Grafos: Busca

SCE-183 Algoritmos e Estruturas de Dados 2

Thiago A. S. Pardo
Maria Cristina

Percorrendo um grafo

Percorrendo um Grafo

- Percorrer um grafo é um problema fundamental
- Deve-se ter uma forma sistemática de visitar as arestas e os vértices
- O algoritmo deve ser suficientemente flexível para se adequar à diversidade de grafos

Eficiência

Percorrendo um Grafo

- Eficiência
  - Não deve haver repetições (desnecessárias) de visitas a um vértice e/ou aresta

Correção

Percorrendo um Grafo

- Correção
  - Todos os vértices e/ou arestas devem ser visitados

Solução

Percorrendo um Grafo

- Solução
  - Marcar os vértices
    - não visitados
    - visitados
    - processados

Solução

Percorrendo um Grafo

- Solução
  - Manter uma lista de vértices no estado visitados
  - Há duas possibilidades
    - Busca em largura (usando uma fila)
    - Busca em profundidade (usando uma pilha)

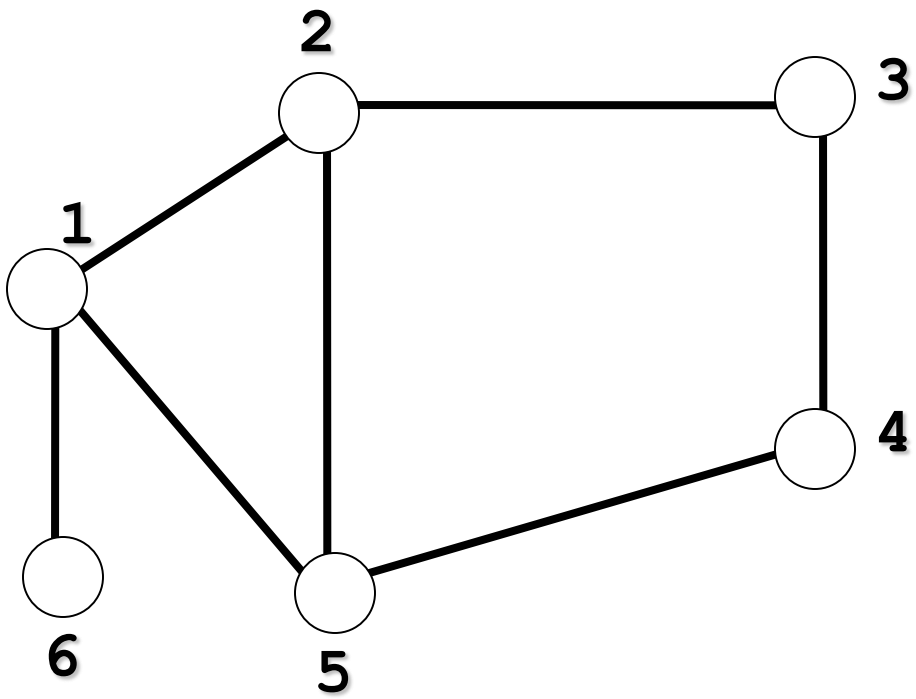
BFS (Busca em Largura)

Percorrendo um Grafo

- BFS – *Breadth-First Search*
  - Todos os nós com distância k a um nó v são visitados antes dos nós com distância k+1
  - Descobre todos os vértices alcançáveis a partir de v

