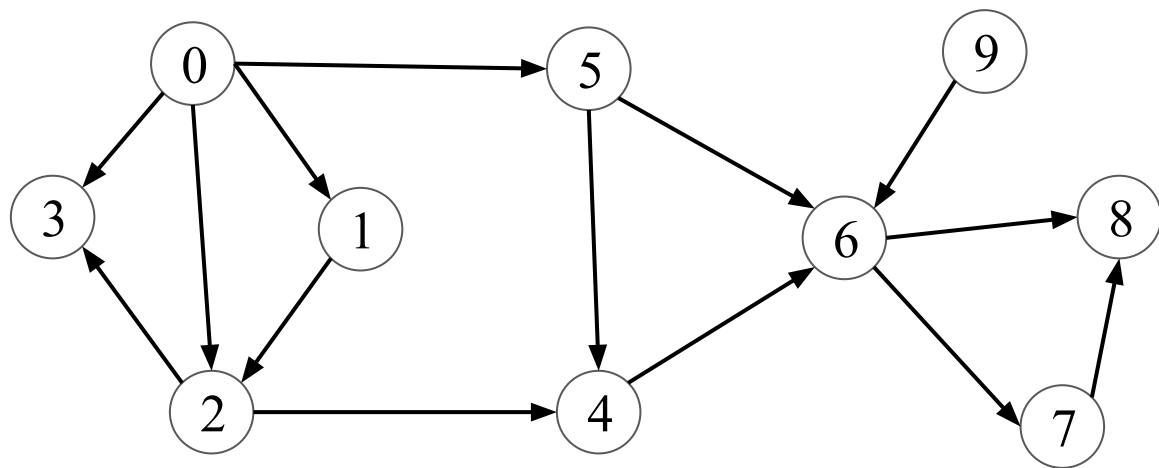
- *Início no vértice com grau de entrada 0*



