

ESTRUTURAS DE DADOS III

Grafos – Aplicações dos Algoritmos de Busca

Profa. Elaine Parros Machado de Sousa

adaptações: Cristina Dutra de Aguiar Ciferri

Material baseado em aulas dos professores:
Gustavo Batista, Robson Cordeiro, Moacir Ponti Jr. e
Maria Cristina Oliveira, Thiago A.S. Pardo

Teste de Grafo Cíclico/Acíclico: Busca em Profundidade

○ Busca em profundidade

- pode ser usada para verificar se um grafo é acíclico ou contém um ou mais ciclos

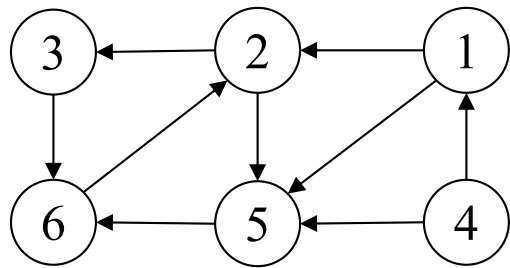
○ Grafo G é **acíclico**

- se e somente se a busca em profundidade em G **não** apresentar **arestas de retorno**.

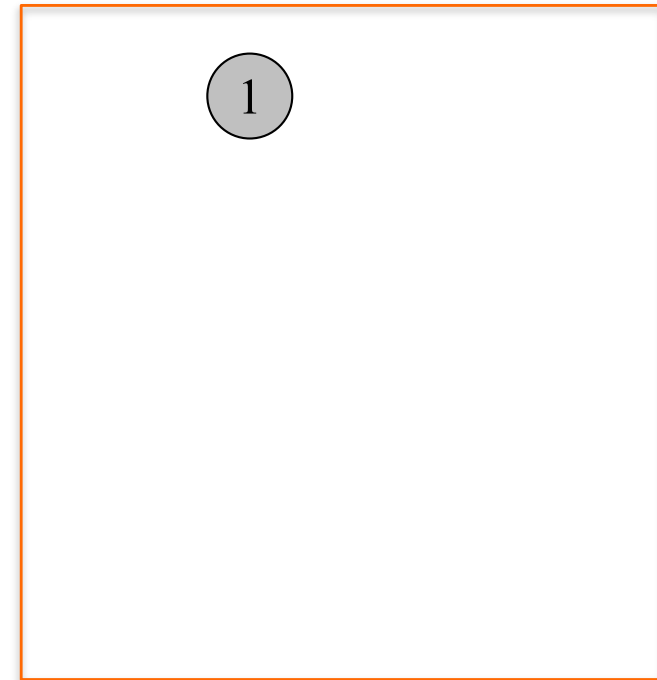
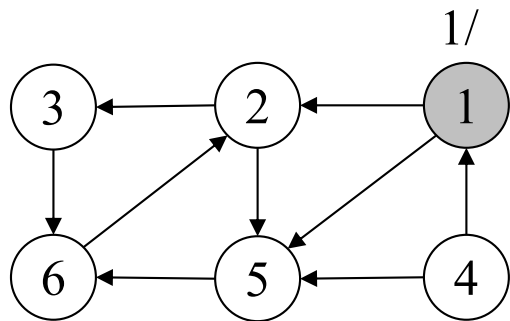
○ Em termos do algoritmo

- verificar se um vértice w adjacente a v possui **cor cinza** na primeira vez que a aresta (v, w) é percorrida.