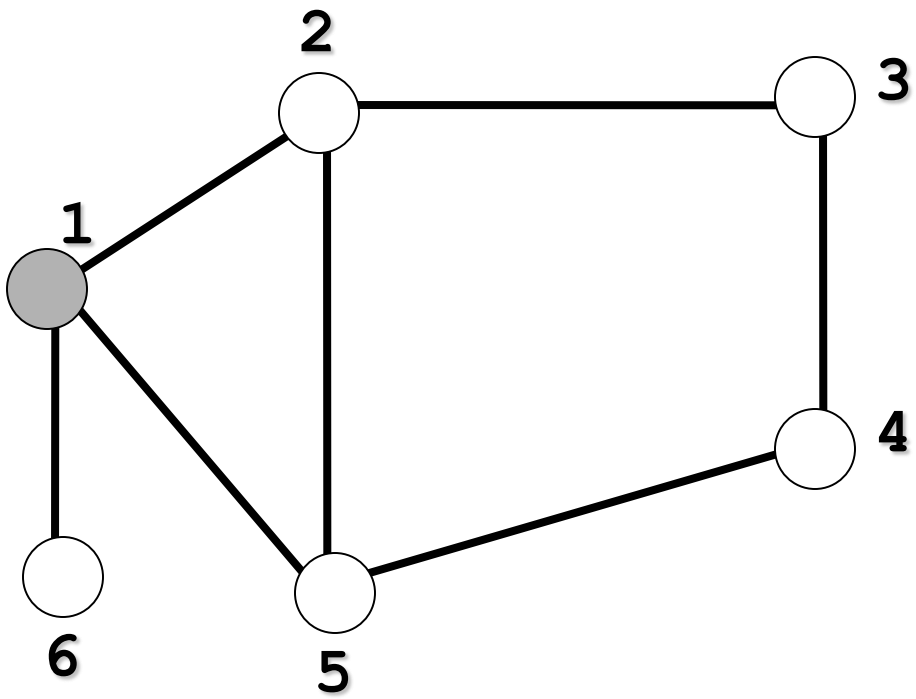
BFS – exemplo

Percorrendo um Grafo: BFS



BFS – exemplo

Percorrendo um Grafo: BFS

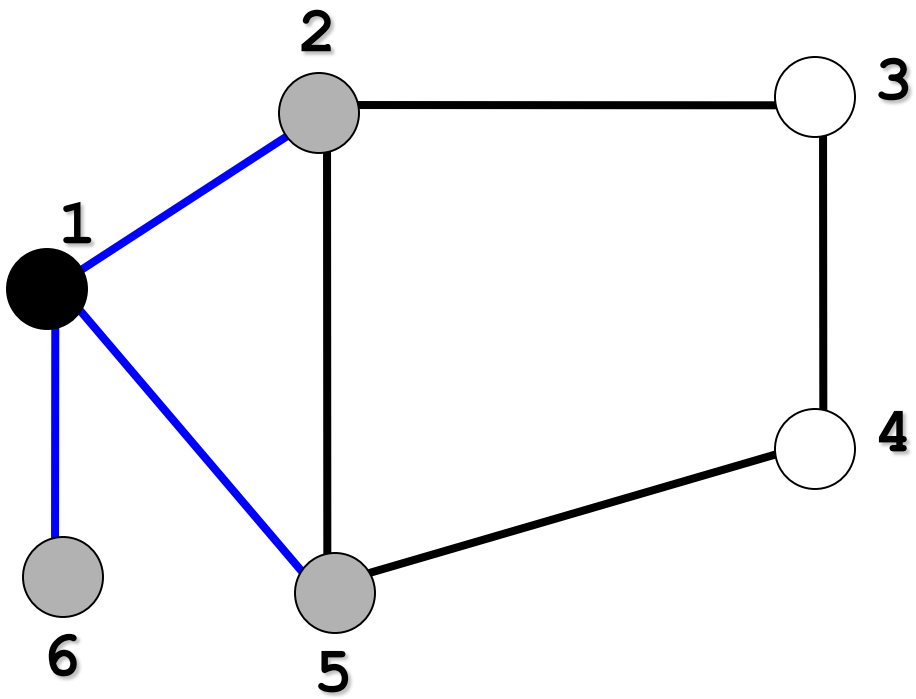


Nó inicial: 1

BFS – exemplo

Percorrendo um Grafo: BFS

K=1

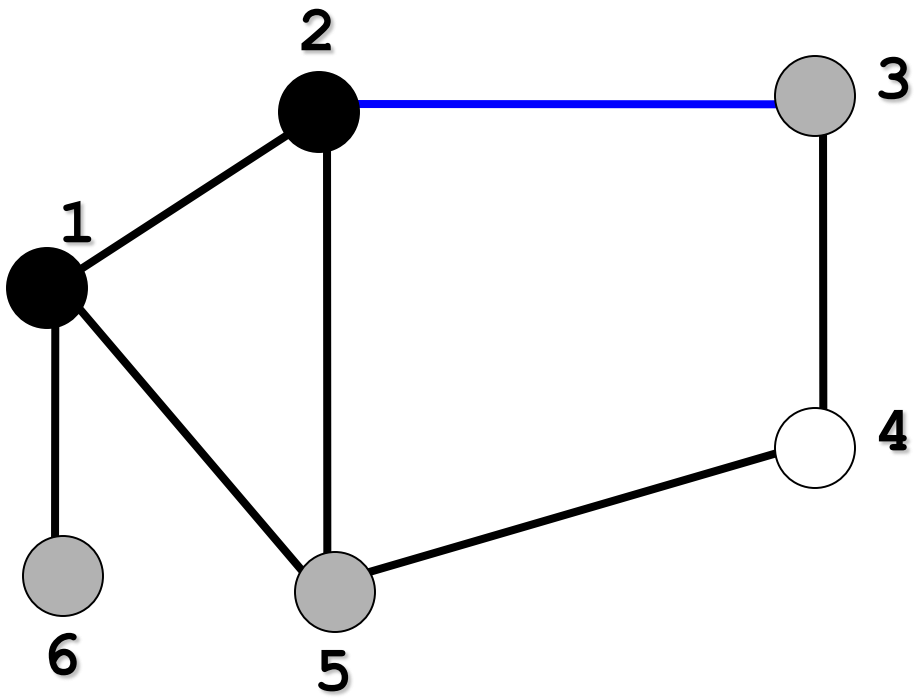


Visitam-se todos os nós não visitados adjacentes a 1: 2, 5 e 6

BFS – exemplo

Percorrendo um Grafo: BFS

K=2

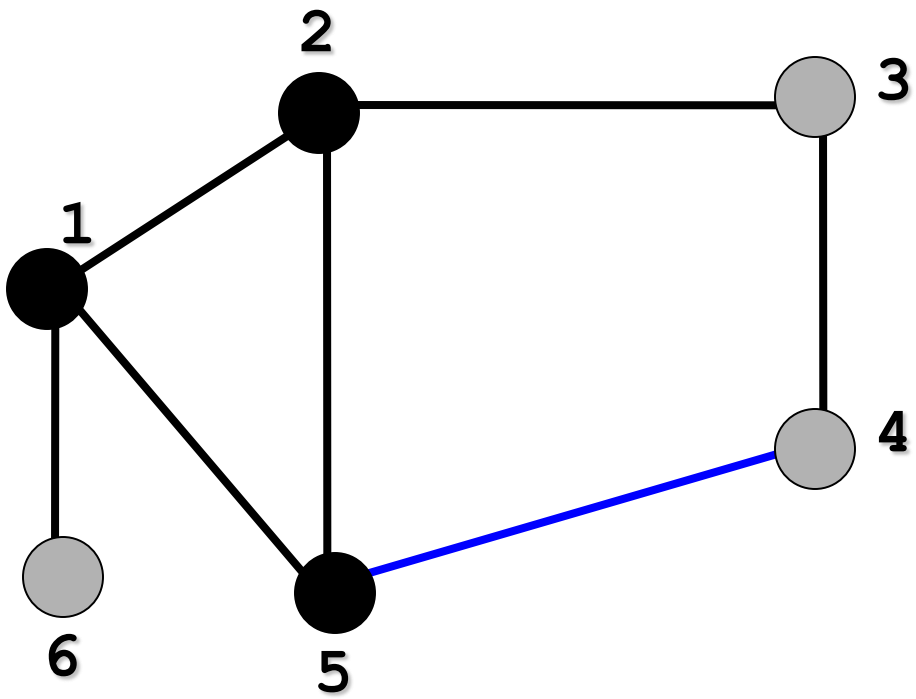


Visitam-se todos os nós não visitados adjacentes a 2: 3

BFS – exemplo

Percorrendo um Grafo: BFS

K=2

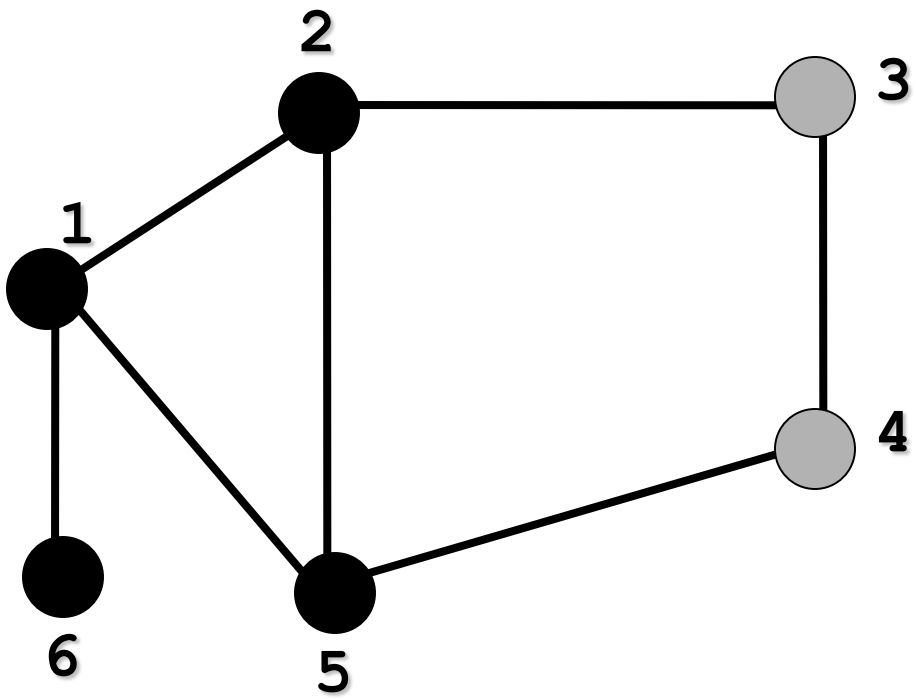


Visitam-se todos os nós não visitados adjacentes a 5: 4

BFS – exemplo

Percorrendo um Grafo: BFS

K=2

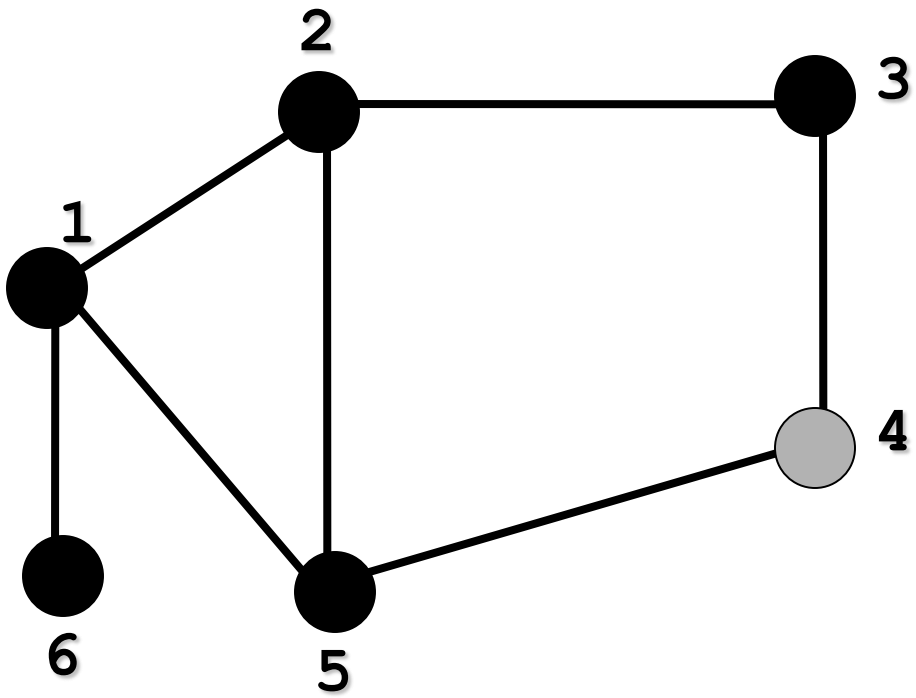


Visitam-se todos os nós não visitados adjacentes a 6: nenhum

BFS – exemplo

Percorrendo um Grafo: BFS

K=3

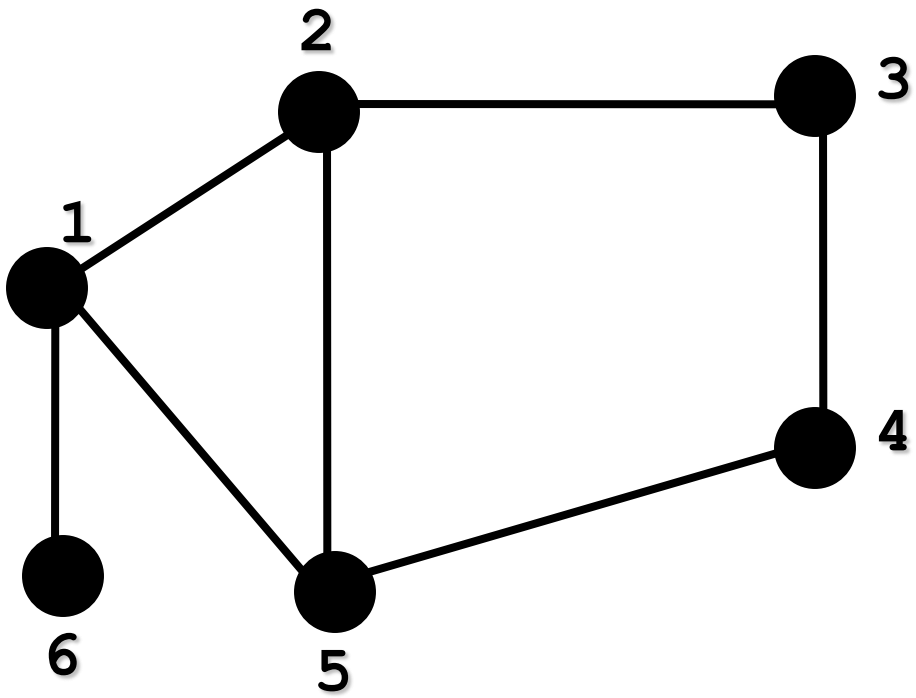


Visitam-se todos os nós não visitados adjacentes a 3: nenhum

BFS – exemplo

Percorrendo um Grafo: BFS

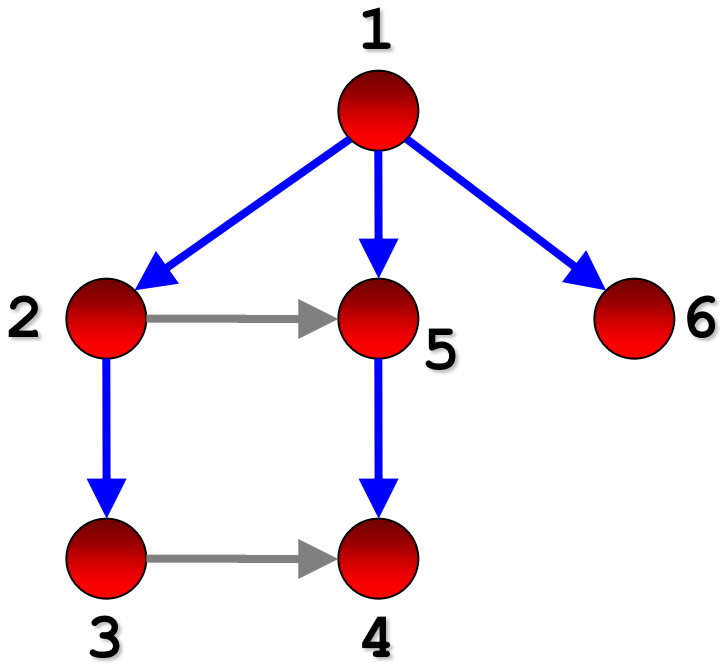
K=3



Visitam-se todos os nós não visitados adjacentes a 4: nenhum

BFS – exemplo

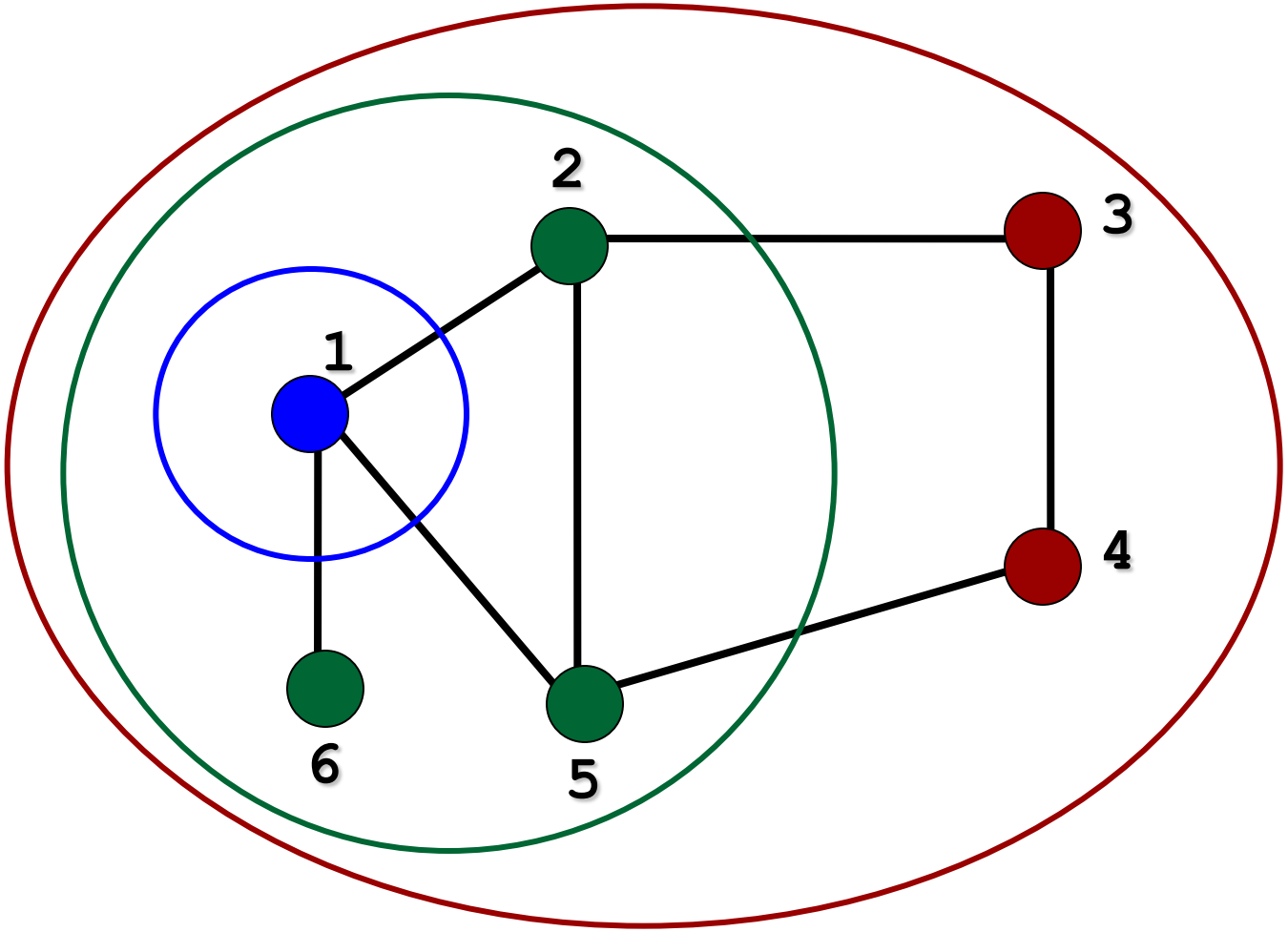
Percorrendo um Grafo: **árvore de busca em largura**



Atenção: arestas (2,5) e (3,4) não percorridas!