9
0
5
1
2
4
6
7
8
3

Pilha

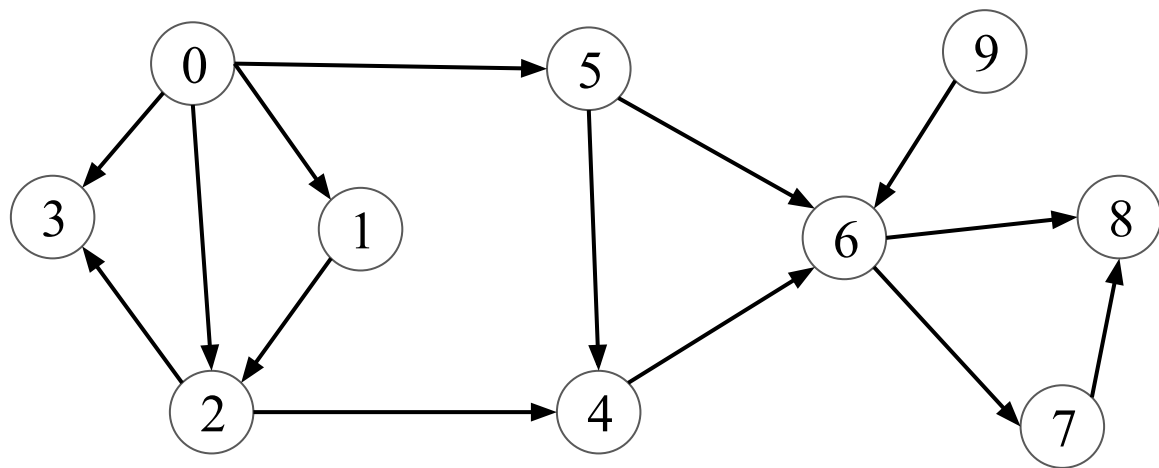
Ordenação Topológica



9
0
5
1
2
4
6
7
8
3

Pilha

Ordenação Topológica

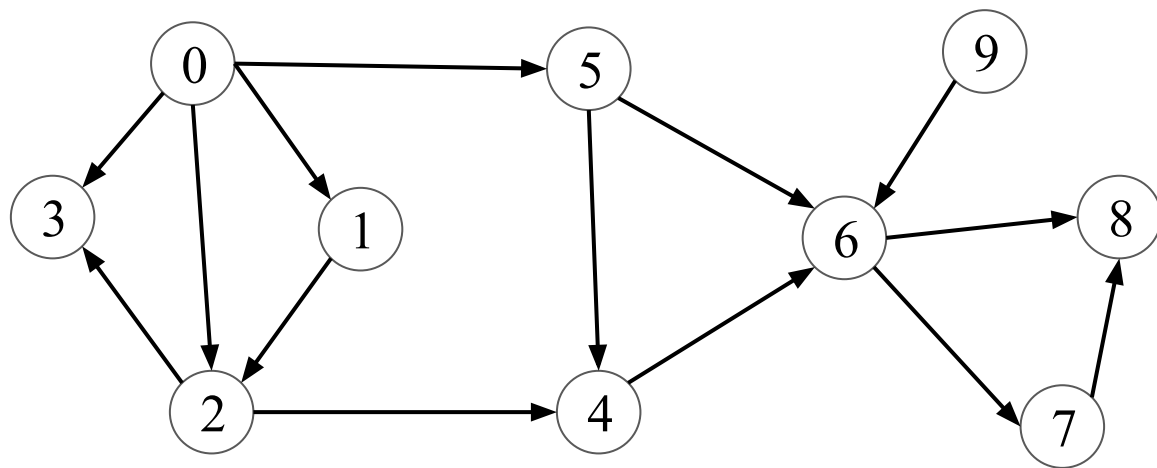


0
5
1
2
4
6
7
8
3

Pilha

9

Ordenação Topológica



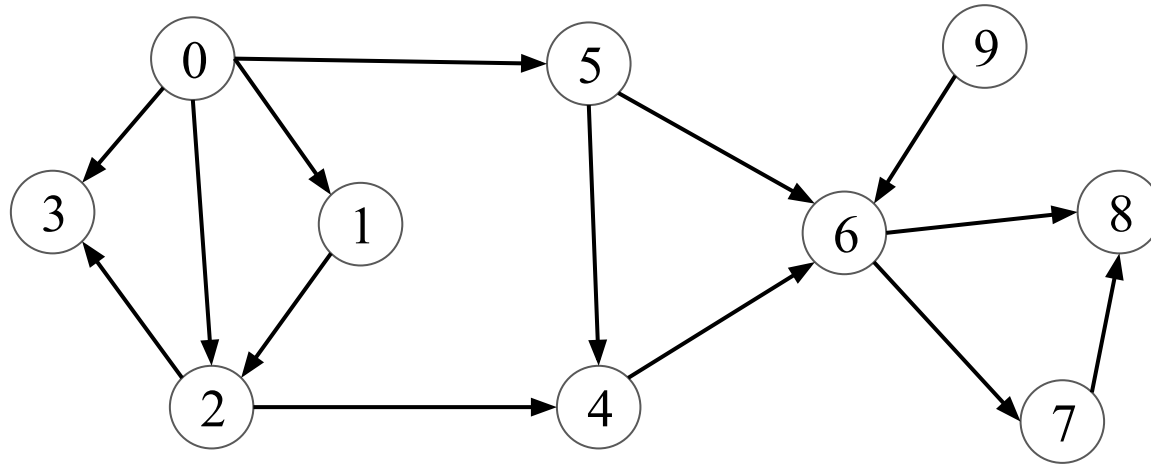
5
1
2
4
6
7
8
3

Pilha

9

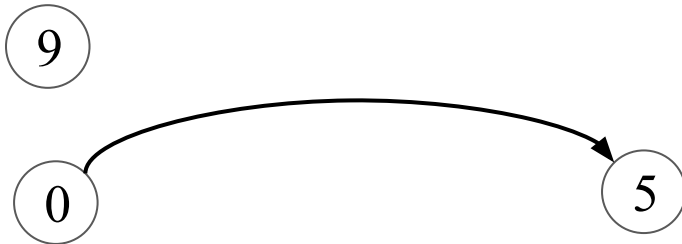
0

Ordenação Topológica

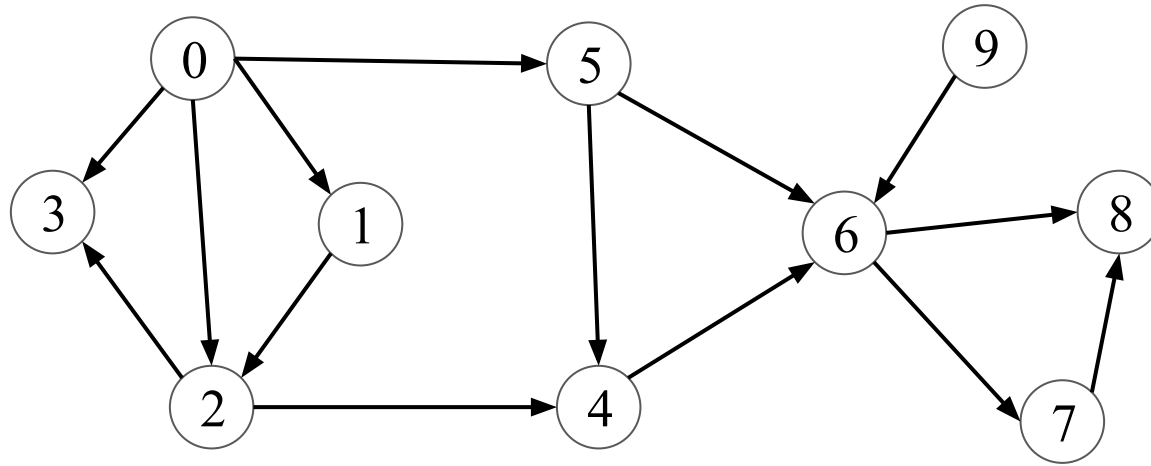


1
2
4
6
7
8
3

Pilha

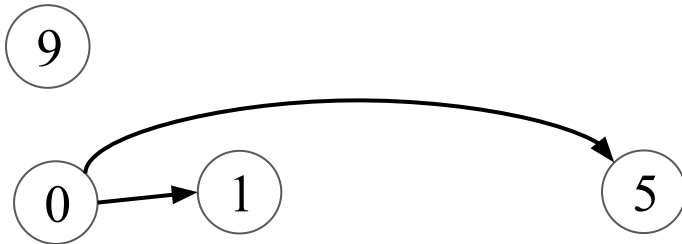


Ordenação Topológica

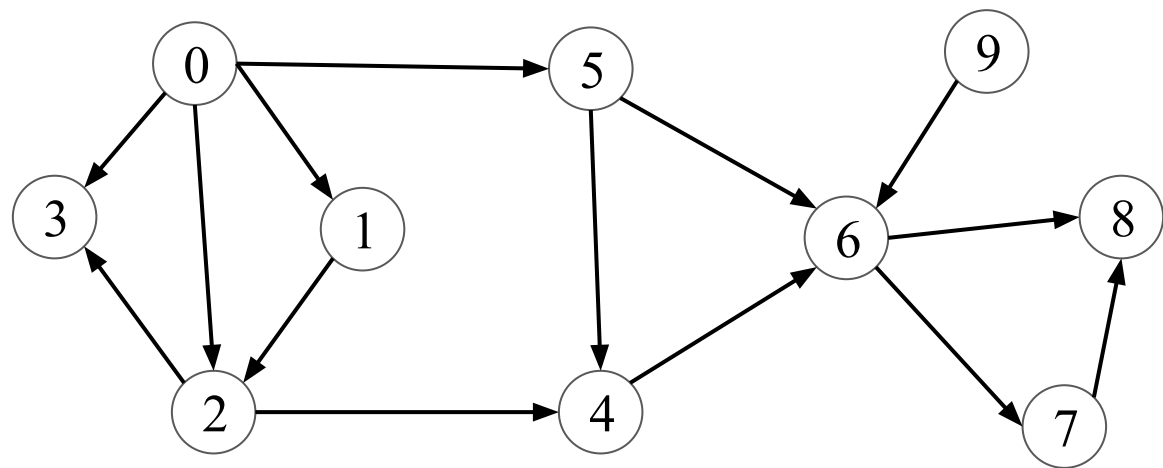


2
4
6
7
8
3

Pilha

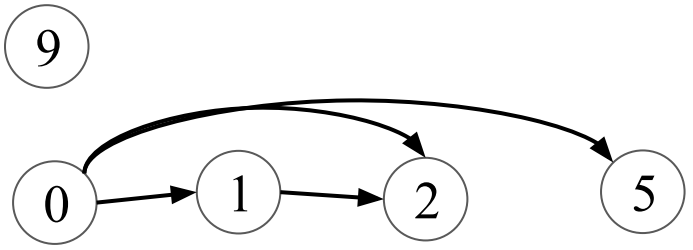


Ordenação Topológica

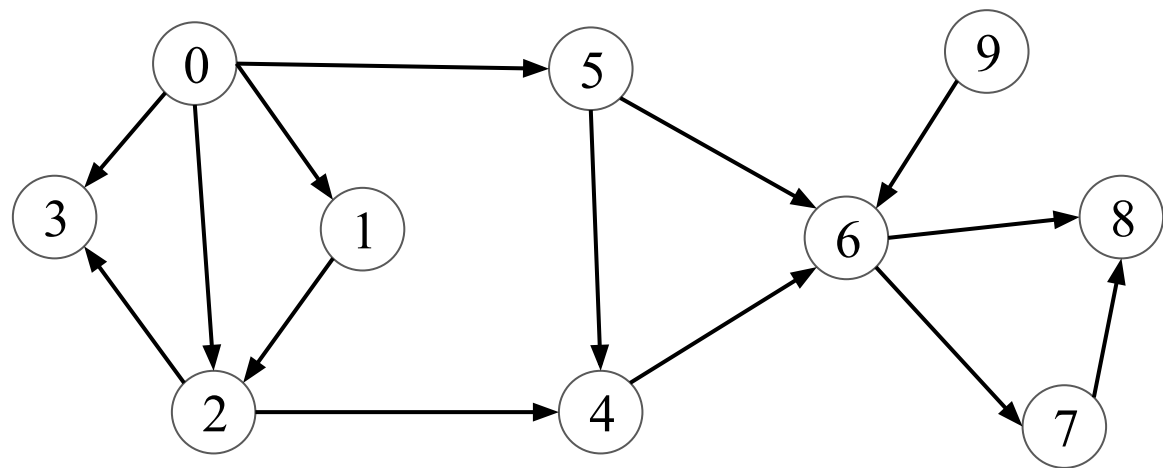


4
6
7
8
3

Pilha

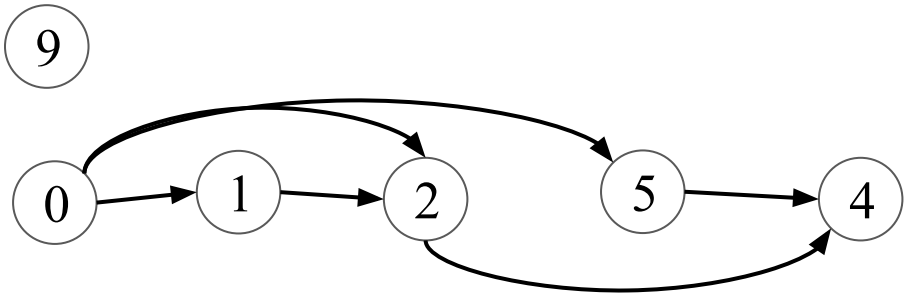


Ordenação Topológica

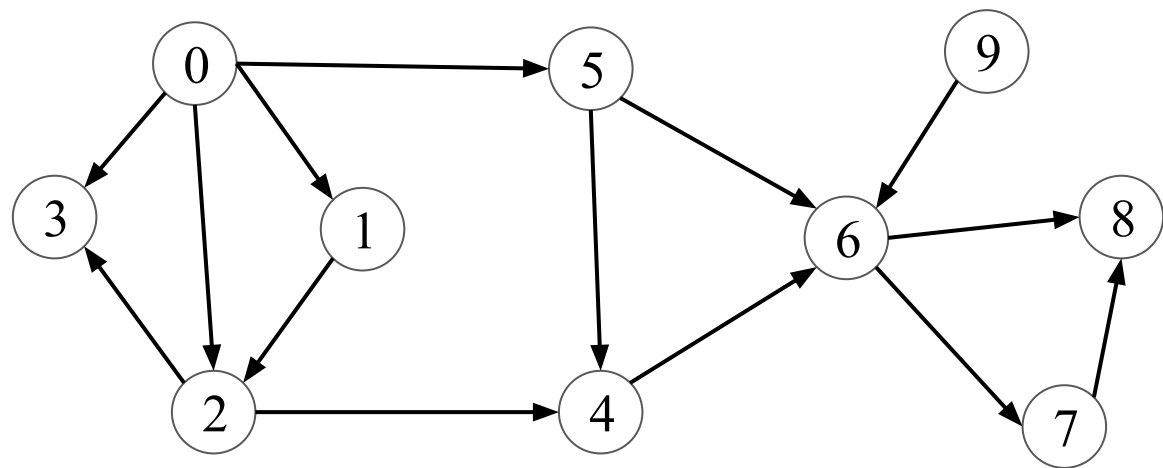


6
7
8
3

Pilha

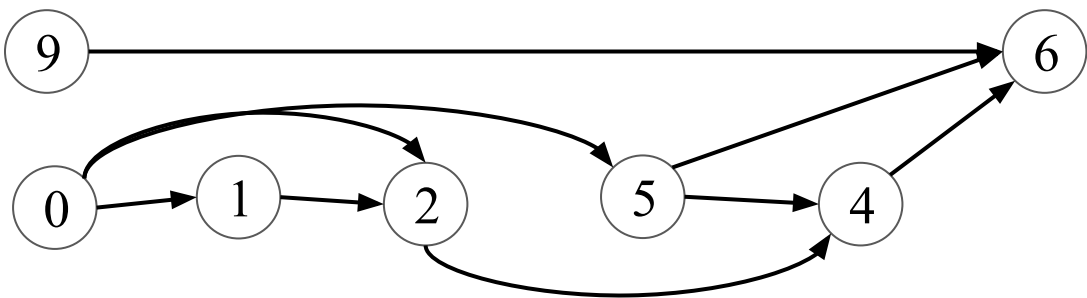


Ordenação Topológica