Teste de Grafo Cíclico/Acíclico: Exemplo



Teste de Grafo Cíclico/Acíclico: Exemplo



árvore de busca
em profundidade

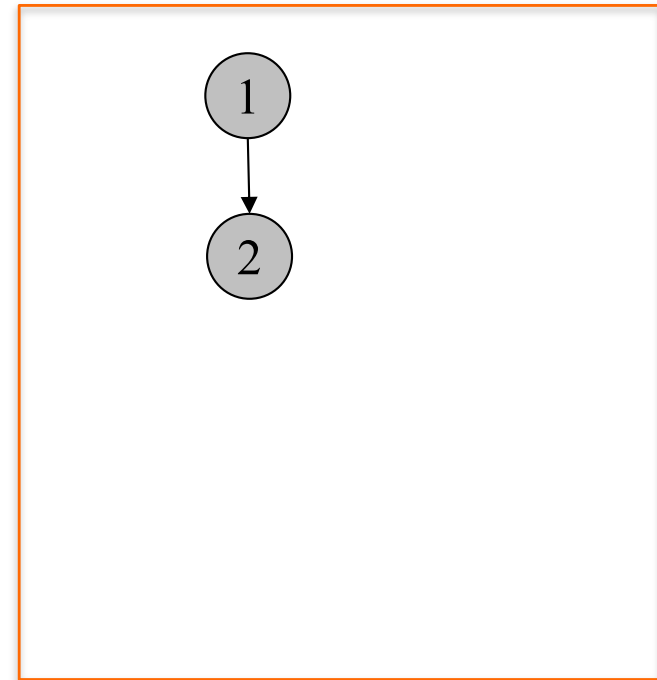
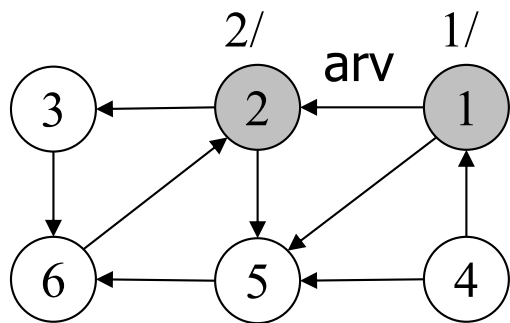
Vértice origem: 1

Tempo de descoberta: 1

Ação: vértice 1 torna-se cinza

Tempo de término: -

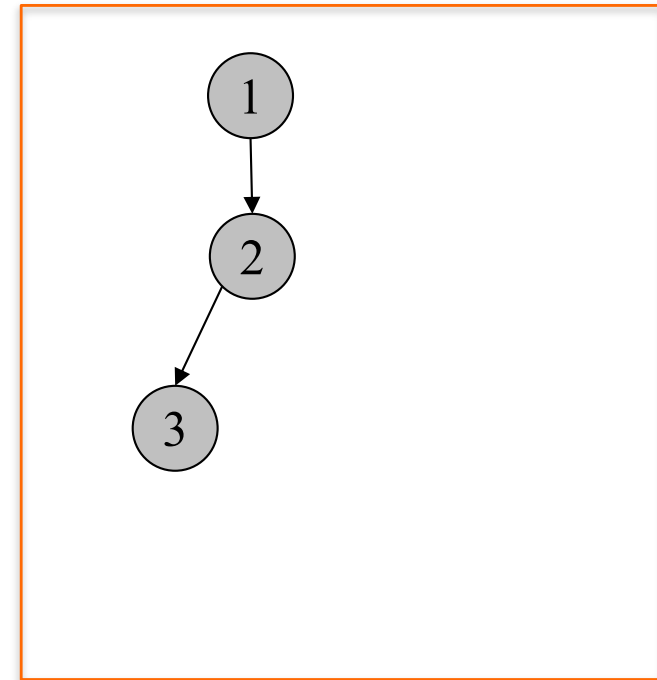
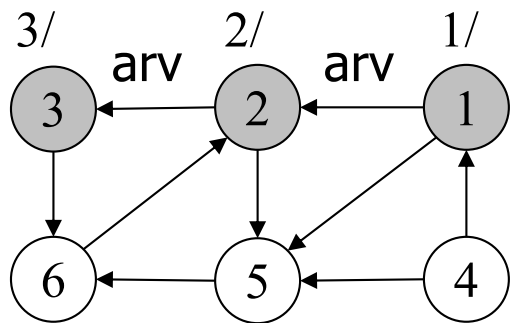
Teste de Grafo Cíclico/Acíclico: Exemplo



árvore de busca
em profundidade

Primeiro vértice não descoberto adjacente a 1: 2
Tempo de descoberta: 2
Ação: vértice 2 torna-se cinza
Tempo de término:

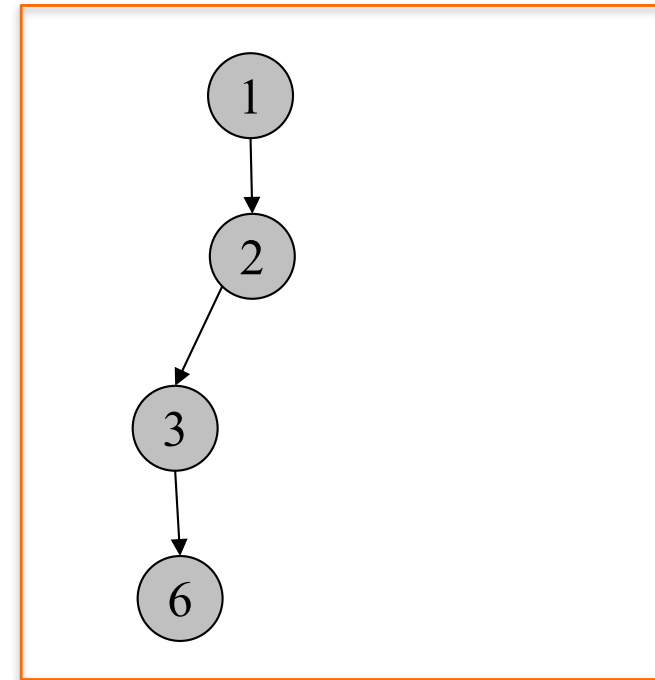
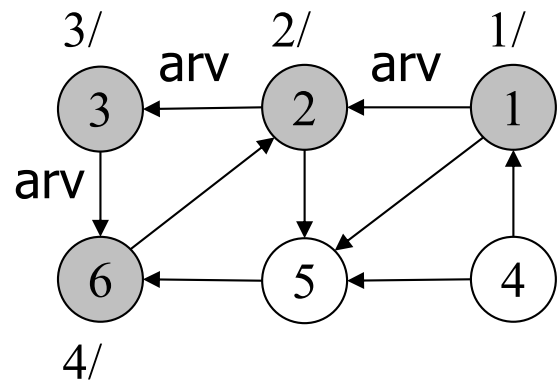
Teste de Grafo Cíclico/Acíclico: Exemplo



árvore de busca
em profundidade

Primeiro vértice não descoberto adjacente a 2: 3
Tempo de descoberta: 3
Ação: vértice 3 torna-se cinza
Tempo de término:

Teste de Grafo Cíclico/Acíclico: Exemplo



árvore de busca
em profundidade

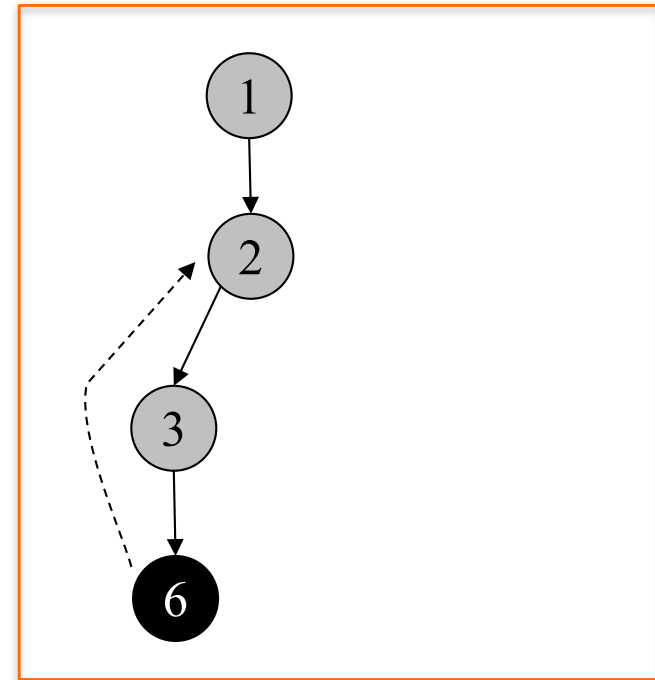
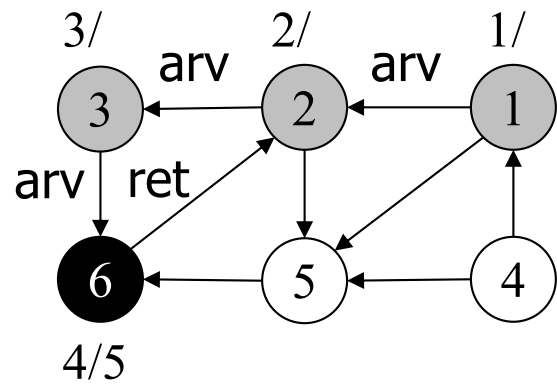
Primeiro vértice não descoberto adjacente a 3: 6

Tempo de descoberta: 4

Ação: vértice 6 torna-se cinza

Tempo de término:

Teste de Grafo Cíclico/Acíclico: Exemplo



árvore de busca
em profundidade

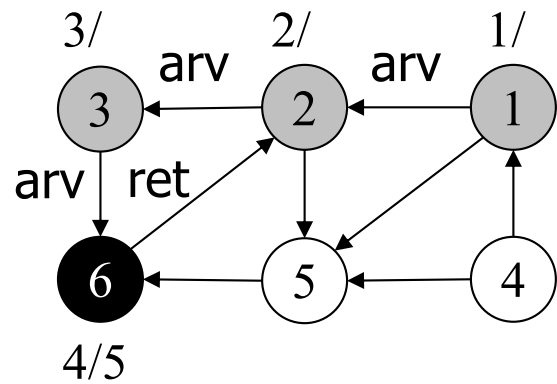
Primeiro vértice não descoberto adjacente a 6: nenhum

Tempo de descoberta: -

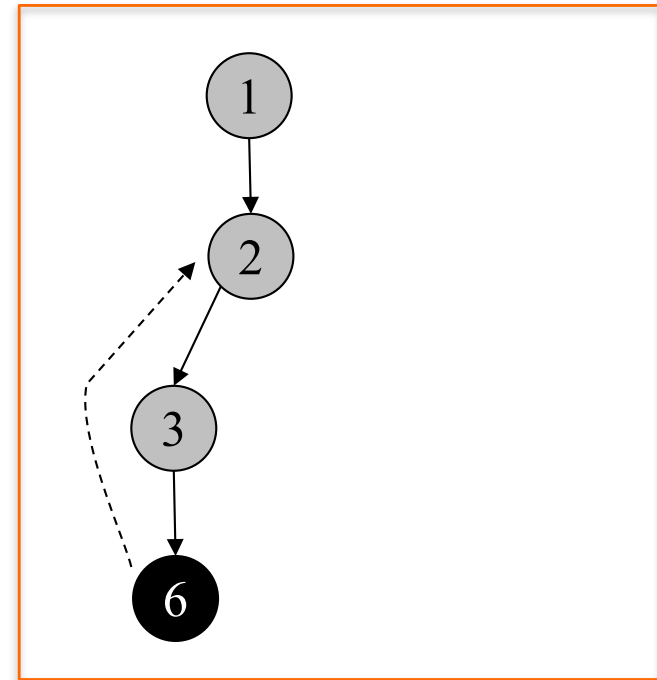
Ação: vértice 6 torna-se preto

Tempo de término: 5

Teste de Grafo Cíclico/Acíclico: Exemplo



grafo **cíclico**



árvore de busca
em profundidade

Teste de Grafo Cíclico/Acíclico: Complexidade

- Mesma da busca em profundidade
 - $O(|V|+|V|^2)$ para matrizes de adjacências
 - $O(|V|+|A|)$ para listas de adjacências

Ordenação Topológica:

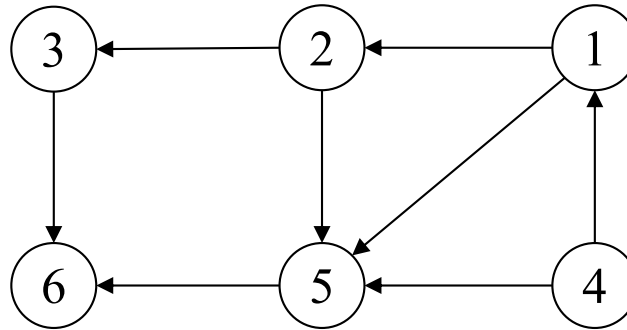
Definição

- Ordenação topológica de um DAG
 - Ordenação linear de todos os vértices de G , tal que se G contém uma aresta (u,v) então u aparece antes de v
 - ordenação dos vértices de G ao longo de uma linha horizontal de tal forma que todas as arestas estão direcionadas da esquerda para a direita
- Aplicada a DAGs
 - DAGs podem ser utilizados, por exemplo, para indicar precedência entre eventos

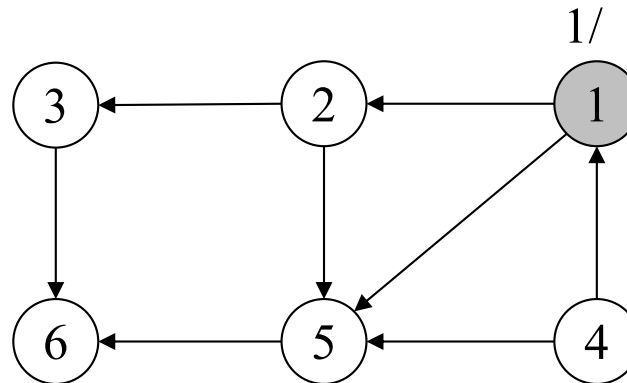
Ordenação Topológica: Busca em Profundidade