BFS – exemplo

Percorre-se o grafo como se houvesse uma onda na água!



BFS (Busca em Largura)

Percorrendo um Grafo

- BFS – *Breadth-First Search*
  - Em grafos não-dirigidos, cada aresta é visitada somente duas vezes
  - Em grafos dirigidos, cada aresta é visitada uma única vez

BFS (Busca em Largura)

Percorrendo um Grafo

- BFS – *Breadth-First Search*
  - Em geral, armazena-se a distância de cada nó em relação ao nó inicial onde a busca se iniciou
    - Útil em aplicações de caminho mais curto, por exemplo
  - É comum a utilização de esquemas de cores para identificar os nós ainda não visitados (branco), visitados (cinza) e já completamente processados (preto)

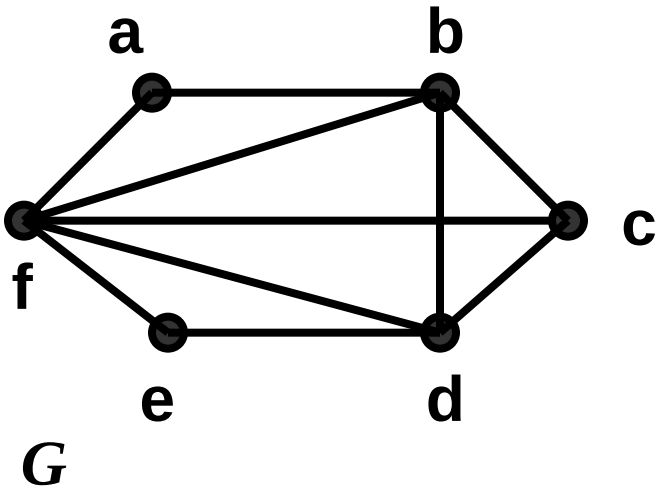
BFS (Busca em Largura)

Percorrendo um Grafo

- BFS – *Breadth-First Search*
  - Todos os vértices são inicializados brancos
  - Quando um vértice v é descoberto pela primeira vez, ele se torna cinza
  - Quando todos os vértices adjacentes a v são descobertos, v se torna preto

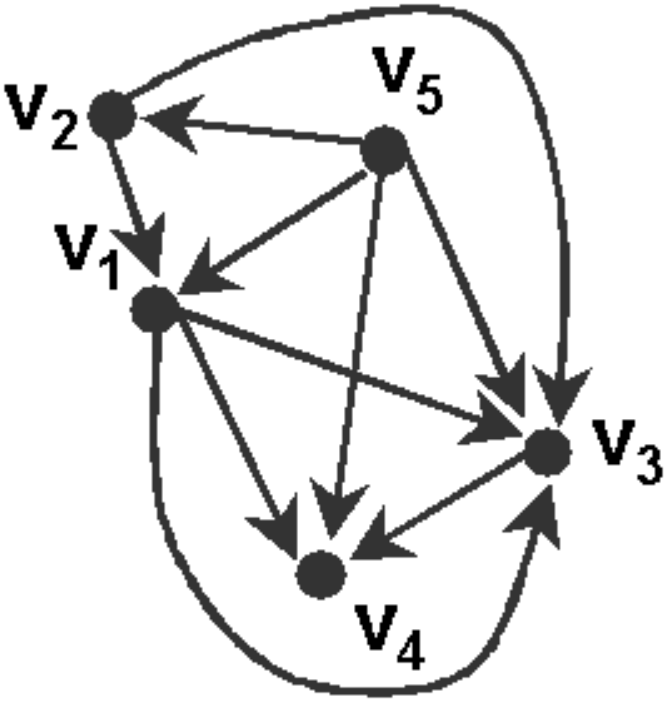
BFS (Busca em Largura)

- Exercício: faça a busca em largura no grafo abaixo, mostrando a ordem de visita aos vértices



BFS (Busca em Largura)

- Exercício: faça a busca em largura no grafo abaixo, mostrando a ordem de visita aos vértices



BFS (Busca em Largura)

- Algoritmo
  - Uso de uma fila para organizar que vértices devem ser visitados
    1. A fila começa com o vértice inicial
    2. O primeiro vértice da fila é recuperado e processado: seus vértices adjacentes são inseridos no fim da fila
    3. Se fila vazia, fim do processo; senão, volta-se ao passo 2

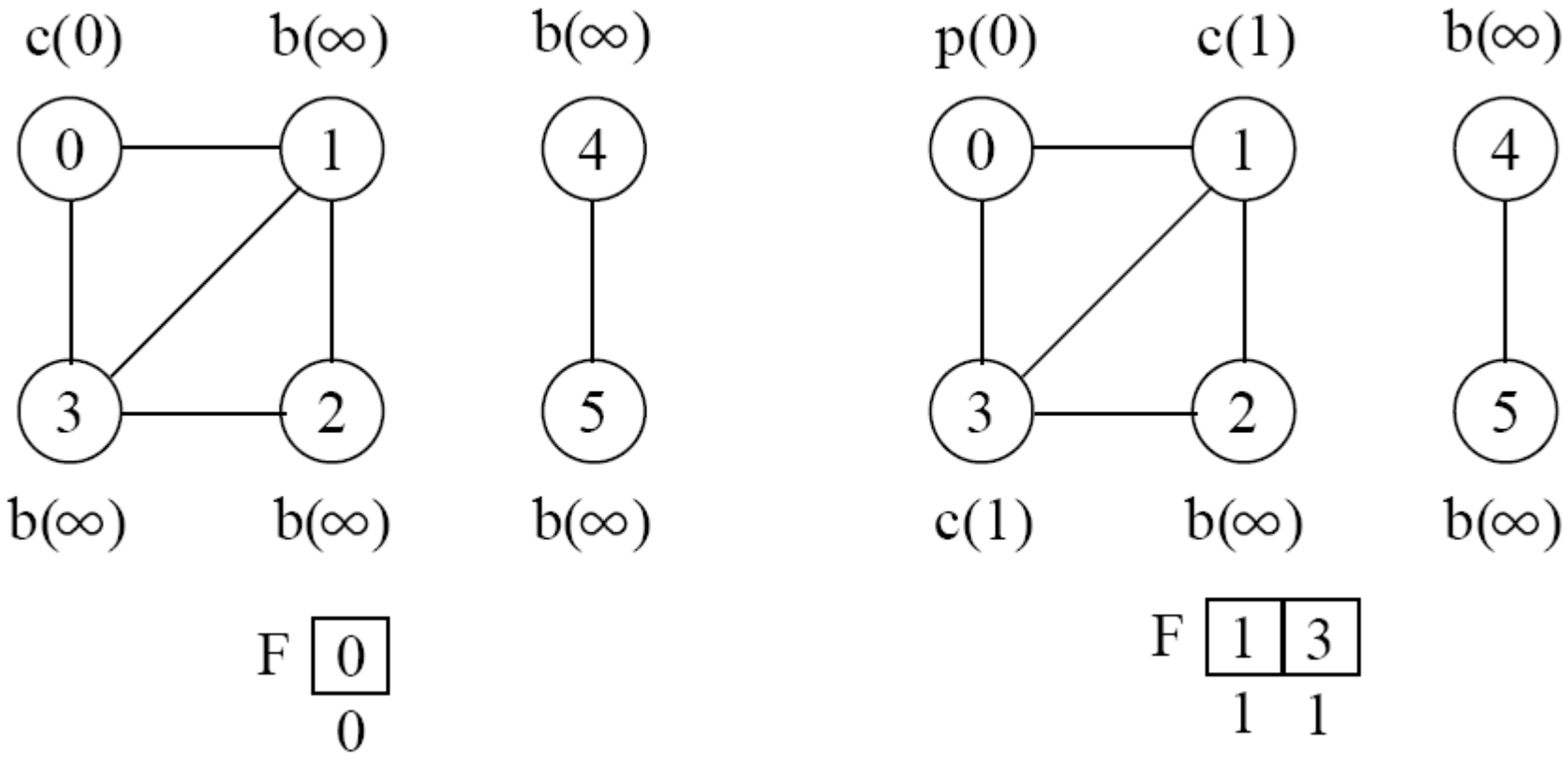
BFS (Busca em Largura)

- Algoritmo

  - Ou seja: a fila garante que os vértices com distância k a um vértice v sejam processados primeiro do que os vértices de distância k+1

BFS (Busca em Largura)

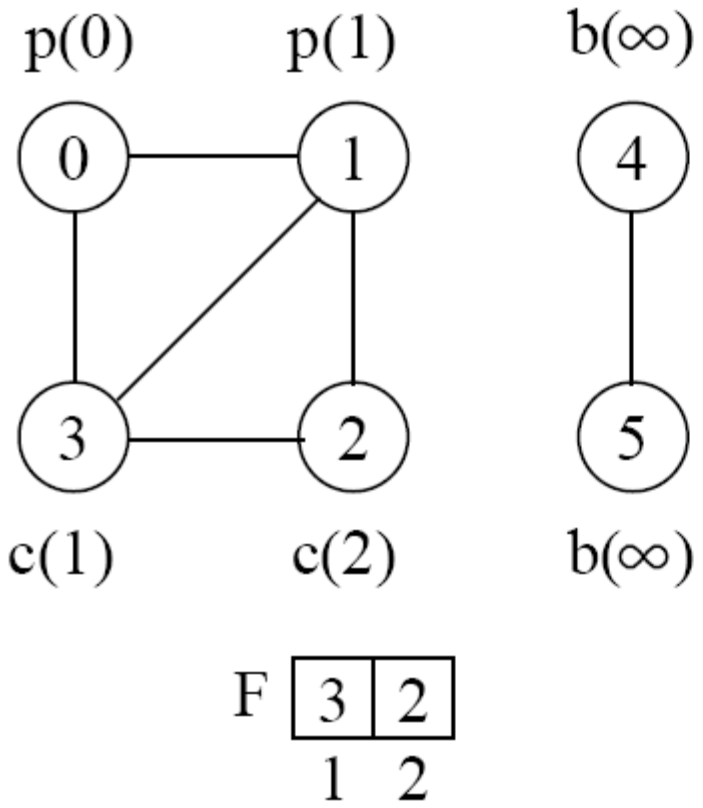
- Exemplo detalhado

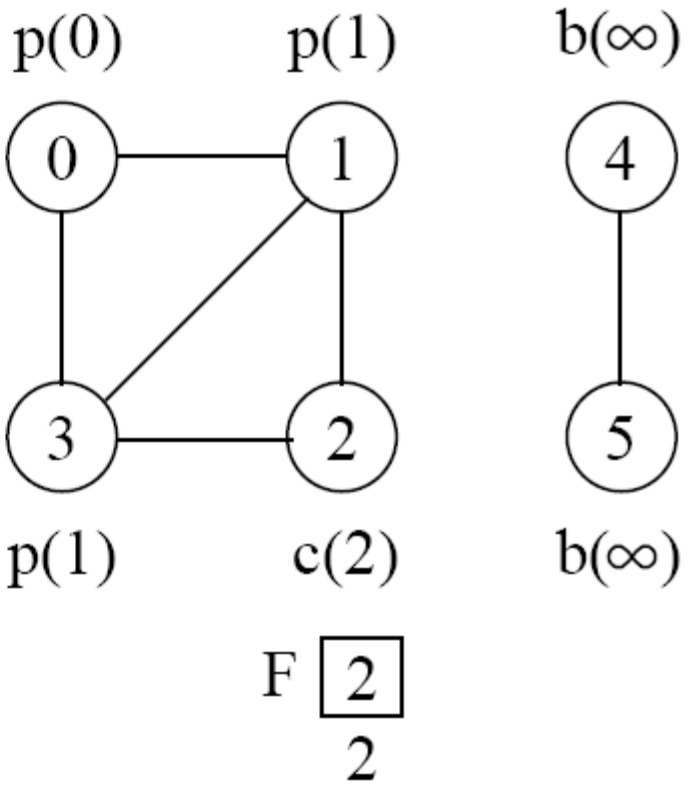


F=fila b=branco, c=cinza, p=preto, distância do vértice inicial entre parênteses

BFS (Busca em Largura)