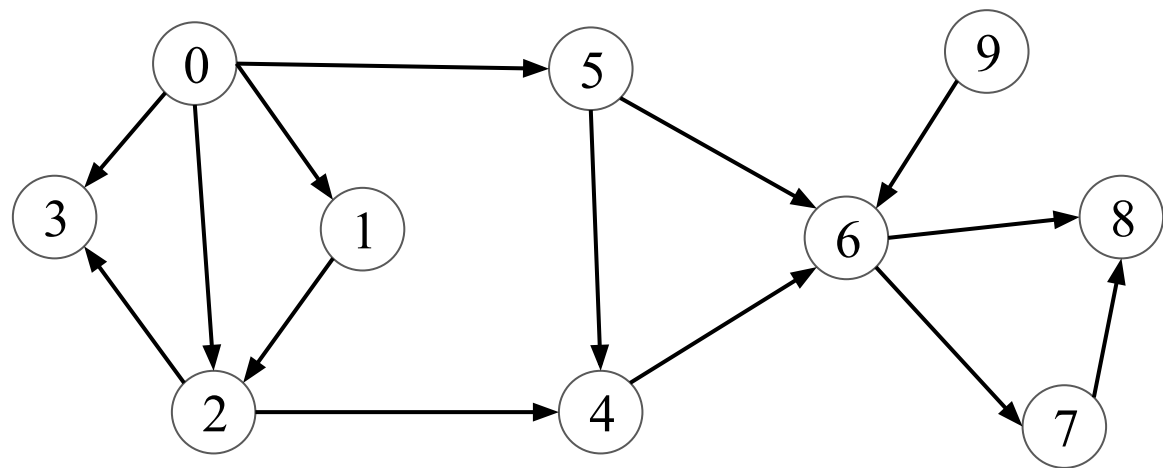


7
8
3

Pilha

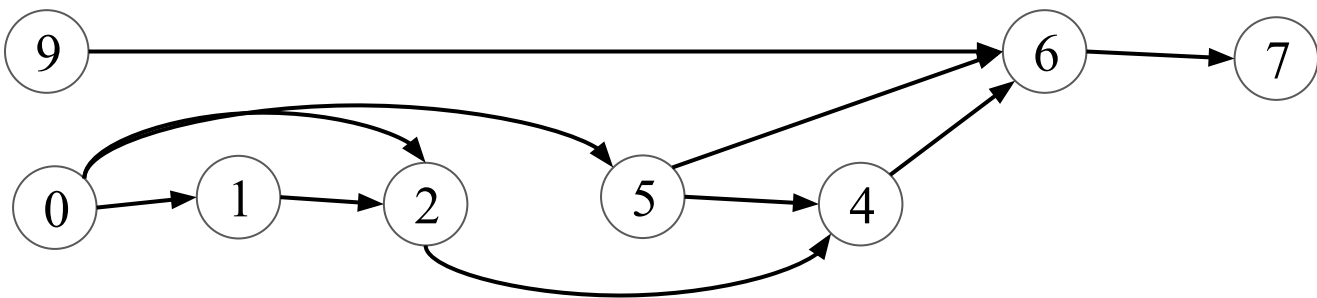


Ordenação Topológica

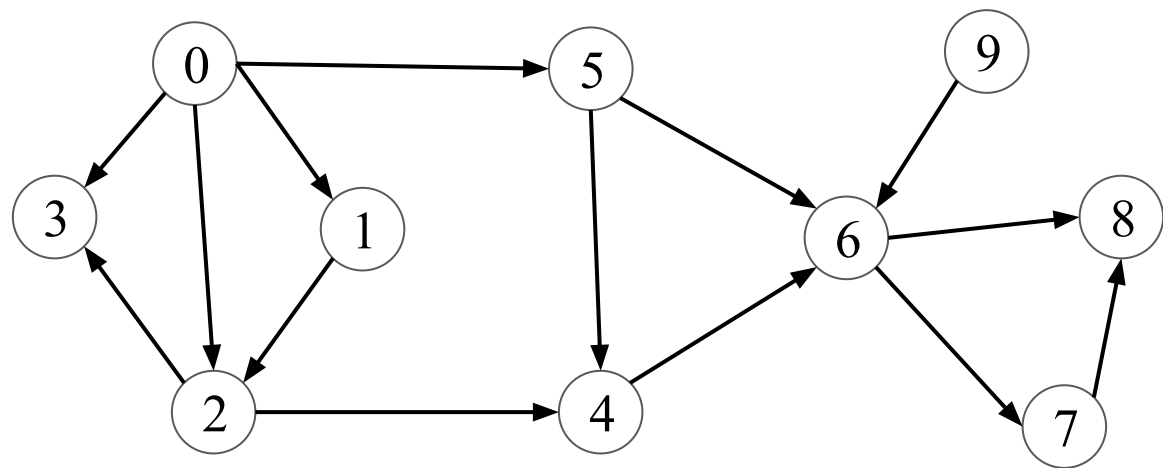


8
3

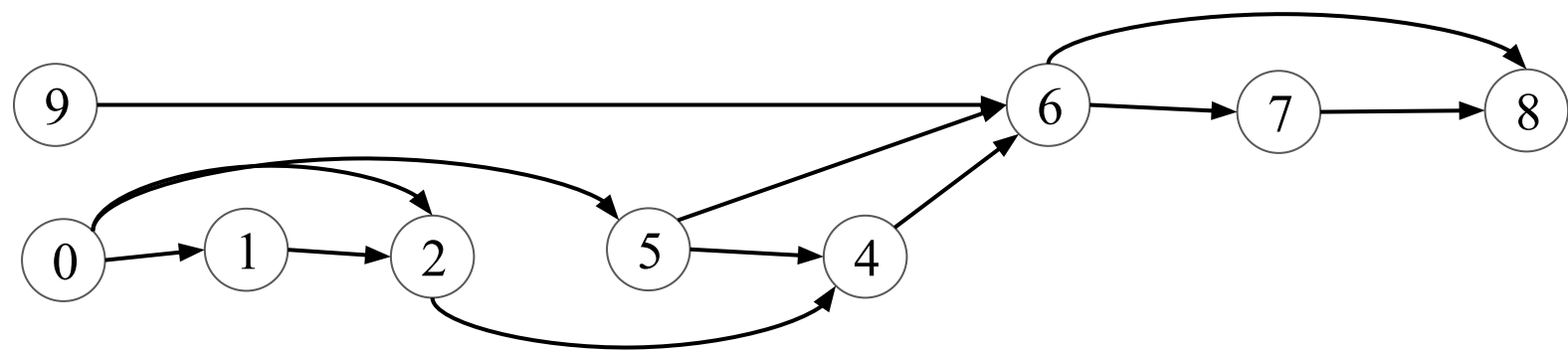
Pilha



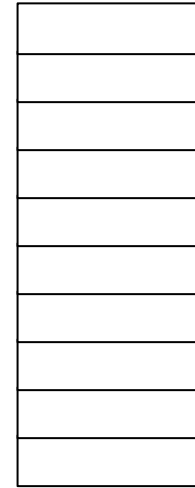
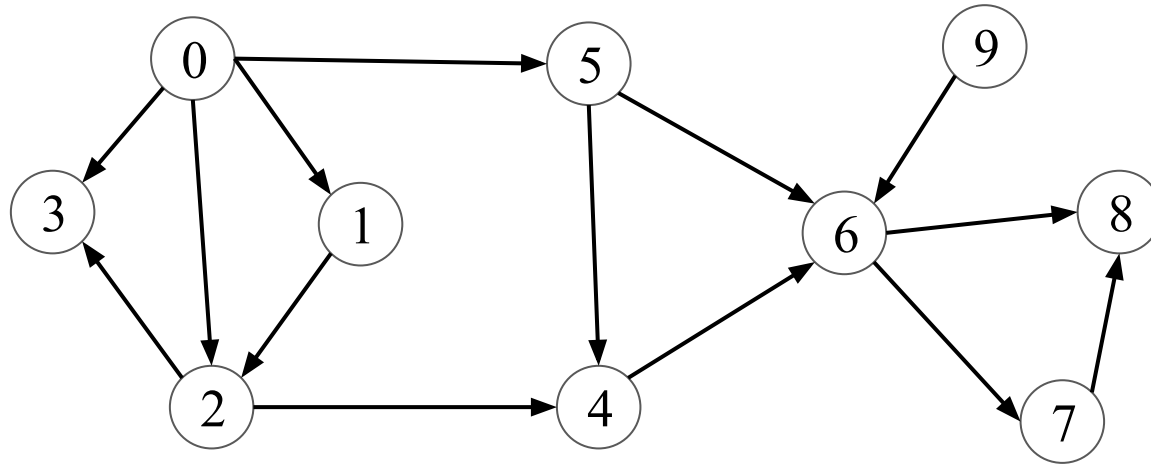
Ordenação Topológica



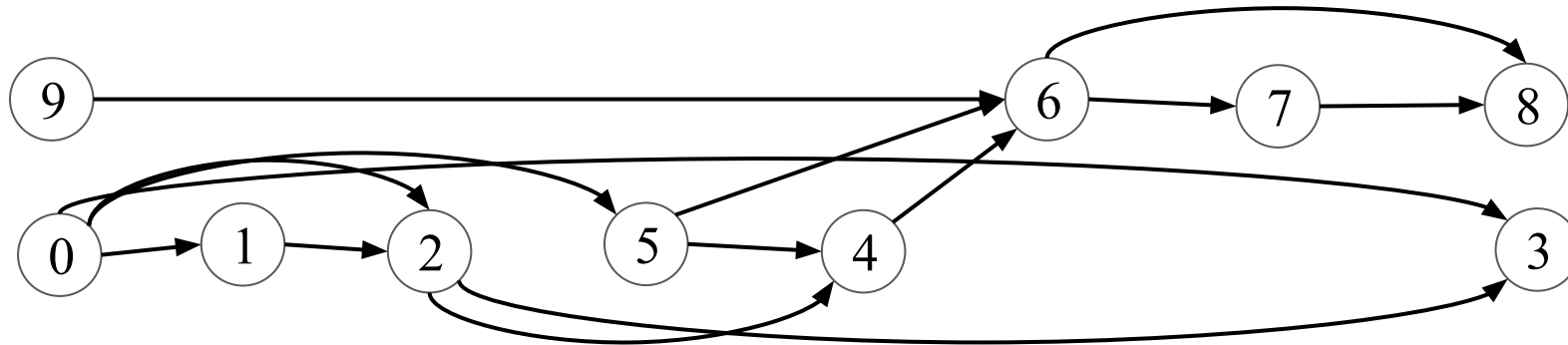
Pilha



Ordenação Topológica



Pilha

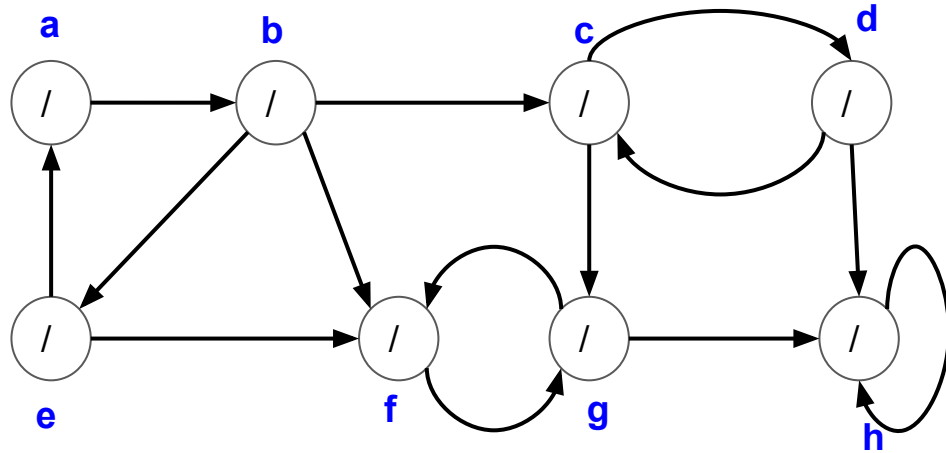


Componentes Conexas

- É um conjunto maximal $C \subseteq V$, tal que para todo par de vértices (u, v) em C , u e v são alcançáveis;
 - Usa-se o **transposto** de G
 - Seja $G = (V, E) \Rightarrow G^t = (V, E^t)$ onde $E^t = \{(u, v): (v, u) \in E\}$
1. Chama DFS(G) para obter o vetor f ;
 - a. Ordena f em ordem decrescente.
 2. Obtém G^t ;
 3. Chama DFS(G^t), realiza busca a partir do vértice de maior f ;
 4. Montar as componentes seguindo as arestas de árvore.