- Ordenação topológica
 - pode ser obtida por busca em profundidade
- Algoritmo
 1. Faça uma busca em profundidade
 2. Quando um vértice é pintado de preto, insira-o na cabeça de uma lista de vértices
 3. Retorne a lista de vértices

Ordenação Topológica: Exemplo



Ordenação Topológica: Exemplo



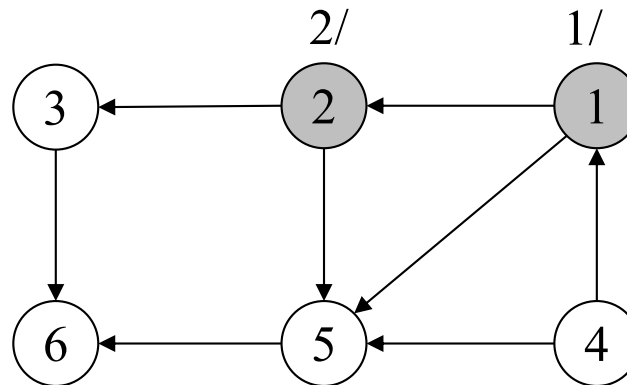
Vértice origem: 1

Tempo de descoberta: 1

Ação: vértice 1 torna-se cinza

Tempo de término: -

Ordenação Topológica: Exemplo



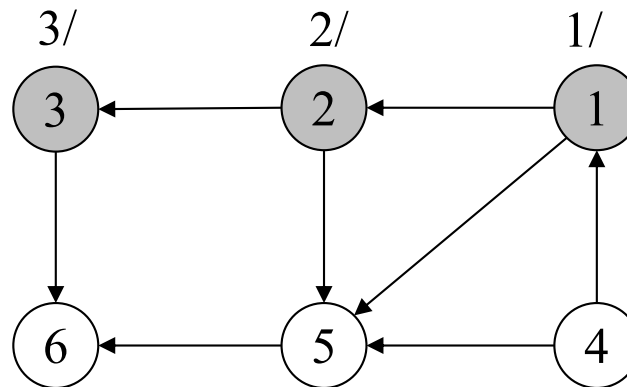
Primeiro vértice não descoberto adjacente a 1: 2

Tempo de descoberta: 2

Ação: vértice 2 torna-se cinza

Tempo de término:

Ordenação Topológica: Exemplo



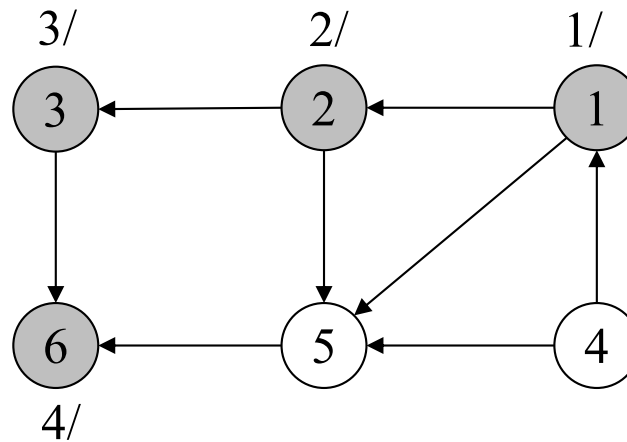
Primeiro vértice não descoberto adjacente a 2: 3

Tempo de descoberta: 3

Ação: vértice 3 torna-se cinza

Tempo de término:

Ordenação Topológica: Exemplo



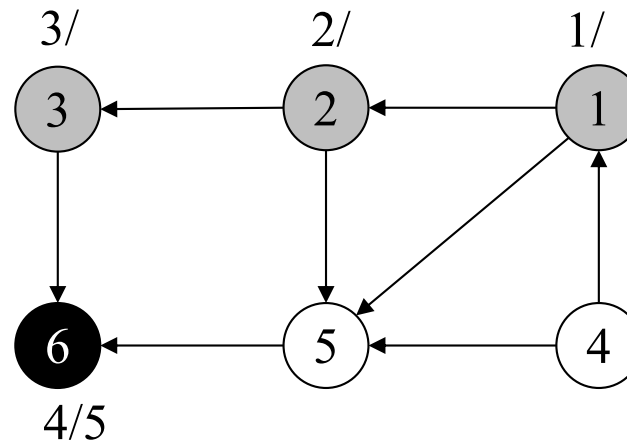
Primeiro vértice não descoberto adjacente a 3: 6