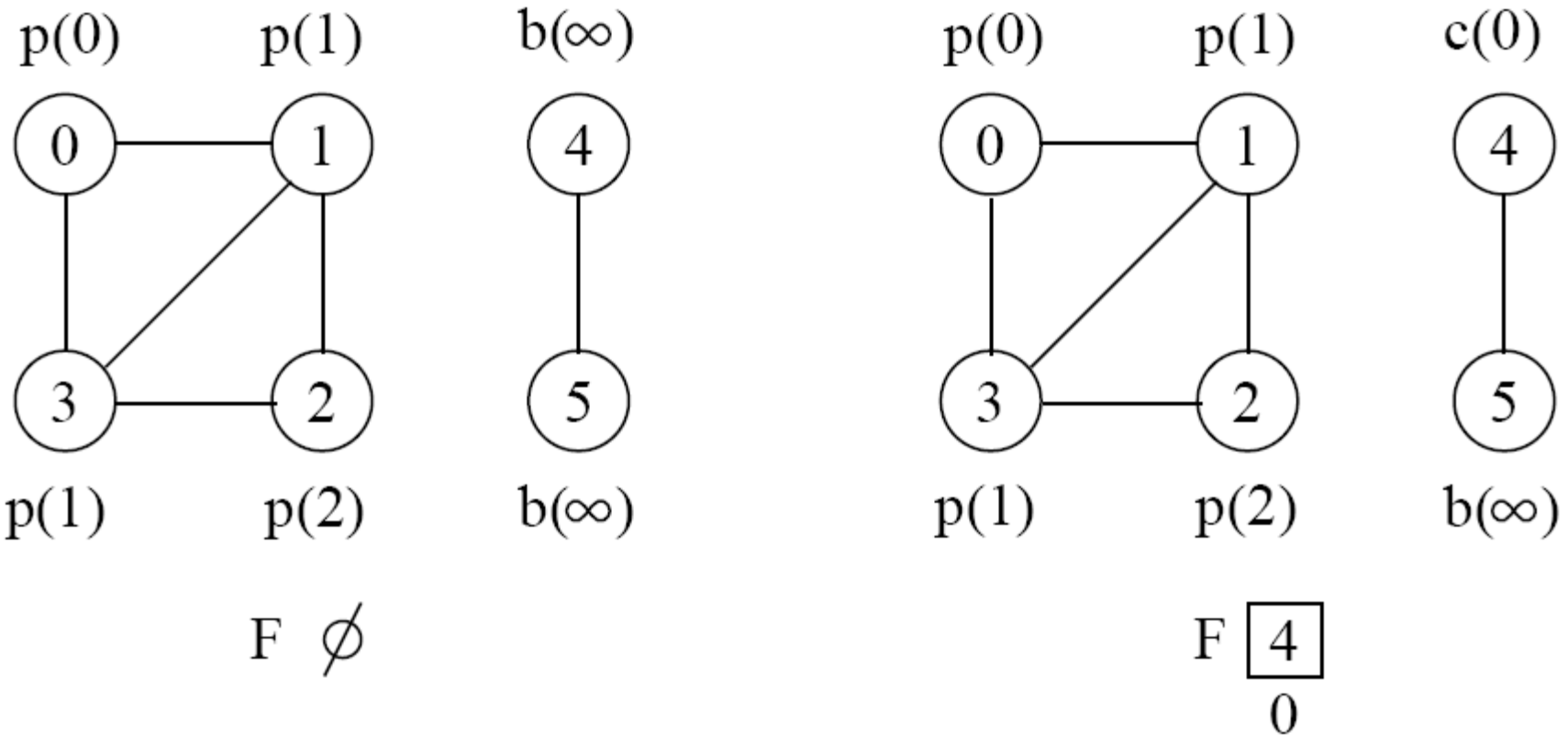
- Exemplo detalhado





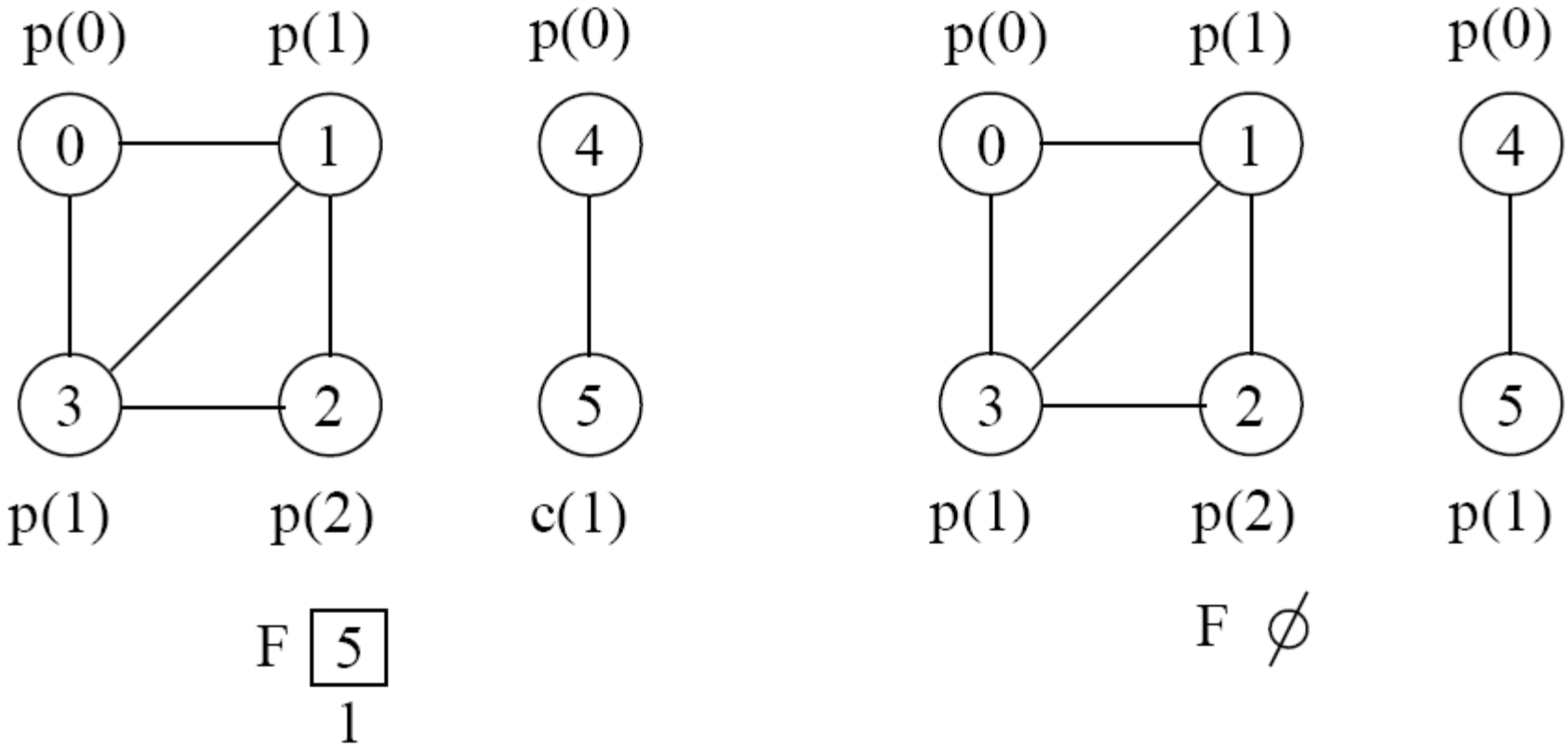
BFS (Busca em Largura)

- Exemplo detalhado



BFS (Busca em Largura)

- Exemplo detalhado



BFS (Busca em Largura)

- Implementação da busca em largura

Complexidade do BFS

O(|V| + |E|), ou seja, linear em relação ao tamanho da representação do grafo por listas de adjacências

- Todos os vértices são enfileirados/desenfileirados no máximo uma vez; o custo de cada uma dessas operações é O(1), e elas são executadas O(|V|) vezes
- A lista de adjacências de cada vértice é percorrida no máximo uma vez (quando o vértice é desenfileirado); o tempo total é O(|E|) (soma dos comprimentos de todas as listas, igual ao número de arestas)

BFS (Busca em Largura)

- A busca em largura resulta no caminho mais curto entre o vértice inicial e um vértice qualquer x
  - O procedimento VisitaBfs constrói uma árvore de busca em largura que pode ser recuperada pela variável Antecessor

BFS (Busca em Largura)

- Exercício para entregar na próxima aula
  - À mão (e individualmente), implemente em C uma sub-rotina que imprima os vértices do caminho mais curto entre o vértice inicial e outro vértice qualquer do grafo

DFS – Busca em Profundidade

Percorrendo um Grafo

DFS - *Depth-First Search*

- Explora-se profundamente cada cada vértice do grafo
  - Sempre se procura avançar na busca, sem olhar para os nós vizinhos de mesmo nível
  - Quando necessário, faz-se backtracking e volta-se aos vizinhos abandonados antes

DFS – Busca em Profundidade

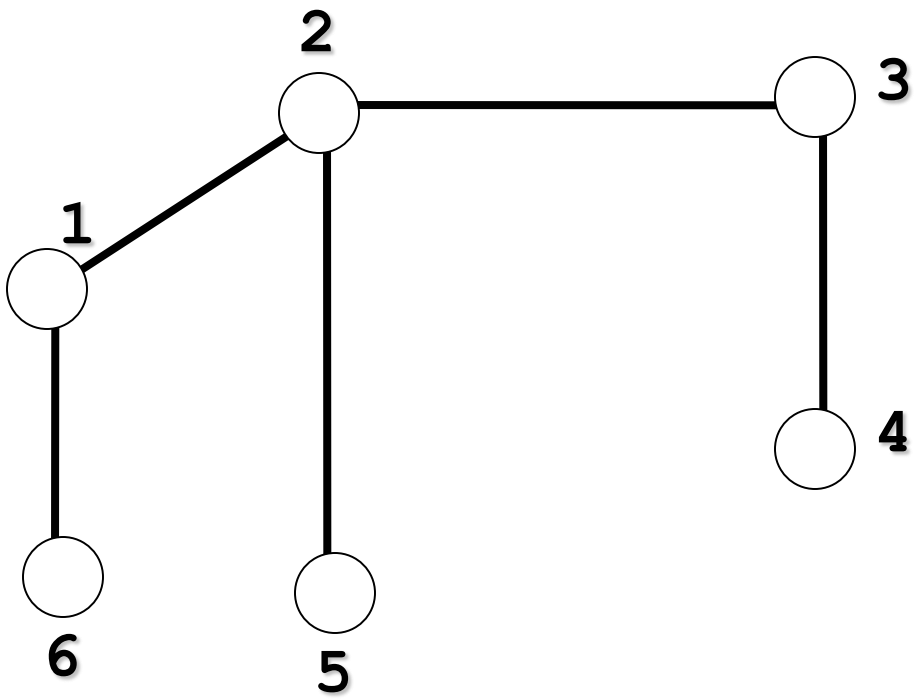
Percorrendo um Grafo

DFS - *Depth-First Search*

- Também se pode usar o esquema de cores para guiar a busca
  - Todos os vértices são inicializados brancos
  - Quando um vértice v é descoberto pela primeira vez, ele se torna cinza
  - Quando todos os vértices adjacentes a v são descobertos, v se torna preto