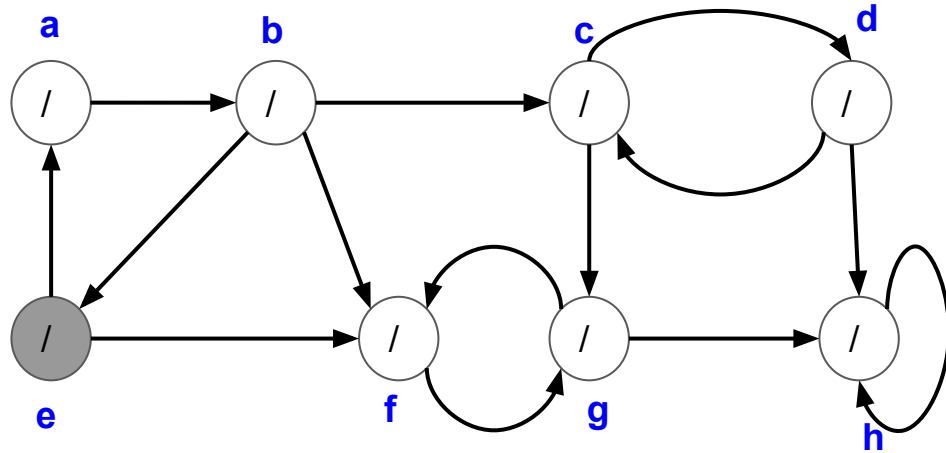
Componentes Conexas - Passo 1

1. Chama DFS(G) para obter o vetor f ;
 - a. Busca pode iniciar em qualquer vértice



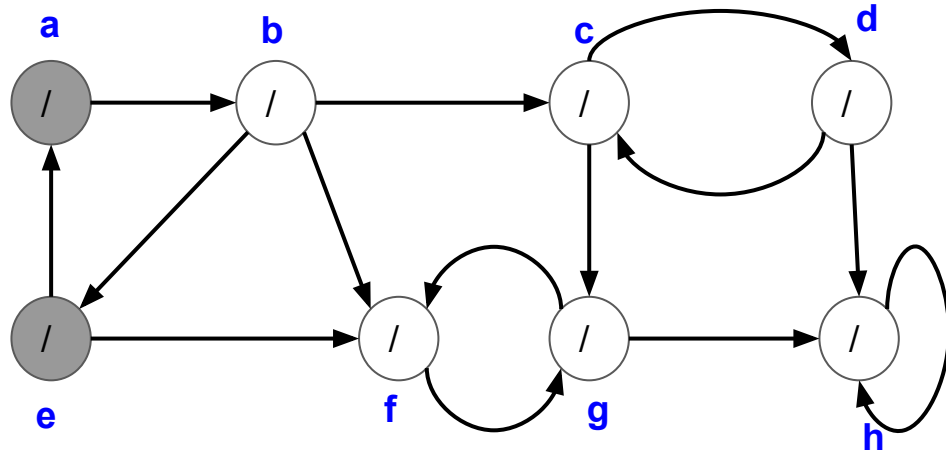
Componentes Conexas - Passo 1

1. Chama DFS(G) para obter o vetor f ;
 - a. Busca pode iniciar em qualquer vértice



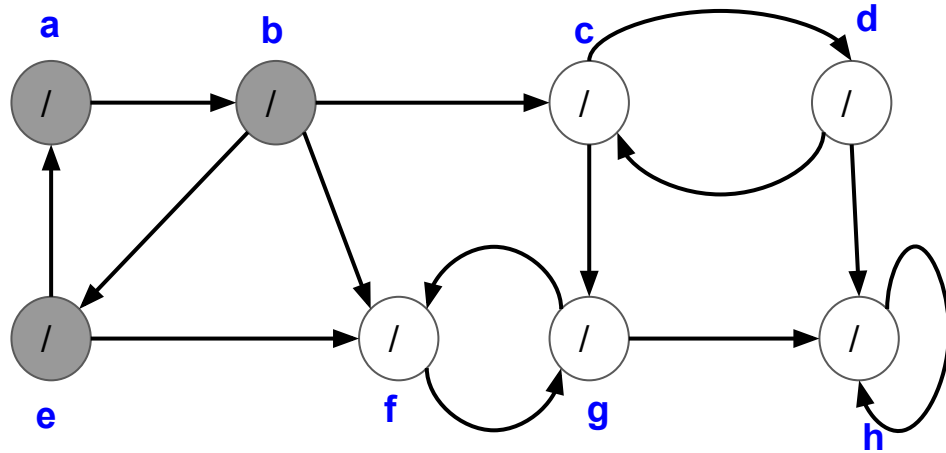
Componentes Conexas - Passo 1

1. Chama DFS(G) para obter o vetor f ;
 - a. Busca pode iniciar em qualquer vértice



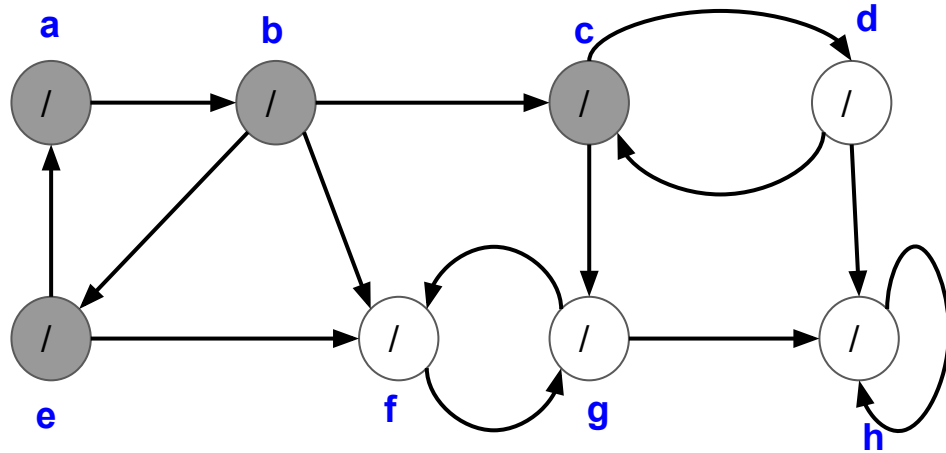
Componentes Conexas - Passo 1

1. Chama DFS(G) para obter o vetor f ;
 - a. Busca pode iniciar em qualquer vértice



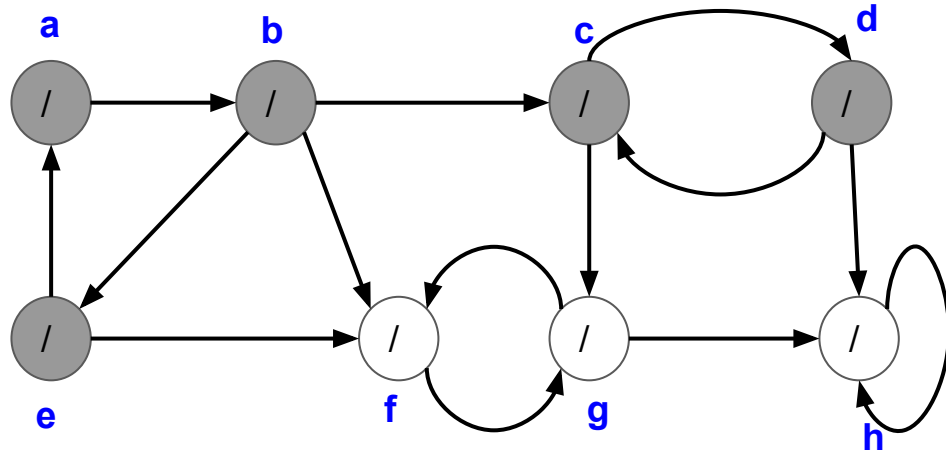
Componentes Conexas - Passo 1

1. Chama DFS(G) para obter o vetor f ;
 - a. Busca pode iniciar em qualquer vértice



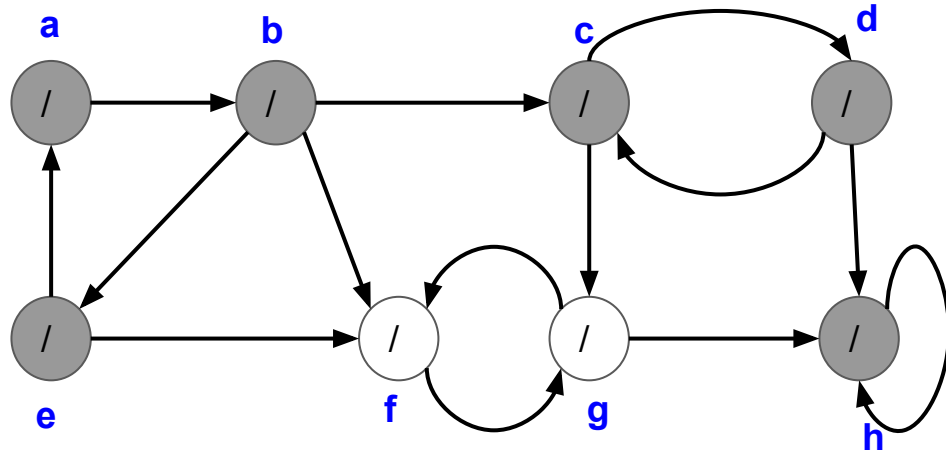
Componentes Conexas - Passo 1

1. Chama DFS(G) para obter o vetor f ;
 - a. Busca pode iniciar em qualquer vértice



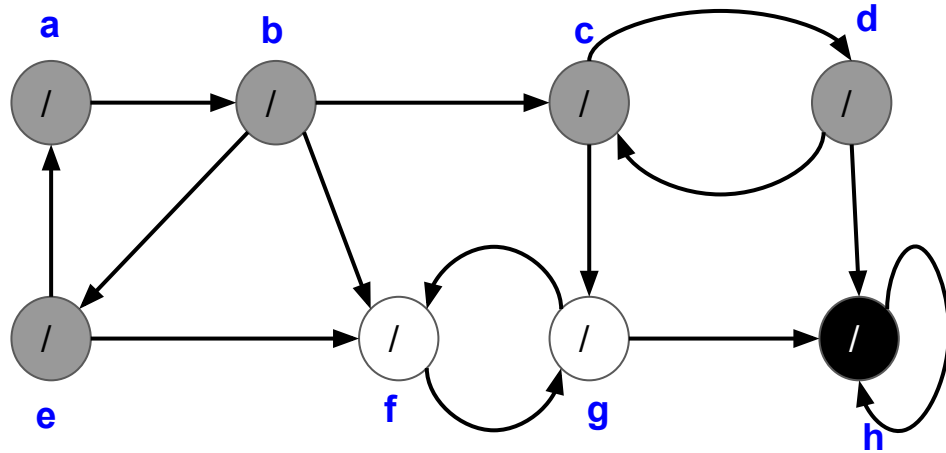
Componentes Conexas - Passo 1

1. Chama DFS(G) para obter o vetor f ;
 - a. Busca pode iniciar em qualquer vértice



Componentes Conexas - Passo 1

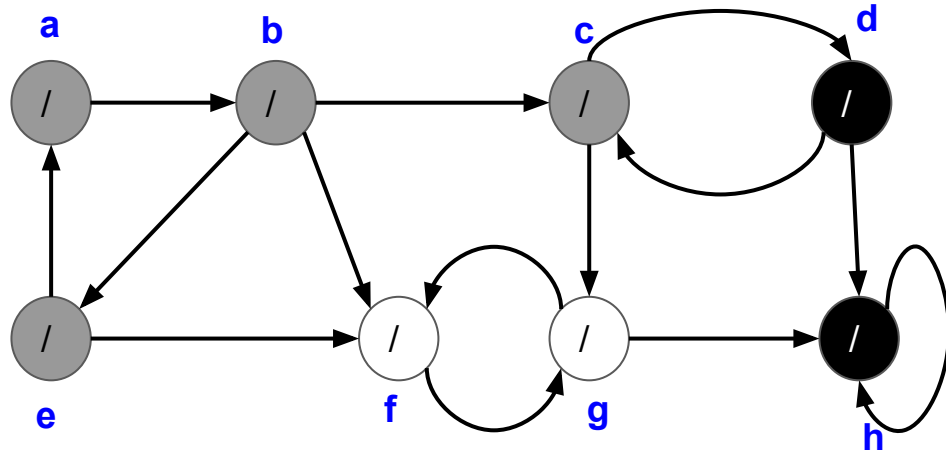
1. Chama DFS(G) para obter o vetor f ;
 - a. Busca pode iniciar em qualquer vértice



h

Componentes Conexas - Passo 1

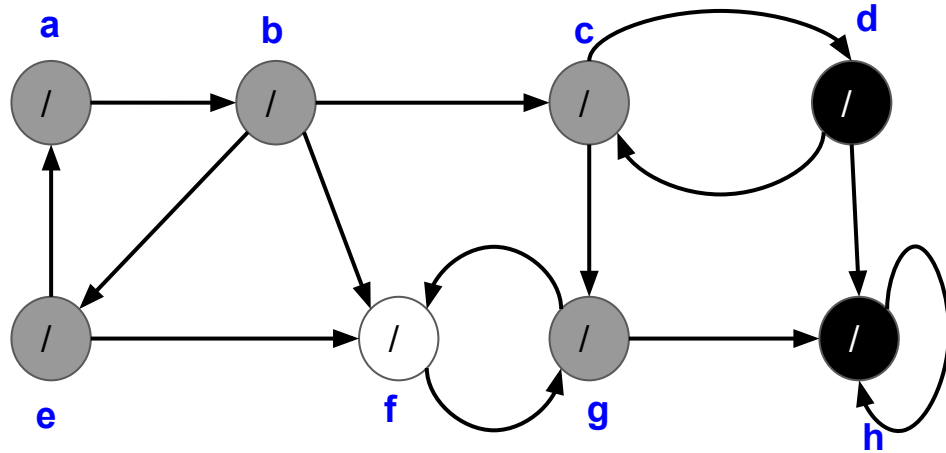
1. Chama DFS(G) para obter o vetor f ;
 - a. Busca pode iniciar em qualquer vértice



h d

Componentes Conexas - Passo 1

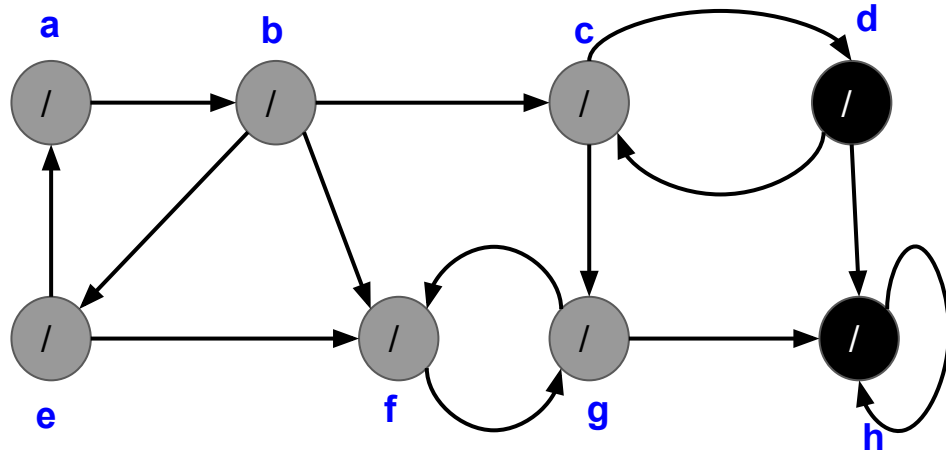
1. Chama DFS(G) para obter o vetor f ;
 - a. Busca pode iniciar em qualquer vértice