Tempo de descoberta: 4

Ação: vértice 6 torna-se cinza

Tempo de término:

Ordenação Topológica: Exemplo



lista de
vértices



Primeiro vértice não descoberto adjacente a 6: nenhum

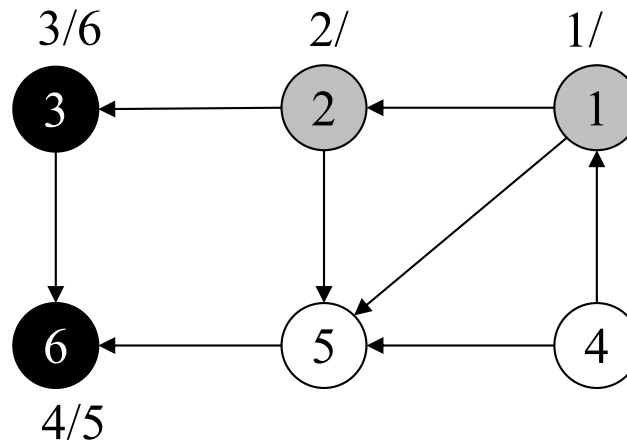
Tempo de descoberta: -

Ação: vértice 6 torna-se preto

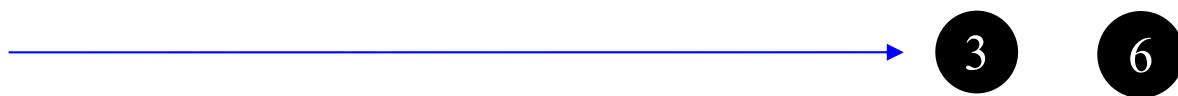
Tempo de término: 5

Ação: insere o vértice 6 na cabeça da lista de vértices

Ordenação Topológica: Exemplo



lista de
vértices



Primeiro vértice não descoberto adjacente a 3: nenhum

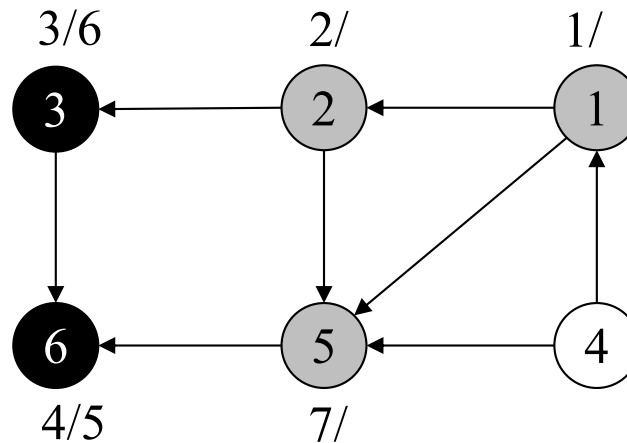
Tempo de descoberta: -

Ação: vértice 3 torna-se preto

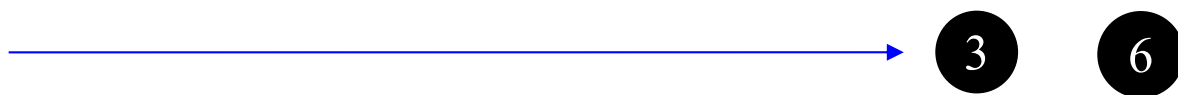
Tempo de término: 6

Ação: insere o vértice 3 na cabeça da lista de vértices

Ordenação Topológica: Exemplo



lista de
vértices



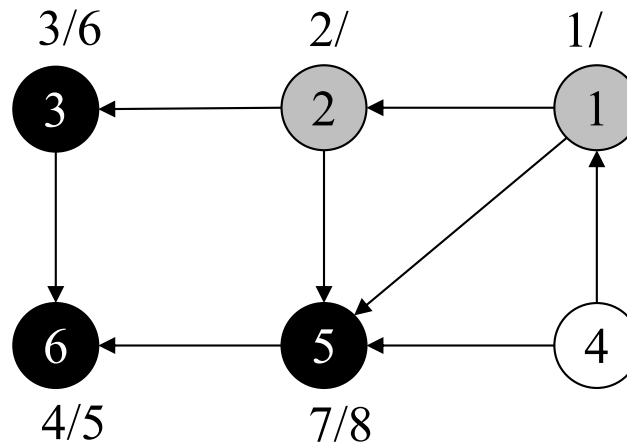
Primeiro vértice não descoberto adjacente a 2: 5