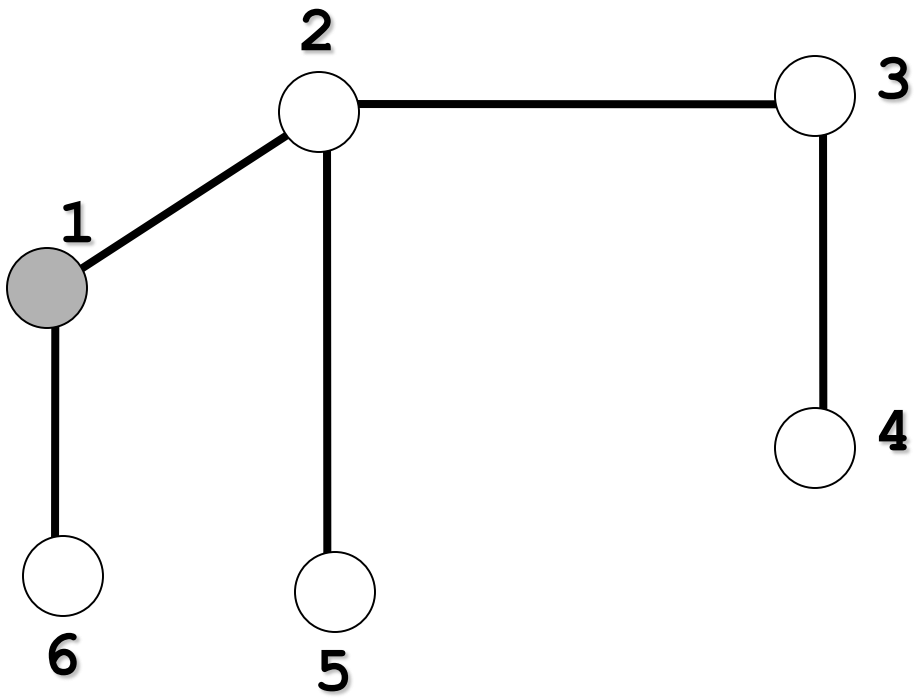
DFS – exemplo

Percorrendo um Grafo: DFS



DFS – exemplo

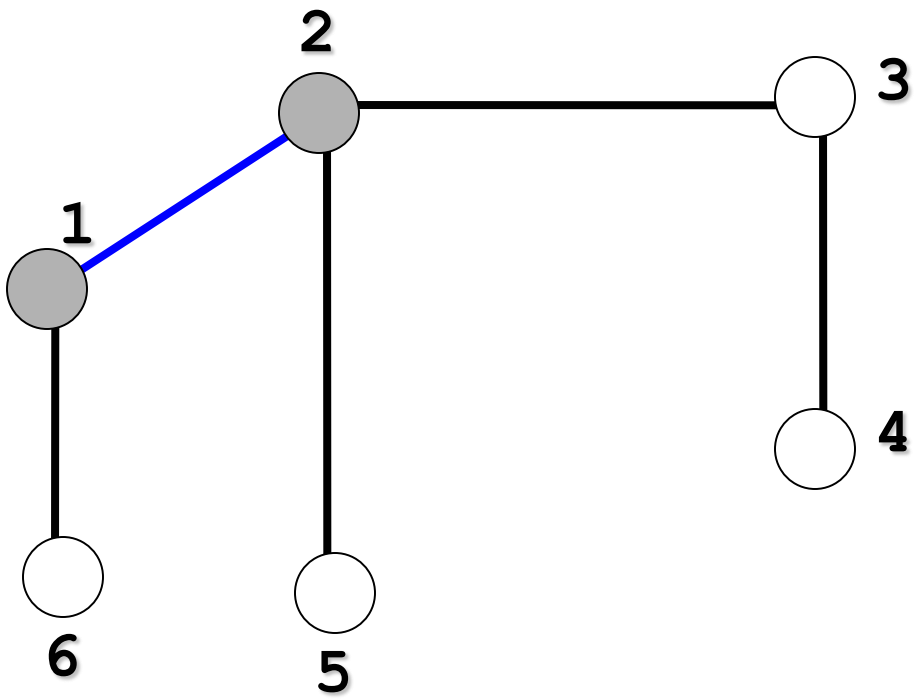
Percorrendo um Grafo: DFS



Nó inicial: 1

DFS – exemplo

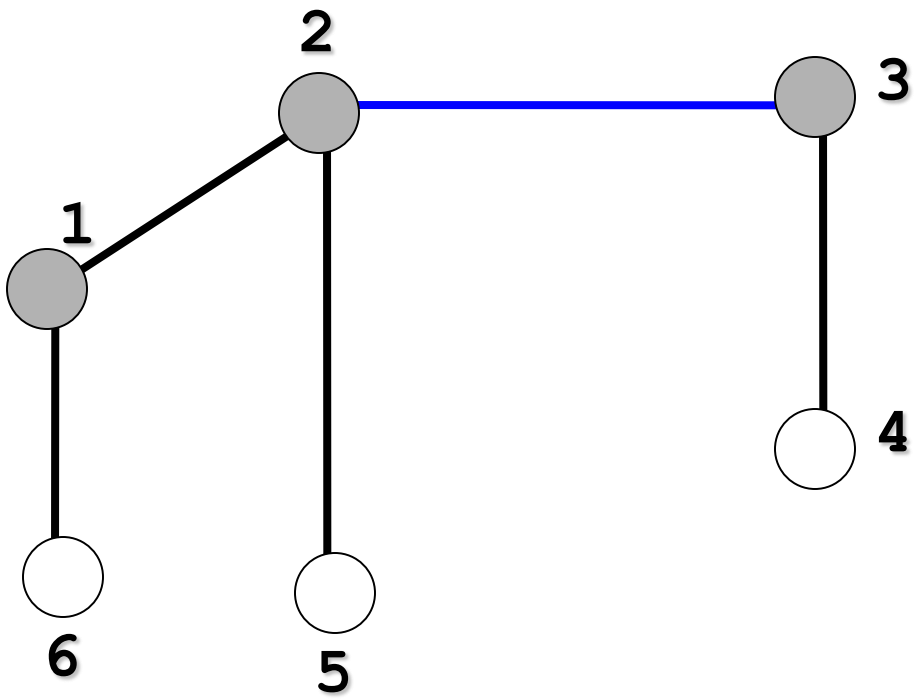
Percorrendo um Grafo: DFS



Busca-se pelo primeiro nó adjacente a 1: 2

DFS – exemplo

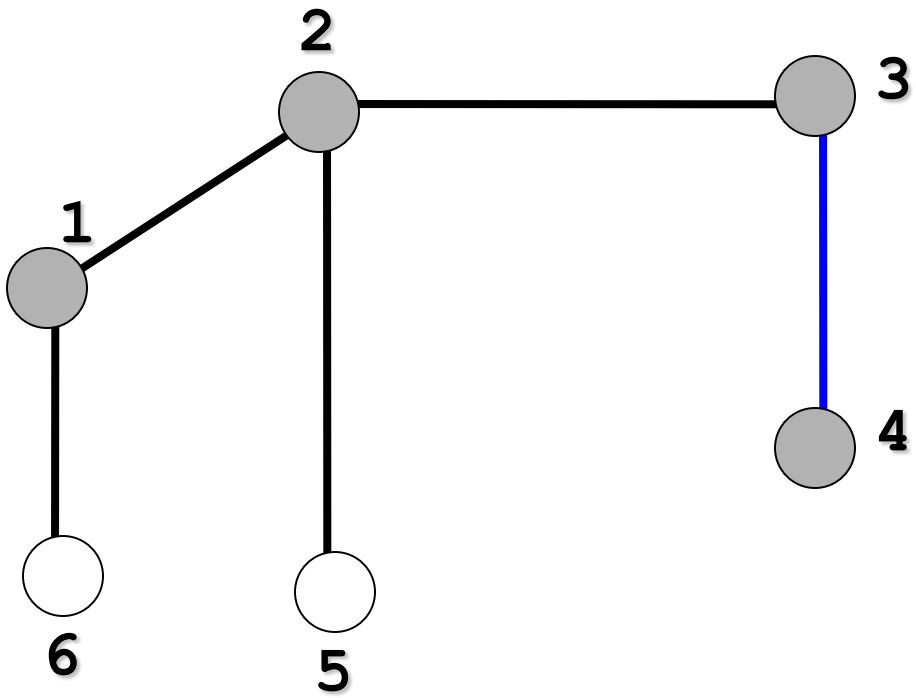
Percorrendo um Grafo: DFS



Busca-se pelo primeiro nó adjacente a 2: 3

DFS – exemplo

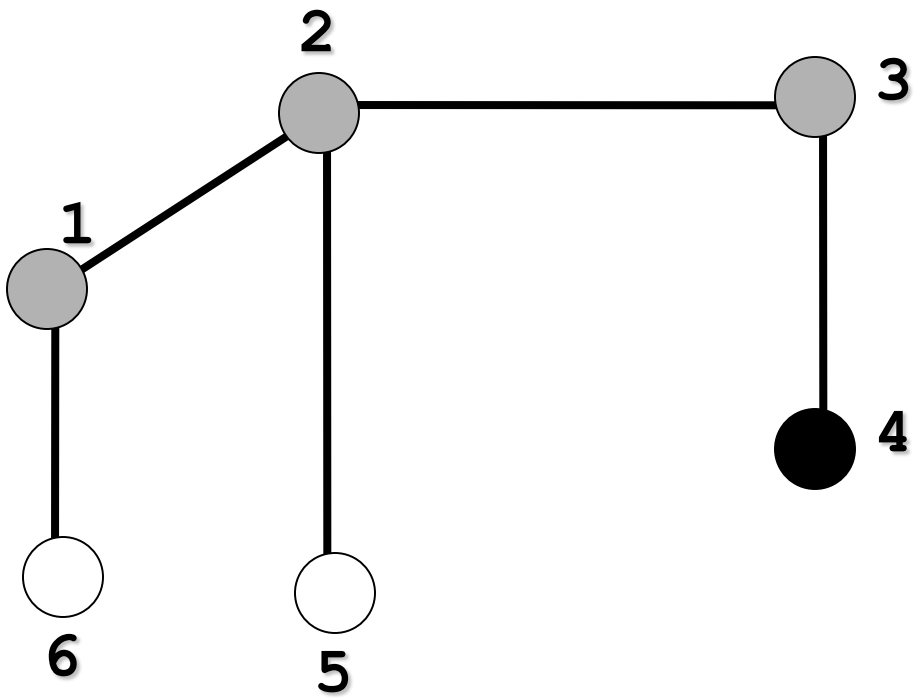
Percorrendo um Grafo: DFS



Busca-se pelo primeiro nó adjacente a 3: 4

DFS – exemplo

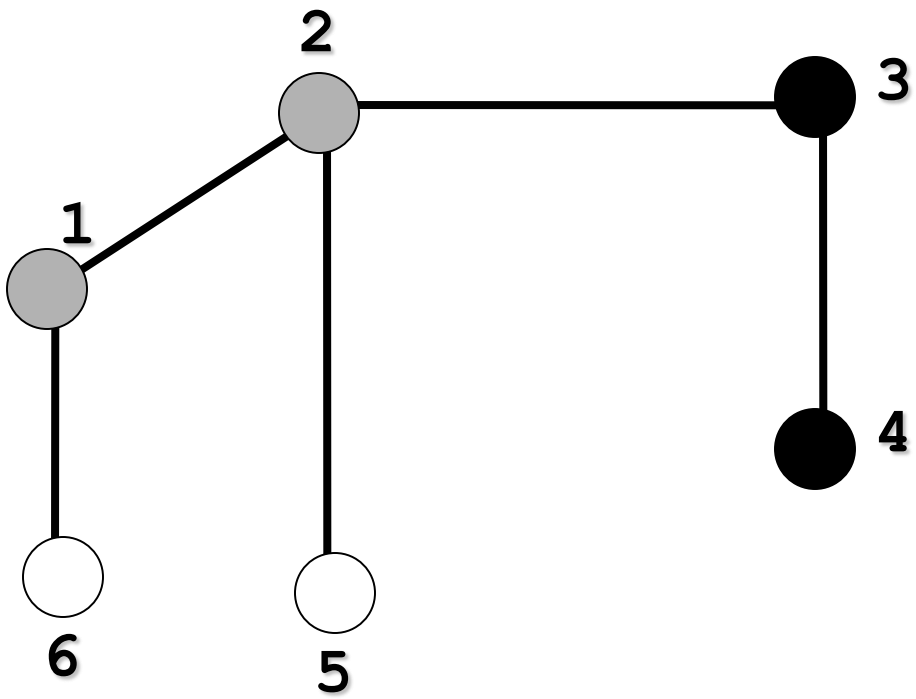
Percorrendo um Grafo: DFS



Nó 4 não tem mais nós adjacentes! Retorna-se ao anterior.