h d

Componentes Conexas - Passo 1

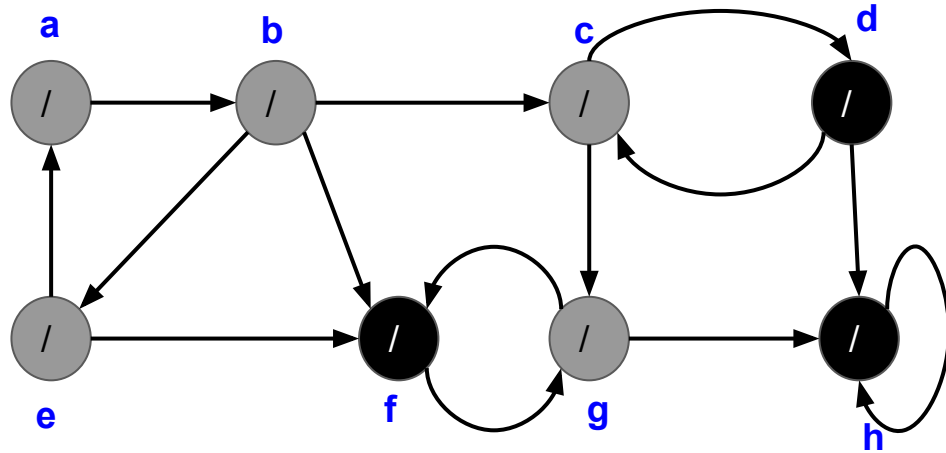
1. Chama DFS(G) para obter o vetor f ;
 - a. Busca pode iniciar em qualquer vértice



h d

Componentes Conexas - Passo 1

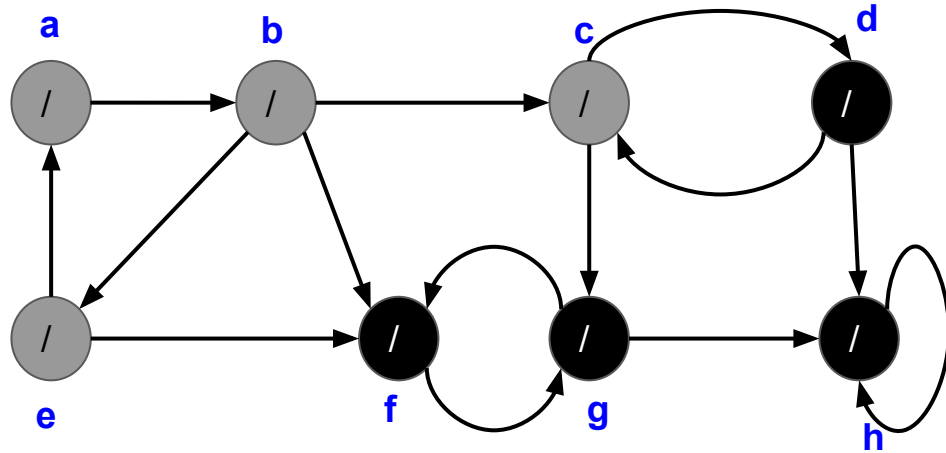
1. Chama DFS(G) para obter o vetor f ;
 - a. Busca pode iniciar em qualquer vértice



h d f

Componentes Conexas - Passo 1

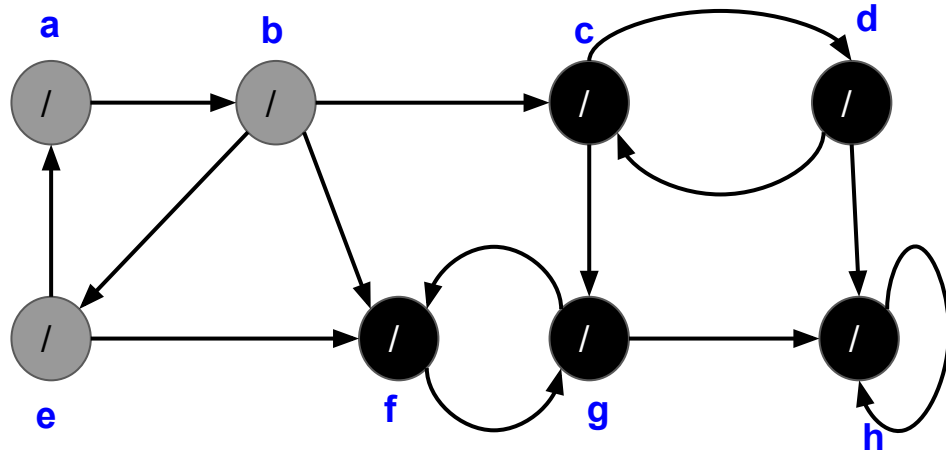
1. Chama DFS(G) para obter o vetor f ;
 - a. Busca pode iniciar em qualquer vértice



h d f g

Componentes Conexas - Passo 1

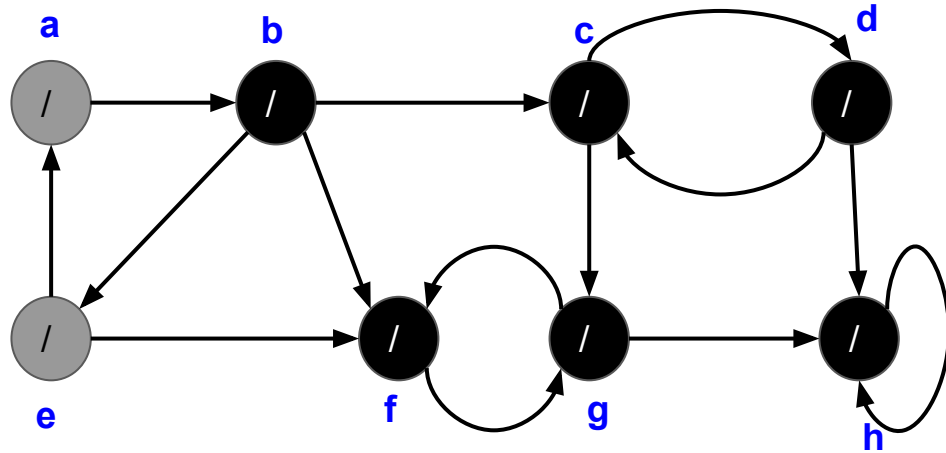
1. Chama DFS(G) para obter o vetor f ;
 - a. Busca pode iniciar em qualquer vértice



h d f g c

Componentes Conexas - Passo 1

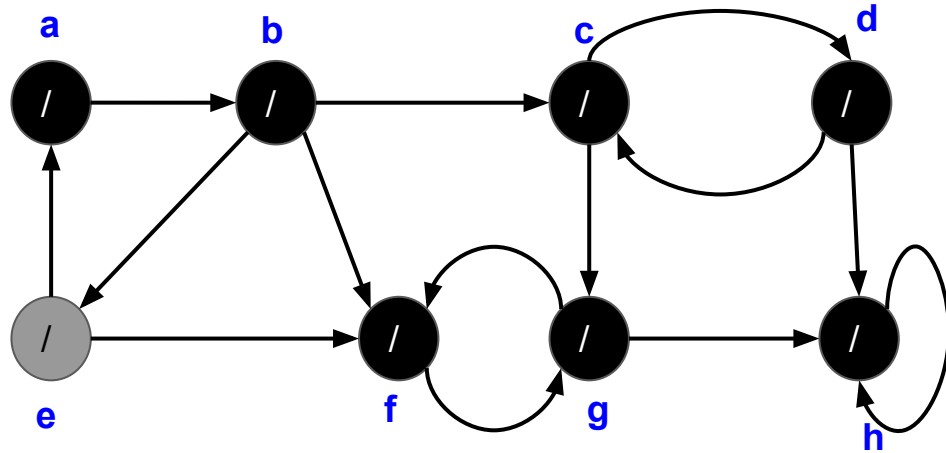
1. Chama DFS(G) para obter o vetor f ;
 - a. Busca pode iniciar em qualquer vértice



h d f g c b

Componentes Conexas - Passo 1

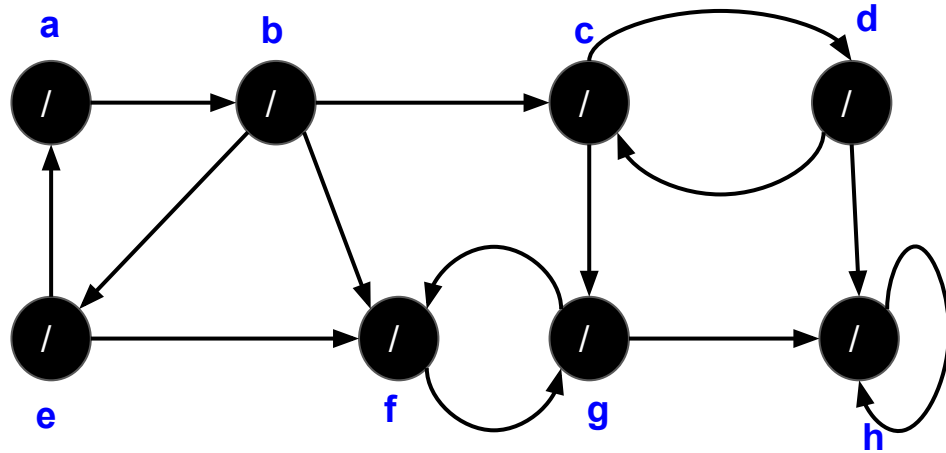
1. Chama DFS(G) para obter o vetor f ;
 - a. Busca pode iniciar em qualquer vértice



h d f g c b a

Componentes Conexas - Passo 1

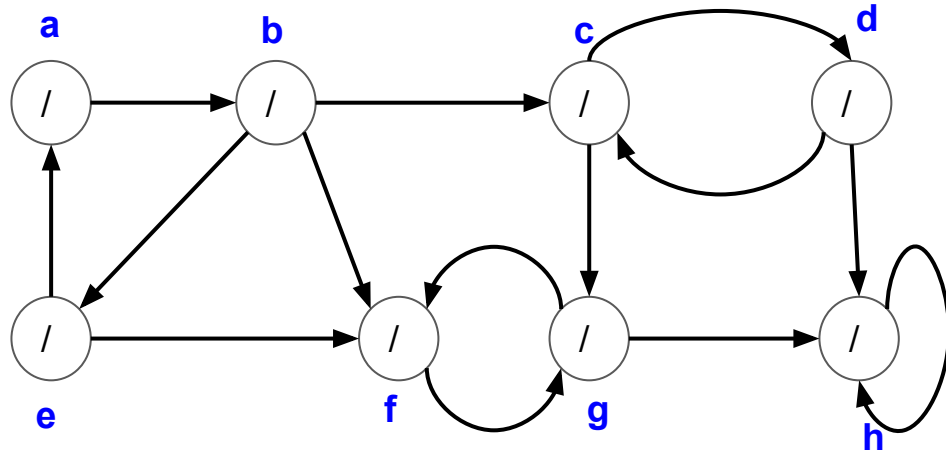
1. Chama DFS(G) para obter o vetor f ;
 - a. Busca pode iniciar em qualquer vértice



h d f g c b a e

Componentes Conexas - Passo 1

1. Chama DFS(G) para obter o vetor f ;
 - a. Busca pode iniciar em qualquer vértice \rightarrow Ordena f em ordem decrescente