Tempo de descoberta: 7

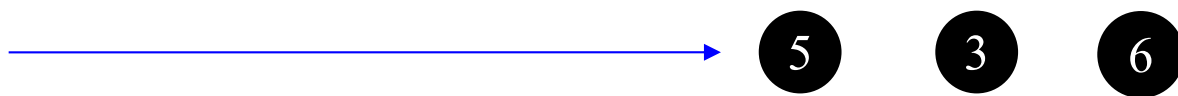
Ação: vértice 5 torna-se cinza

Tempo de término: -

Ordenação Topológica: Exemplo



lista de
vértices



Primeiro vértice não descoberto adjacente a 5: nenhum

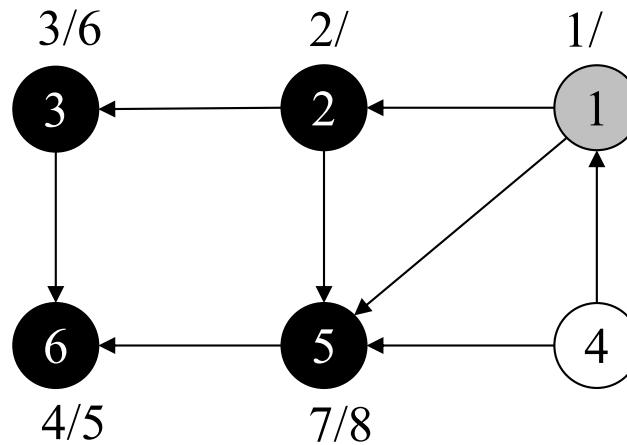
Tempo de descoberta:

Ação: vértice 5 torna-se preto

Tempo de término: 8

Ação: insere o vértice 5 na cabeça da lista de vértices

Ordenação Topológica: Exemplo



lista de
vértices



Primeiro vértice não descoberto adjacente a 2: nenhum

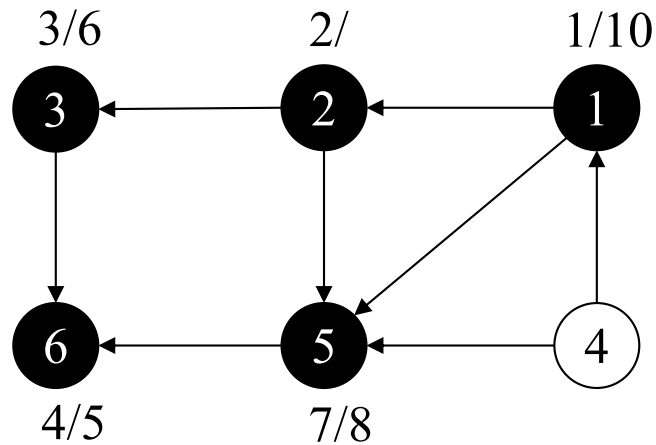
Tempo de descoberta:

Ação: vértice 2 torna-se preto

Tempo de término: 9

Ação: insere o vértice 2 na cabeça da lista de vértices

Ordenação Topológica: Exemplo



lista de
vértices



Primeiro vértice não descoberto adjacente a 1: nenhum

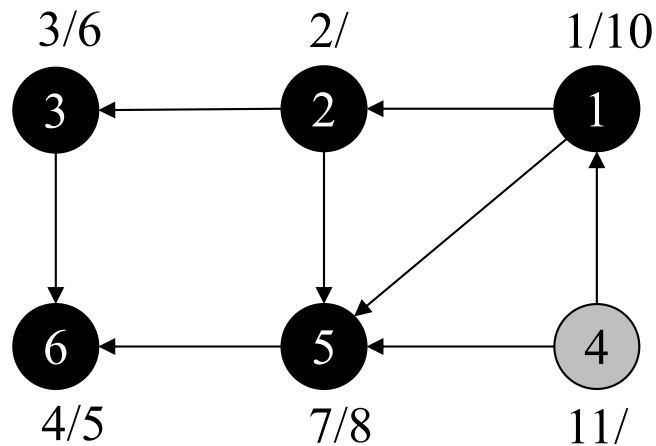
Tempo de descoberta:

Ação: vértice 1 torna-se preto

Tempo de término: 10

Ação: insere o vértice 1 na cabeça da lista de vértices

Ordenação Topológica: Exemplo



lista de
vértices



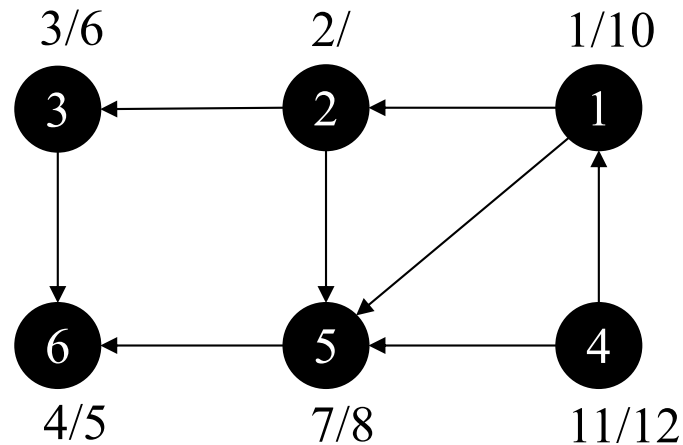
Vértice origem: 4

Tempo de descoberta: 11

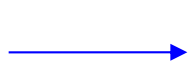
Ação: vértice 4 torna-se cinza

Tempo de término: -

Ordenação Topológica: Exemplo



lista de
vértices



Primeiro vértice não descoberto adjacente a 4: nenhum

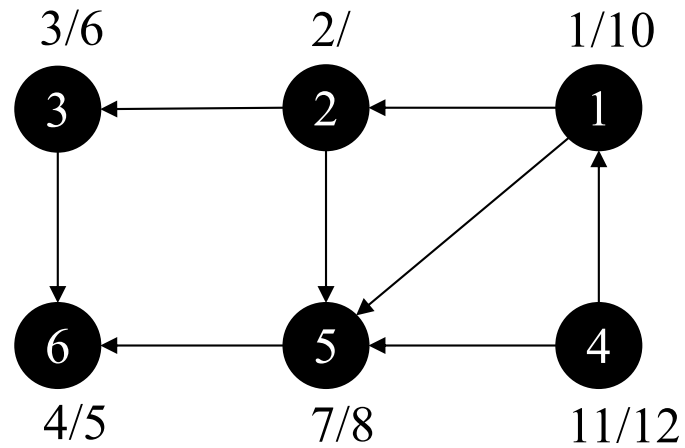
Tempo de descoberta: -

Ação: vértice 4 torna-se preto

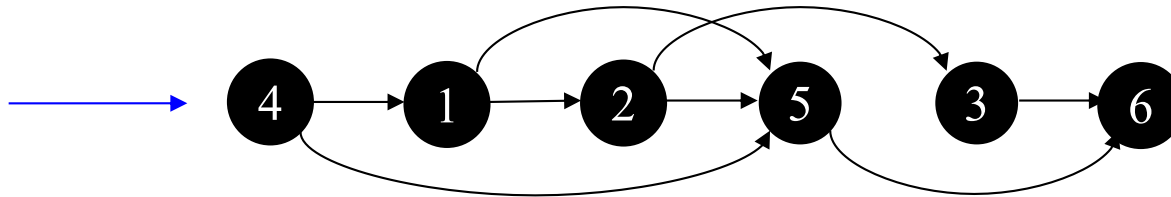
Tempo de término: 12

Ação: insere o vértice 4 na cabeça da lista de vértices

Ordenação Topológica: Exemplo



lista de
vértices



Ordenação Topológica: Considerações Finais

○ Atenção

- não há uma única ordenação topológica
- não há ordenação topológica em grafos com ciclos

○ Complexidade

- mesma da busca em profundidade
 - $O(|V|+|V|^2)$ para matrizes de adjacência
 - $O(|V|+|A|)$ para listas de adjacência
 - Inserir um elemento na cabeça da lista: **$O(1)$**

BIBLIOGRAFIA

- N. Ziviani. Projeto de Algoritmos, Thomson, 2a. Edição, 2004.
- T. H. Cormen, C. E. Leiserson and R. L. Rivest. Introduction to Algorithms, MIT Press, 2nd Edition, 2001.