DFS – exemplo

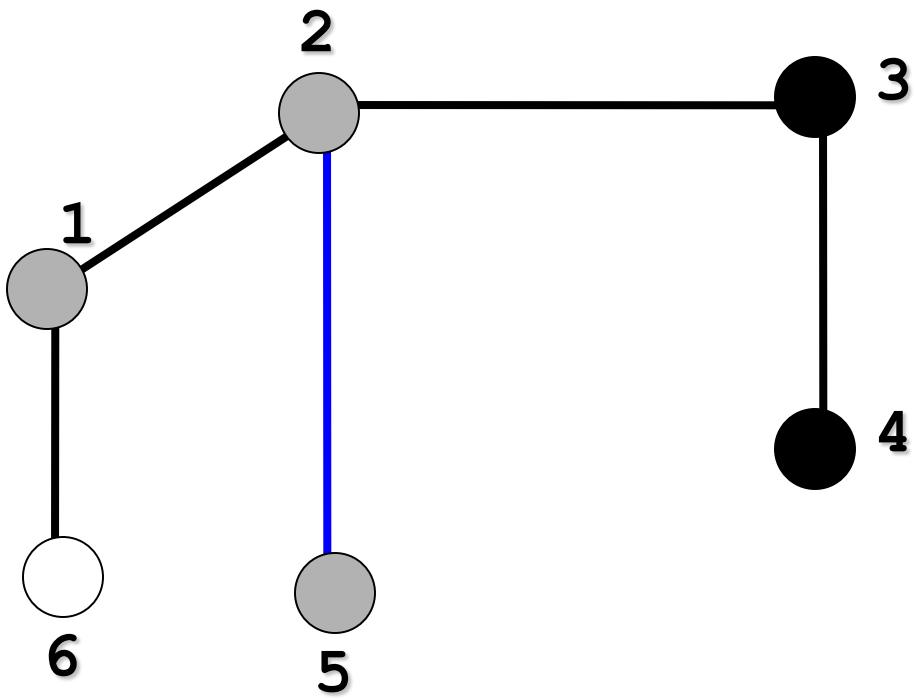
Percorrendo um Grafo: DFS



Nó 3 não tem mais nós adjacentes! Retorna-se ao anterior: 2

DFS – exemplo

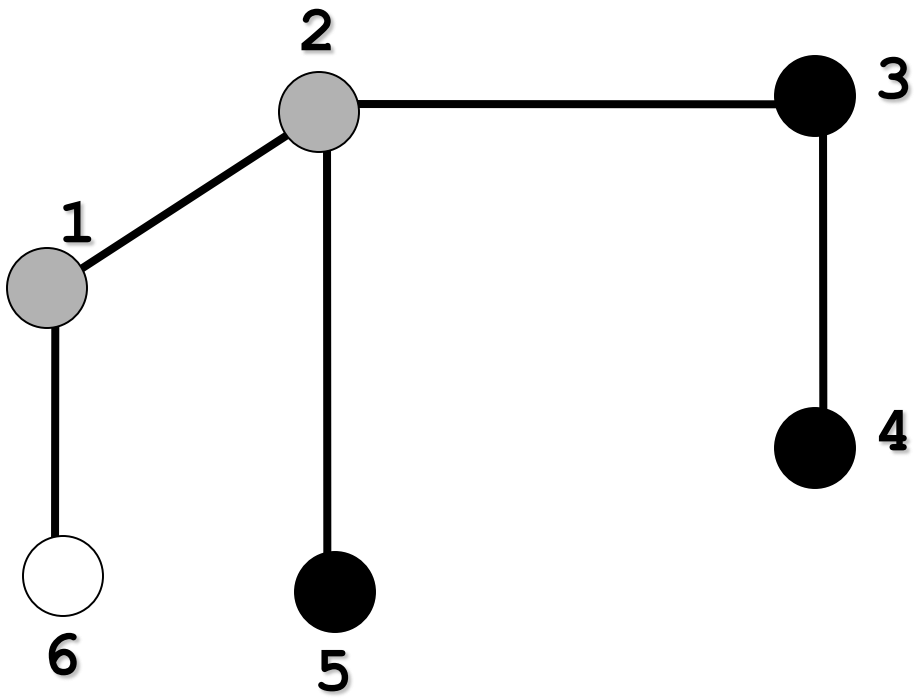
Percorrendo um Grafo: DFS



Busca-se pelo outro nó adjacente a 2: 5

DFS – exemplo

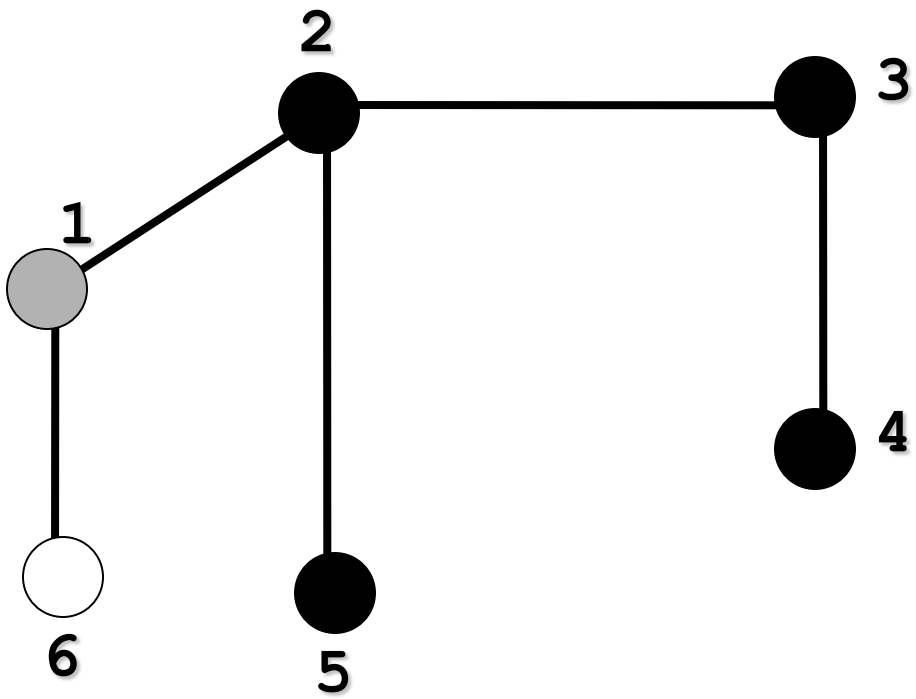
Percorrendo um Grafo: DFS



Nó 5 não tem mais nós adjacentes! Retorna-se ao anterior: 2

DFS – exemplo

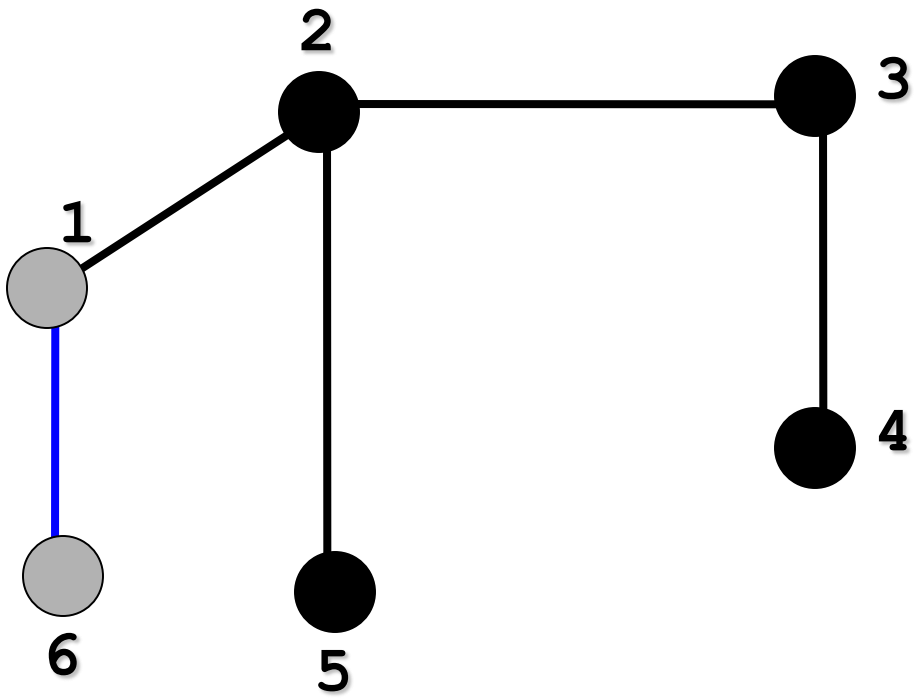
Percorrendo um Grafo: DFS



Nó 2 não tem mais nós adjacentes! Retorna-se ao anterior: 1

DFS – exemplo

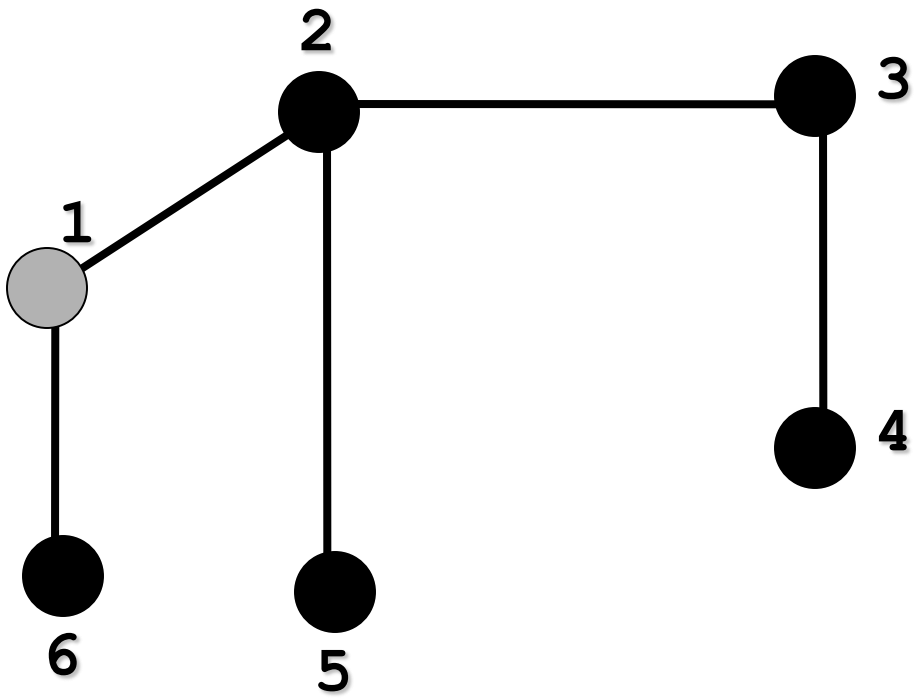
Percorrendo um Grafo: DFS



Busca-se pelo outro nó adjacente a 1: 6

DFS – exemplo

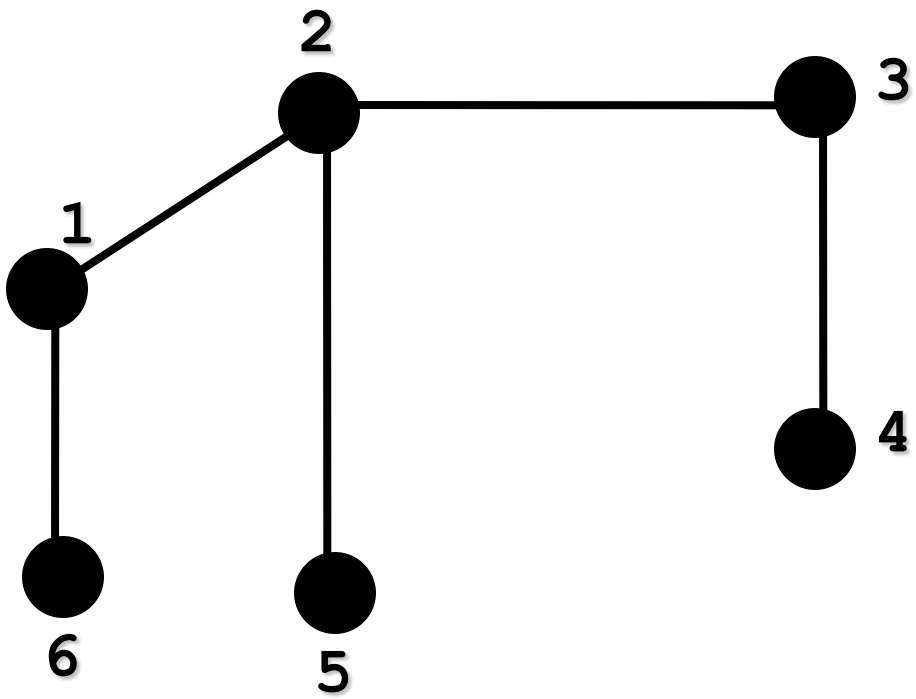
Percorrendo um Grafo: DFS



Nó 6 não tem mais nós adjacentes! Retorna-se ao anterior: 1

DFS – exemplo

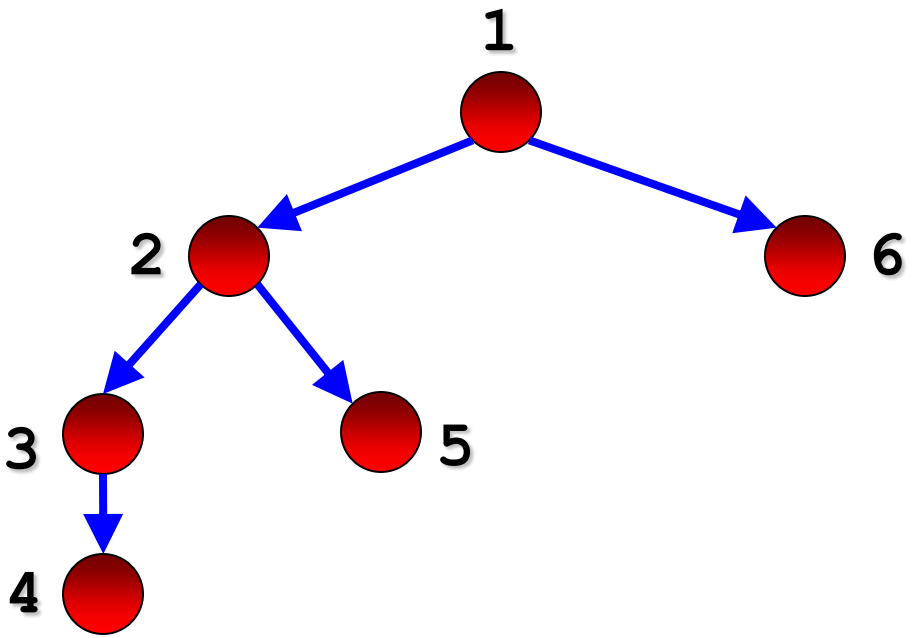
Percorrendo um Grafo: DFS



Nó 1 não tem mais nós adjacentes! Fim da busca

Árvore de busca em profundidade

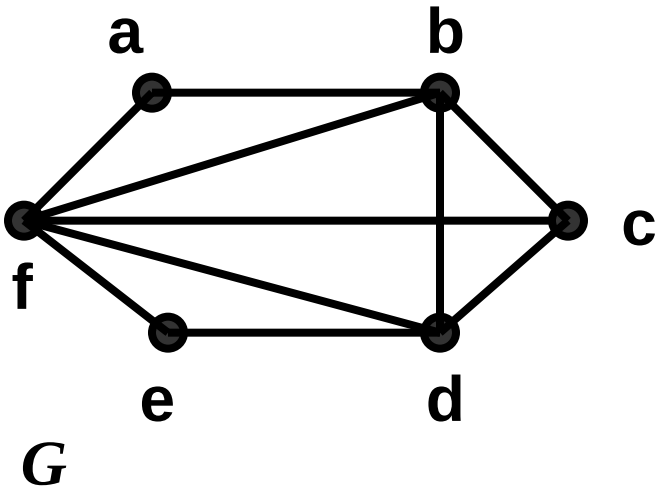