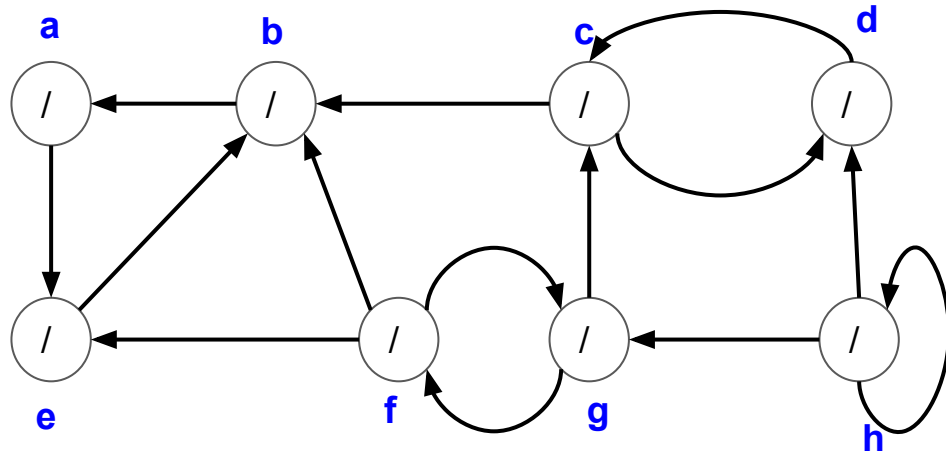


h d f g c b a e



Componentes Conexas - Passo 2

➤ Obter G^t ;

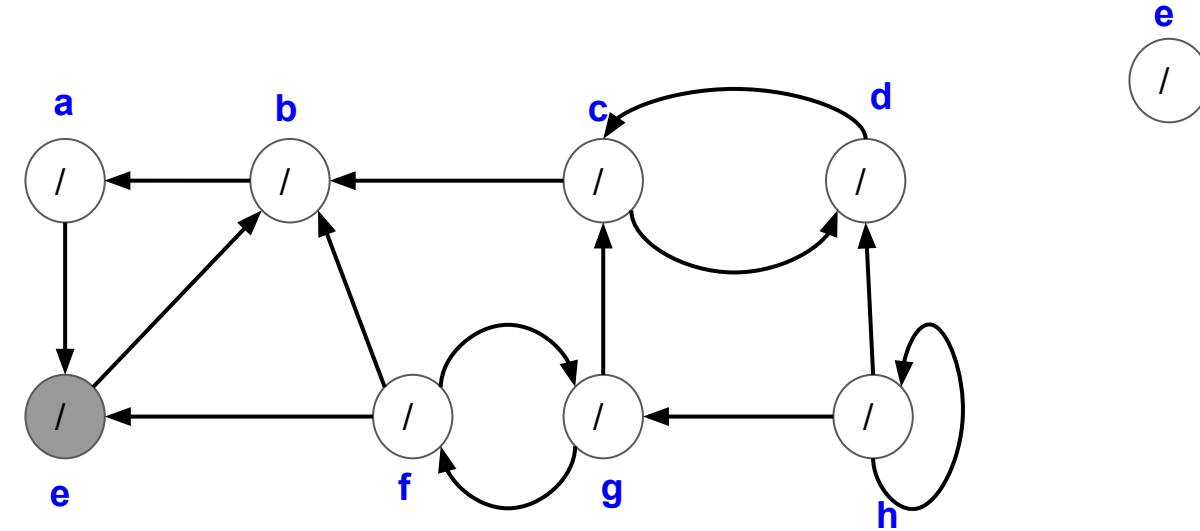


h d f g c b a e



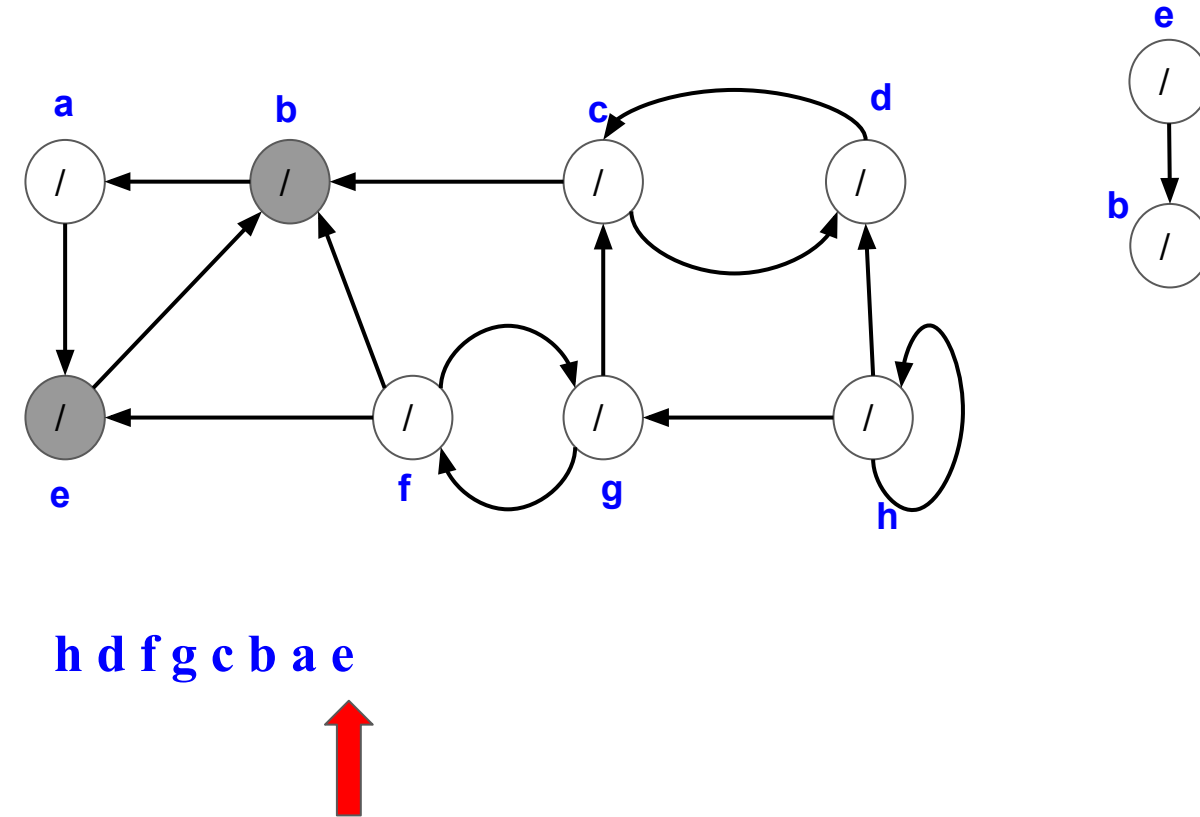
Componentes Conexas - Passo 3

➤ Chama $\text{DFS}(G^t)$, realiza busca a partir do vértice de maior f ;



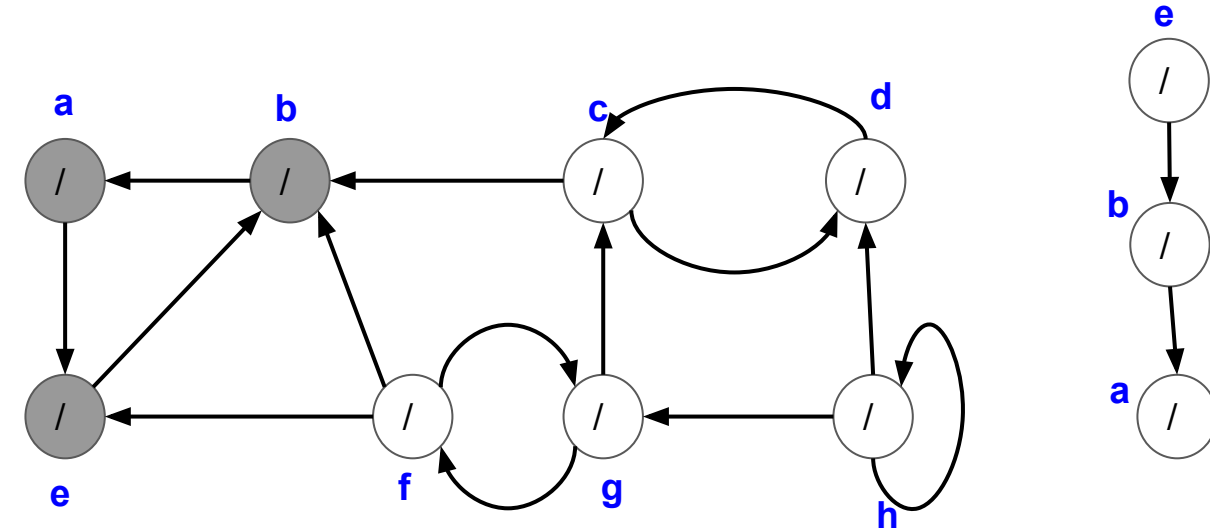
Componentes Conexas - Passo 3

➤ Chama $\text{DFS}(G^t)$, realiza busca a partir do vértice de maior f ;



Componentes Conexas - Passo 3

➤ Chama $\text{DFS}(G^t)$, realiza busca a partir do vértice de maior f ;

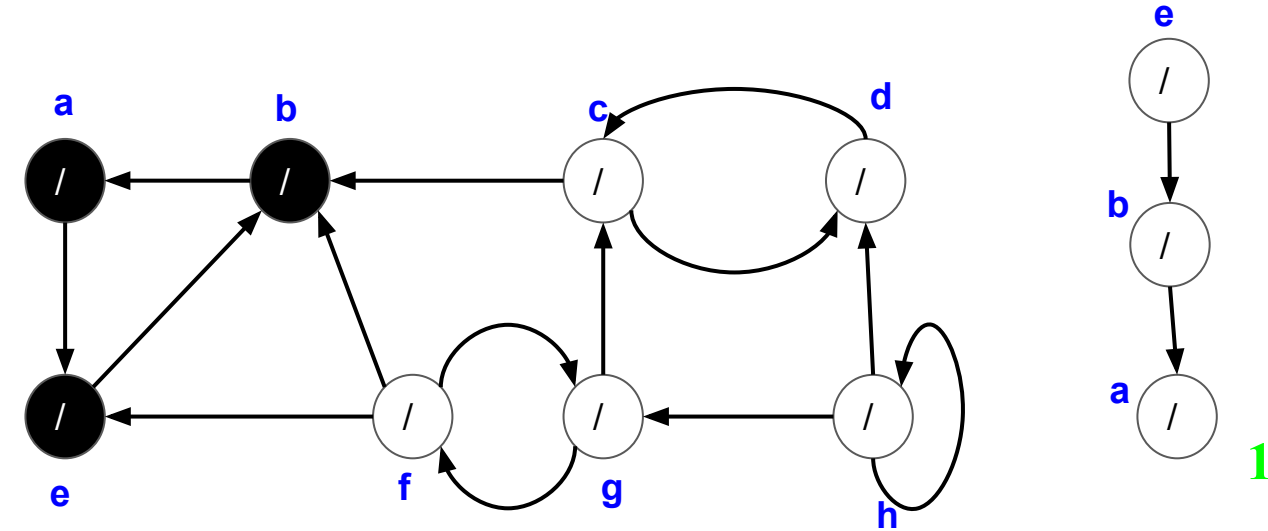


h d f g c b a e



Componentes Conexas - Passo 3

➤ Chama $\text{DFS}(G^t)$, realiza busca a partir do vértice de maior f ;