Percorrendo um Grafo: **Árvore de Busca em Profundidade**



Todas as arestas foram percorridas: mero acaso!

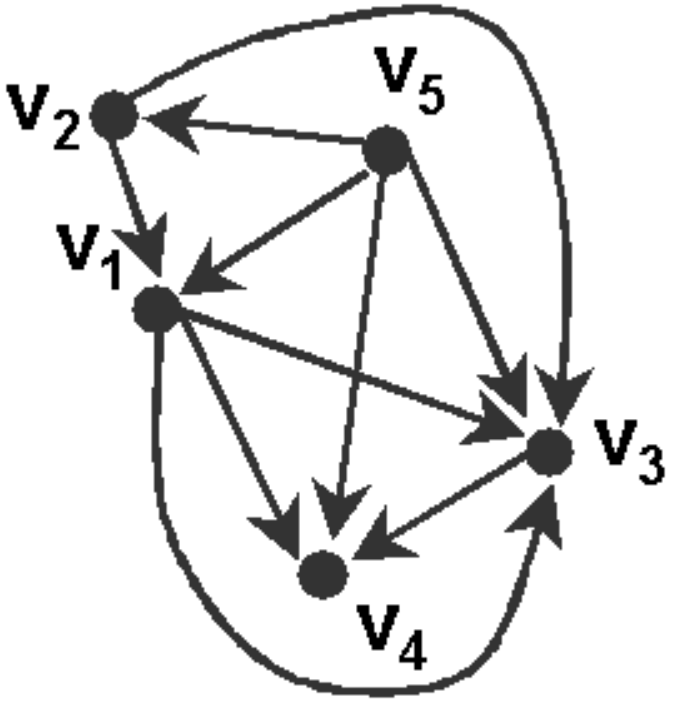
DFS

- Exercício: faça a busca em profundidade no grafo abaixo, mostrando a ordem de visita aos vértices



DFS

- Exercício: faça a busca em profundidade no grafo abaixo, mostrando a ordem de visita aos vértices

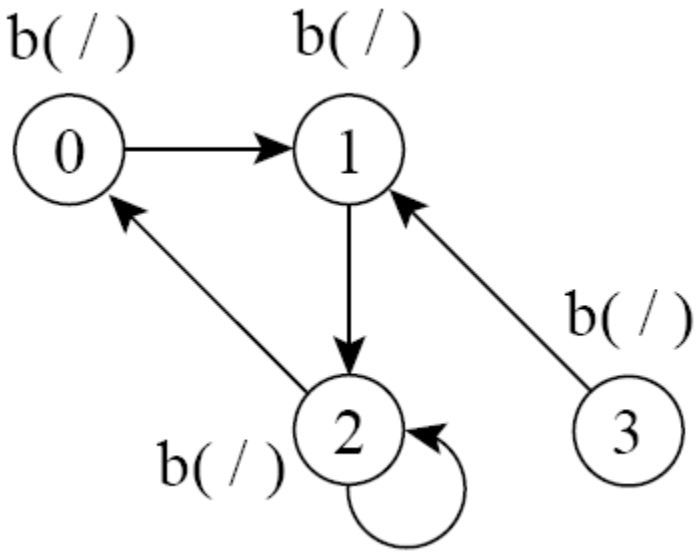


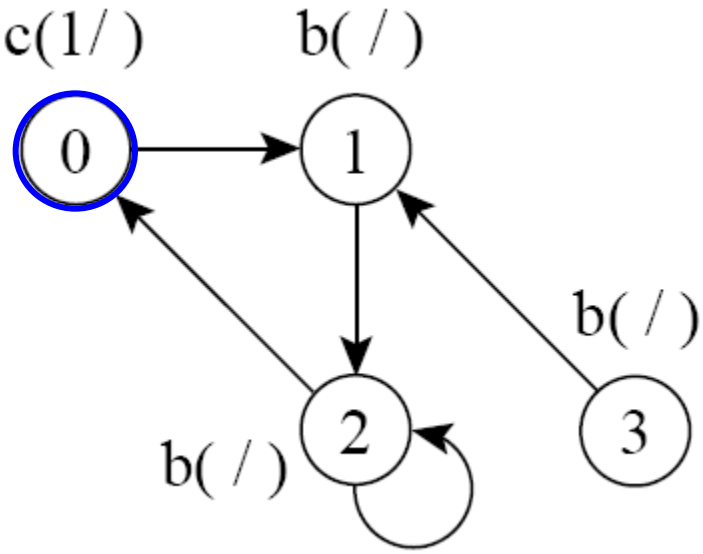
DFS

- Algoritmo
  - Uso de uma pilha para organizar que vértices devem ser visitados
    1. A cada escolha de caminho a ser desenvolvido, empilha-se o vértice original e segue-se o caminho
    2. Cada vez que o caminho acaba, retorna-se ao vértice anterior empilhado
       - Pilha pode ser implícita (via recursão) ou explícita
  - Costuma-se registrar o tempo de descoberta e o tempo de término da busca de cada vértice

DFS

- Exemplo detalhado

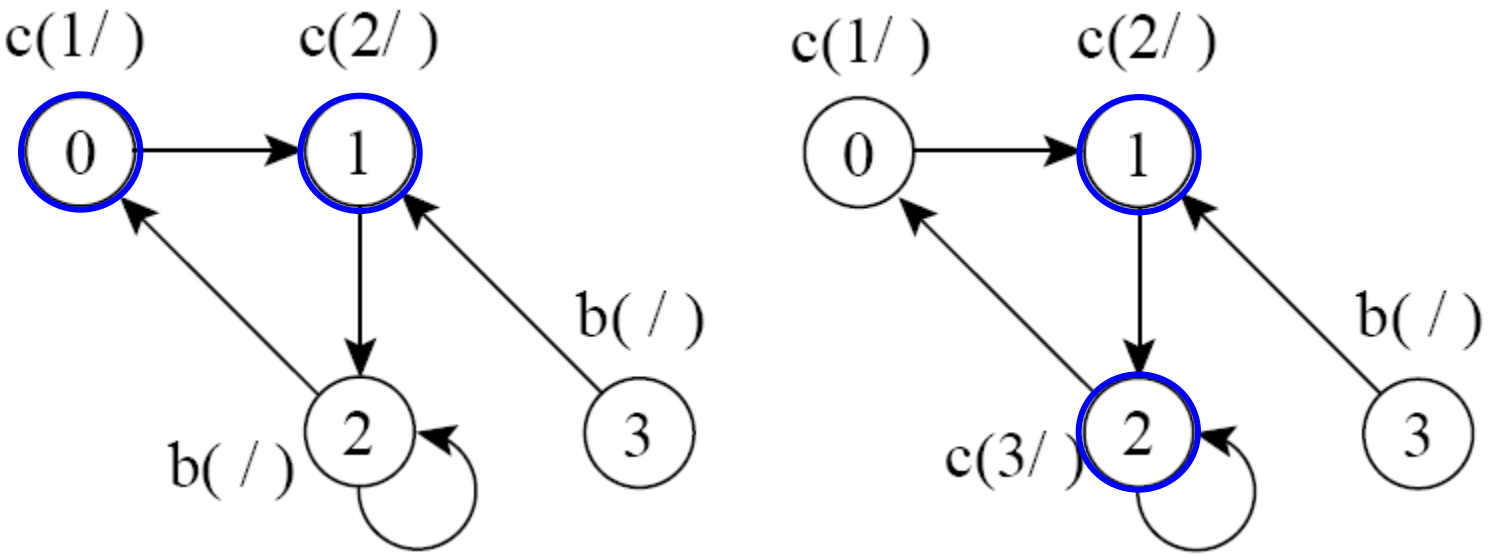




b=branco, c=cinza, p=preto, (tempo de descoberta / tempo de término do vértice)