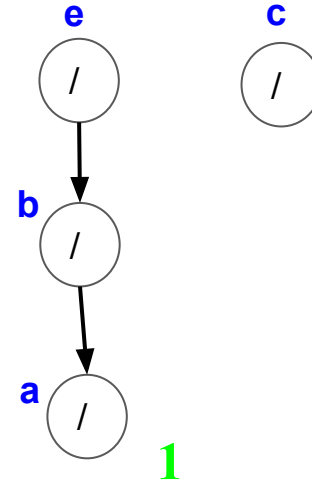
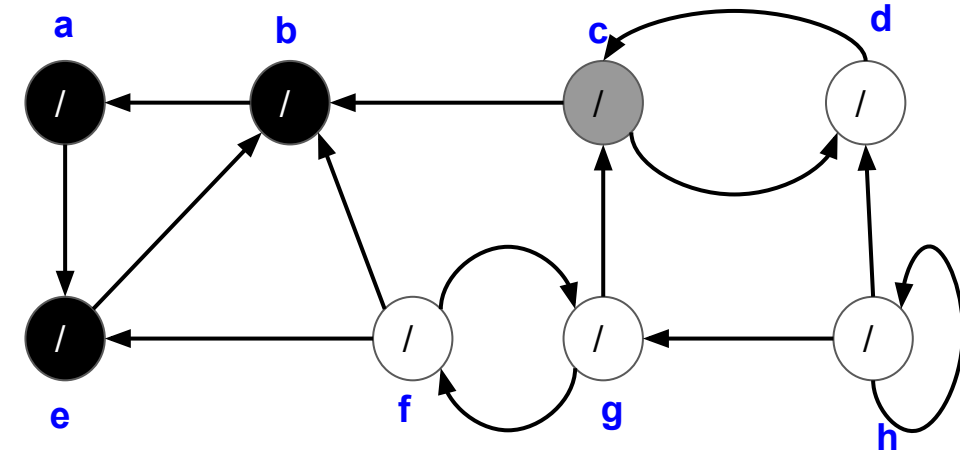


h d f g c b a e

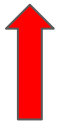


Componentes Conexas - Passo 3

➤ Chama $\text{DFS}(G^t)$, realiza busca a partir do vértice de maior f ;

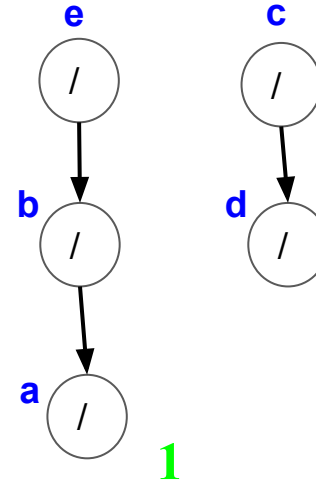
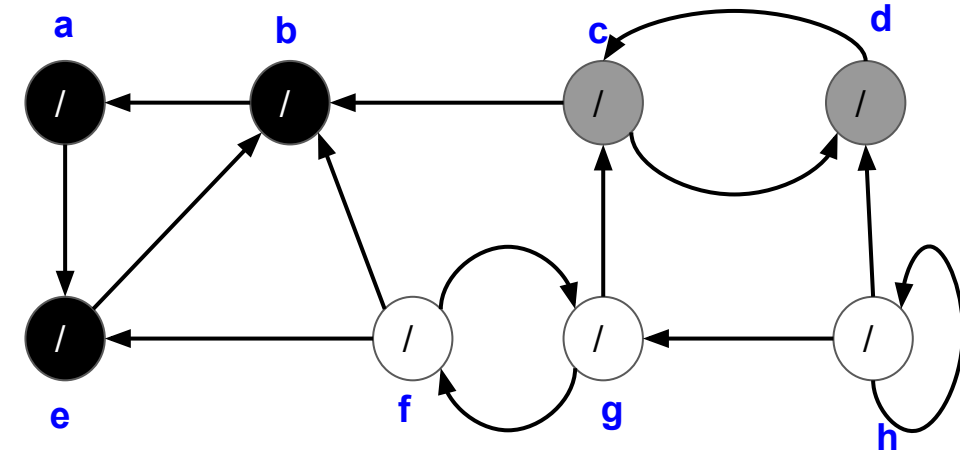


h d f g c b a e

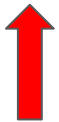


Componentes Conexas - Passo 3

➤ Chama $\text{DFS}(G^t)$, realiza busca a partir do vértice de maior f ;

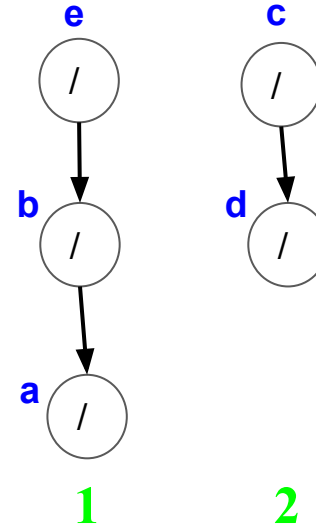
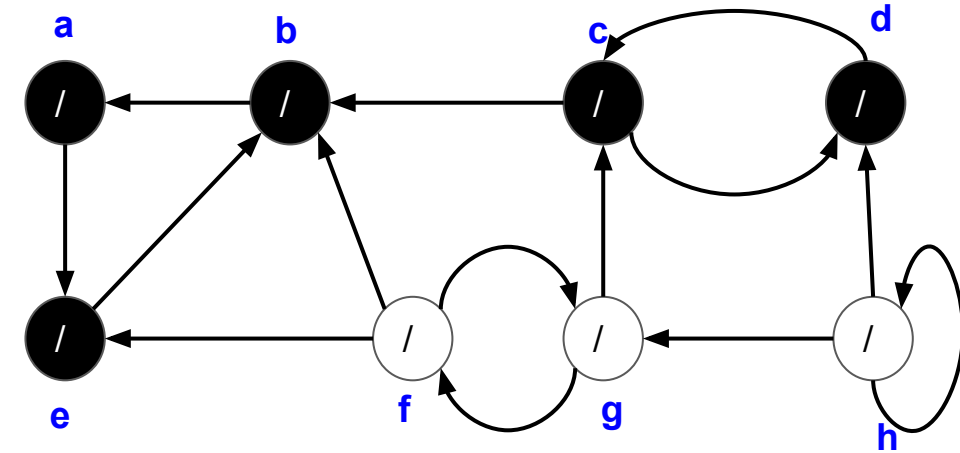


h d f g c b a e

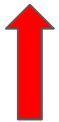


Componentes Conexas - Passo 3

➤ Chama $\text{DFS}(G^t)$, realiza busca a partir do vértice de maior f ;

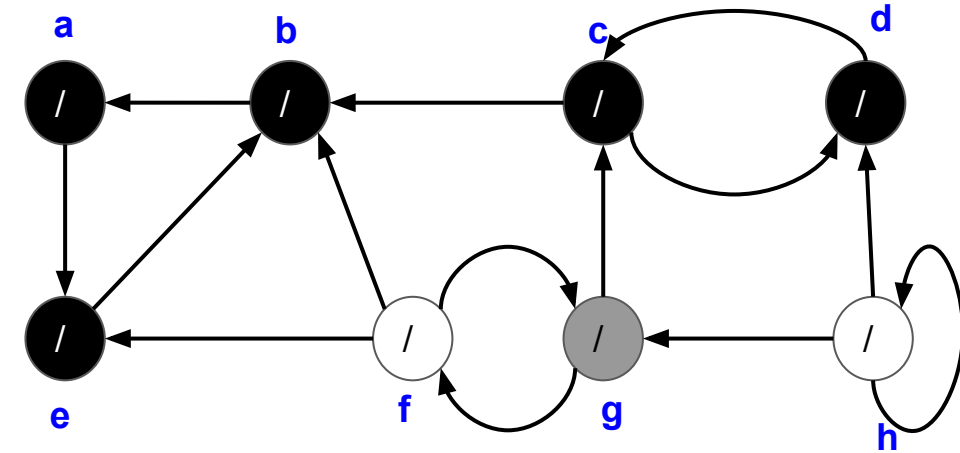


h d f g c b a e

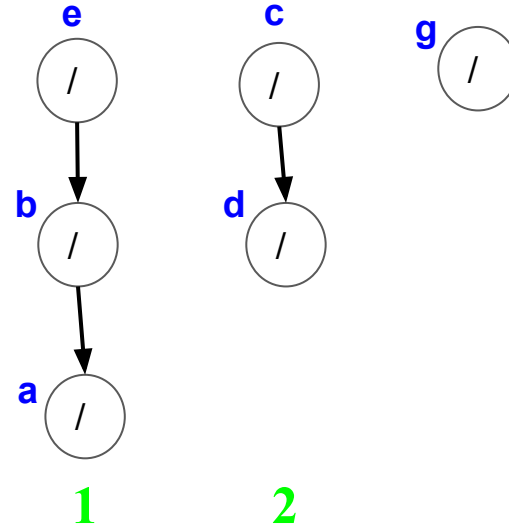


Componentes Conexas - Passo 3

➤ Chama $\text{DFS}(G^t)$, realiza busca a partir do vértice de maior f ;

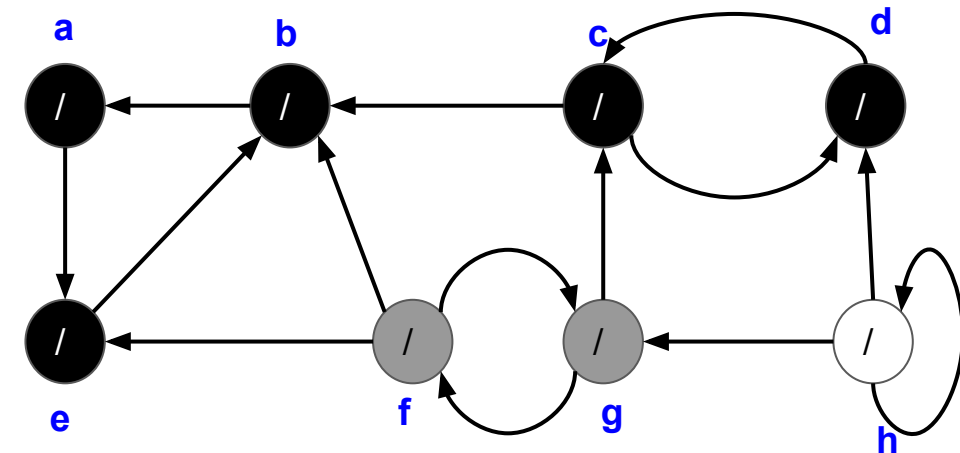


h d f g c b a e

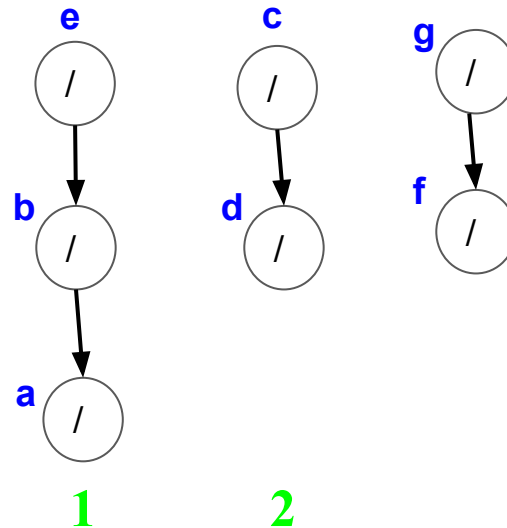


Componentes Conexas - Passo 3

➤ Chama $\text{DFS}(G^t)$, realiza busca a partir do vértice de maior f ;

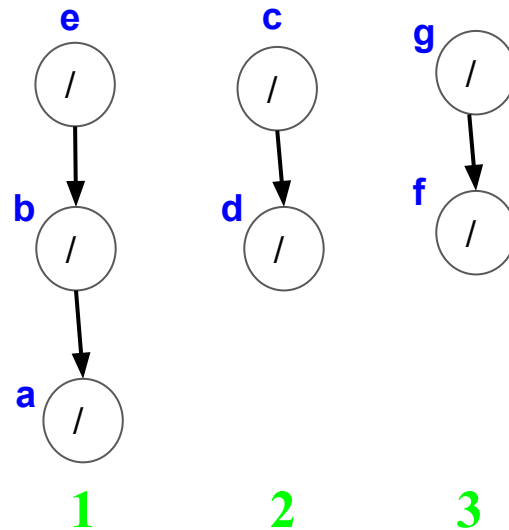
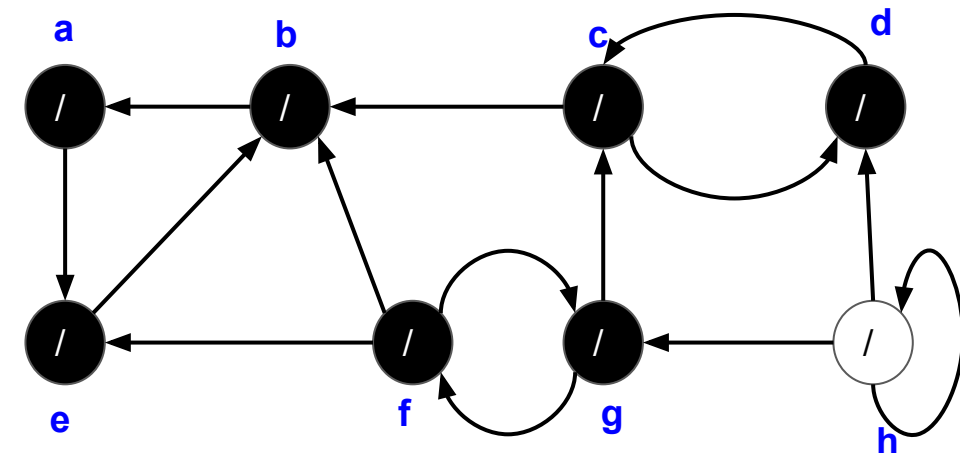


h d f g c b a e



Componentes Conexas - Passo 3

➤ Chama $\text{DFS}(G^t)$, realiza busca a partir do vértice de maior f ;

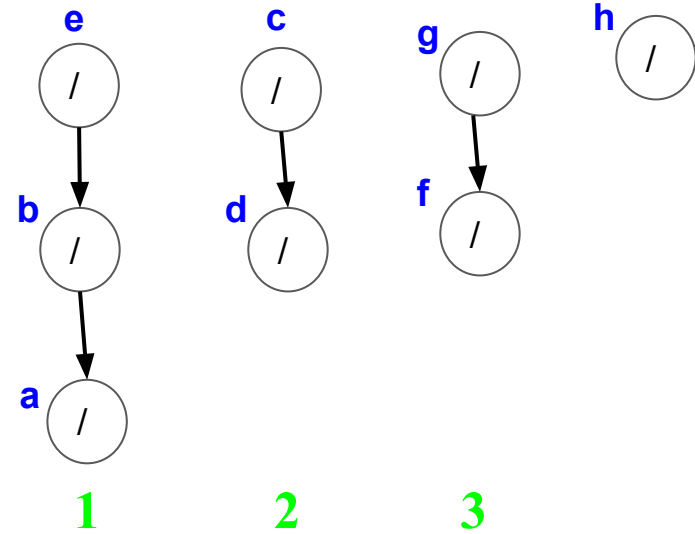
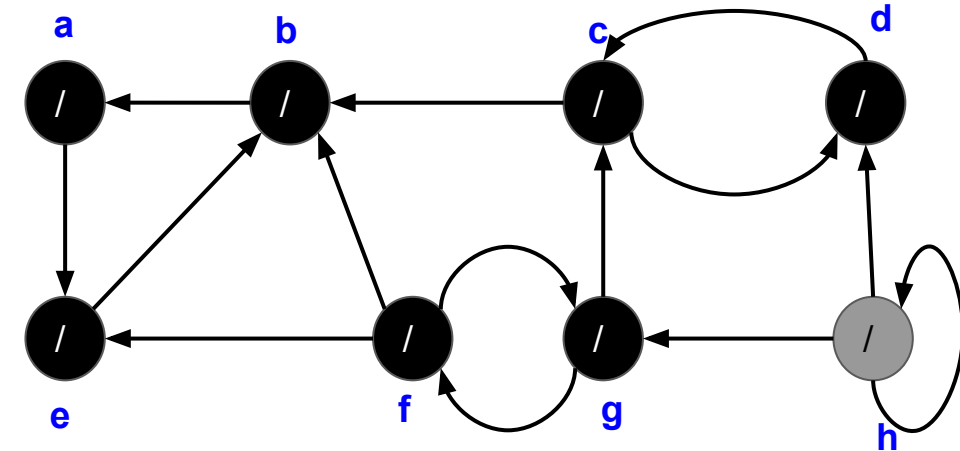


h d f g c b a e



Componentes Conexas - Passo 3

➤ Chama $\text{DFS}(G^t)$, realiza busca a partir do vértice de maior f ;

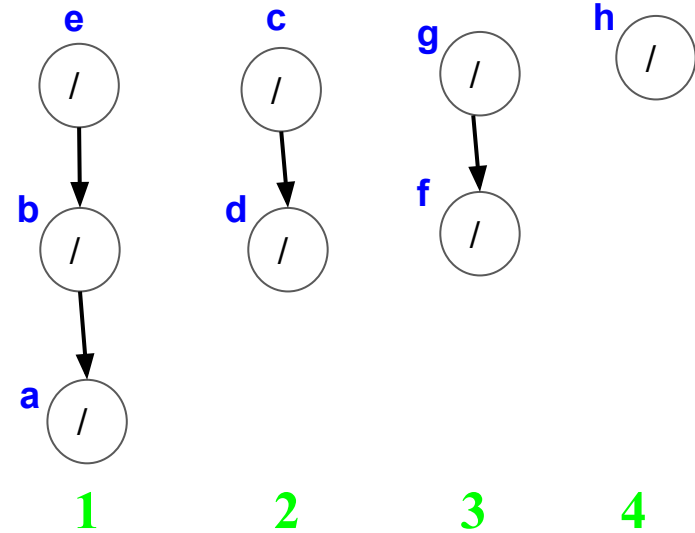
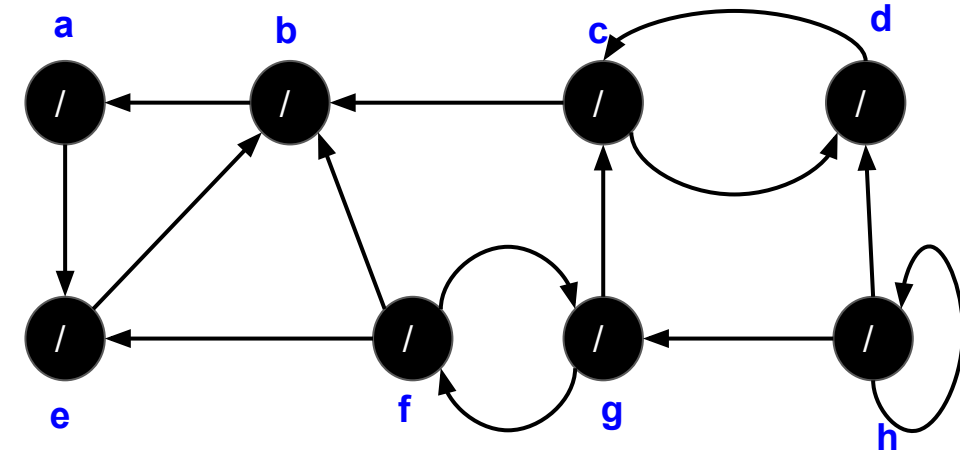


h d f g c b a e



Componentes Conexas - Passo 3

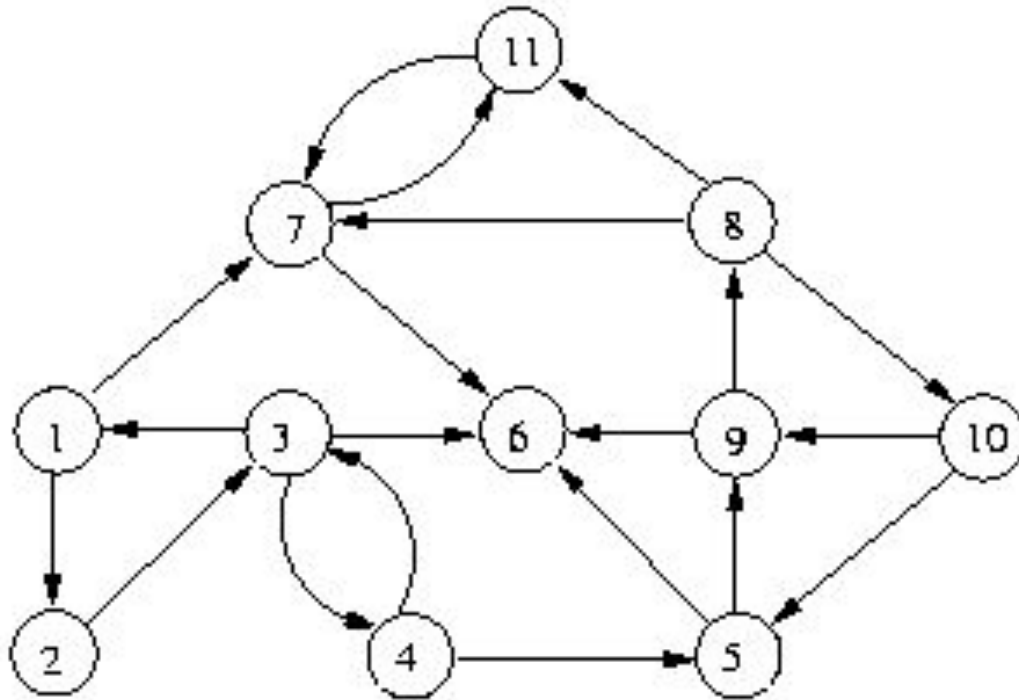
➤ Chama $\text{DFS}(G^t)$, realiza busca a partir do vértice de maior f ;



h d f g c b a e



Componentes Conexas - Exercício





UNIVERSIDADE
FEDERAL DO CEARÁ

CAMPUS DE RUSSAS

Algoritmos em Grafos

Aula 10: Ordenação Topológica & Componentes Conexas

Professor Pablo Soares

2022.1