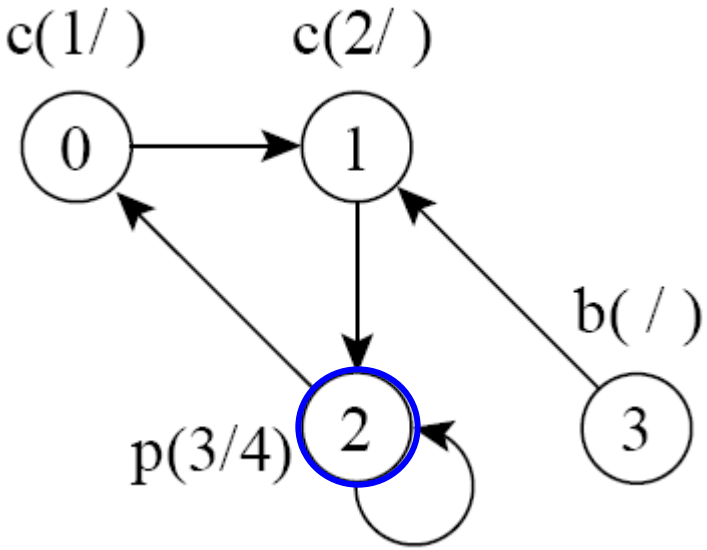
DFS

- Exemplo detalhado



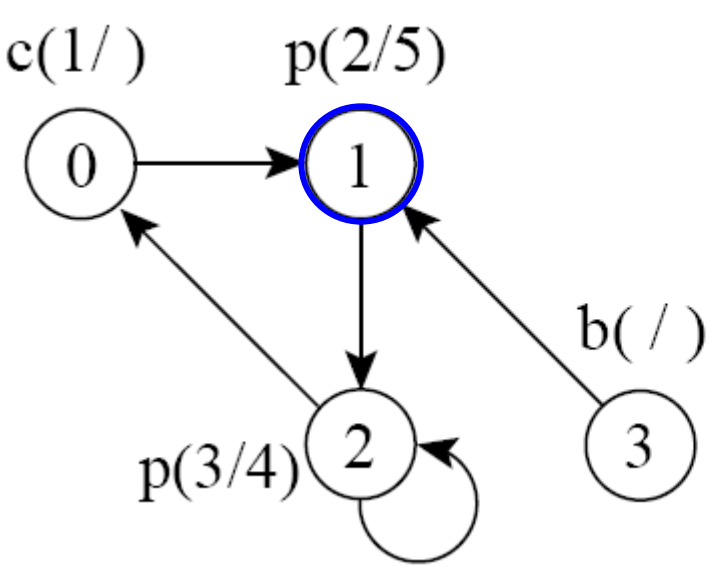
DFS

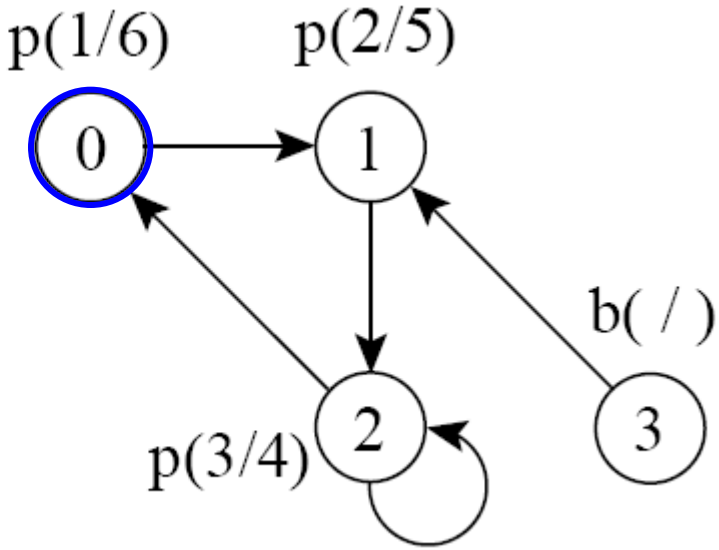
- Exemplo detalhado



DFS

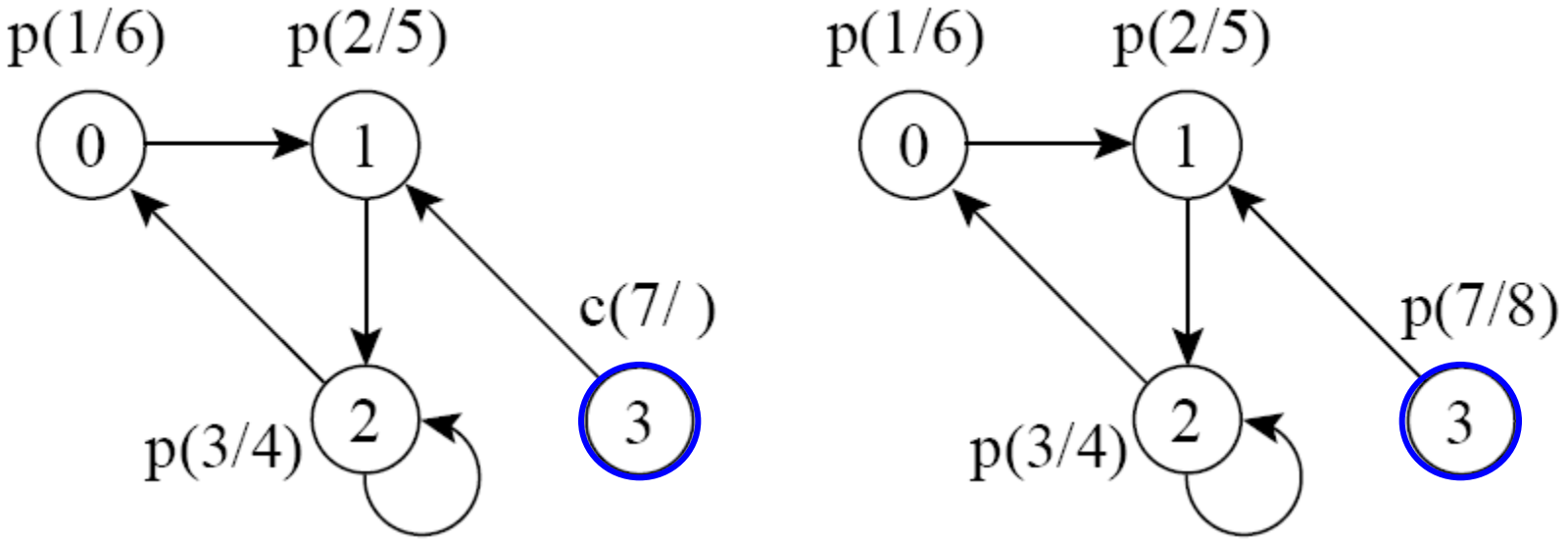
- Exemplo detalhado





DFS

- Exemplo detalhado



DFS

- Implementação da busca em profundidade

Complexidade do DFS

O(|V| + |E|)

- O DFS-visit é chamado exatamente uma vez para cada vértice de V (na pior das hipóteses)
- No DFS-visit, o laço é executado |adj[v]| vezes, i.e., O(|E|) no total

DFS

- Uma aplicação comum da DFS é determinar se um grafo é cíclico
  - Isso acontece quando, ao percorrer uma aresta, um vértice x se conecta a um vértice y que não é branco

DFS

- Uma aplicação comum da DFS é determinar se um grafo é cíclico
  - Isso acontece quando, ao percorrer uma aresta, um vértice x se conecta a um vértice y que não é branco
    - Algoritmo de busca em profundidade é facilmente adaptado para esta tarefa!
      - Como?

DFS

- Ordenação topológica de um grafo direcionado acíclico
  - Ordenação linear dos vértices do grafo tal que u aparece antes de v se há uma aresta (u,v)

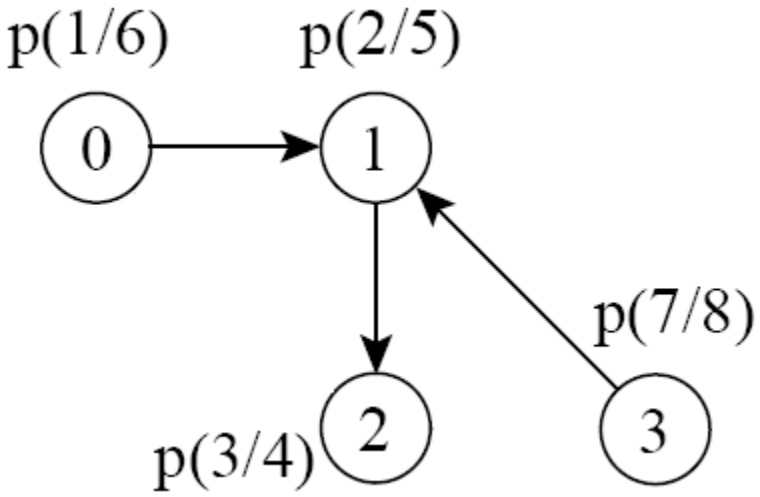
    - Qual a utilidade da ordenação topológica? Exemplos?

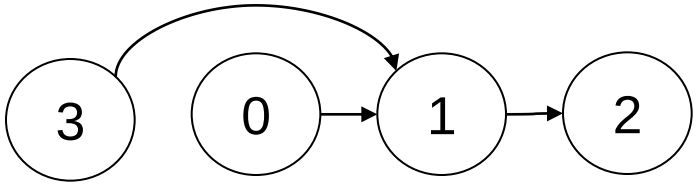
DFS

- Ordenação topológica de um grafo direcionado acíclico
  - Ordenação linear dos vértices do grafo tal que u aparece antes de v se há uma aresta (u,v)
  - Algoritmo
    1. Faça a busca em profundidade no grafo
       - Ao término de cada vértice, insira seu tempo de término no começo de uma lista linear (inicialmente vazia)
    2. Ao se percorrer a lista do começo ao final, tem-se a ordenação topológica do grafo

DFS

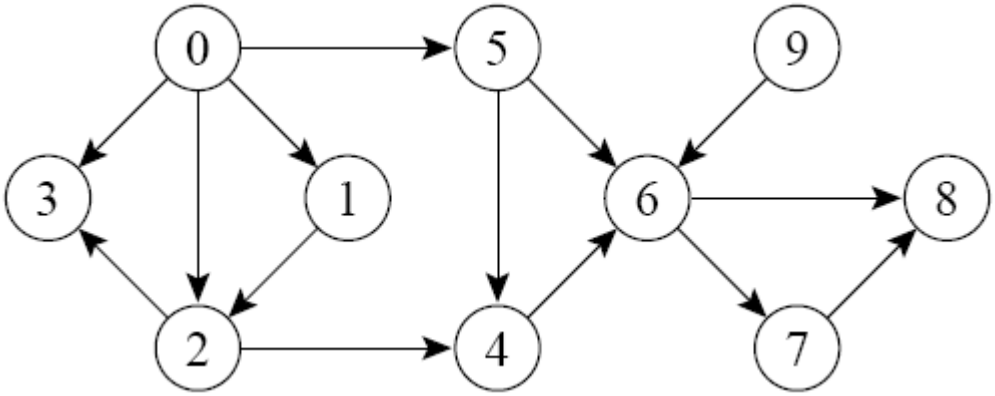
- Ordenação topológica: exemplo





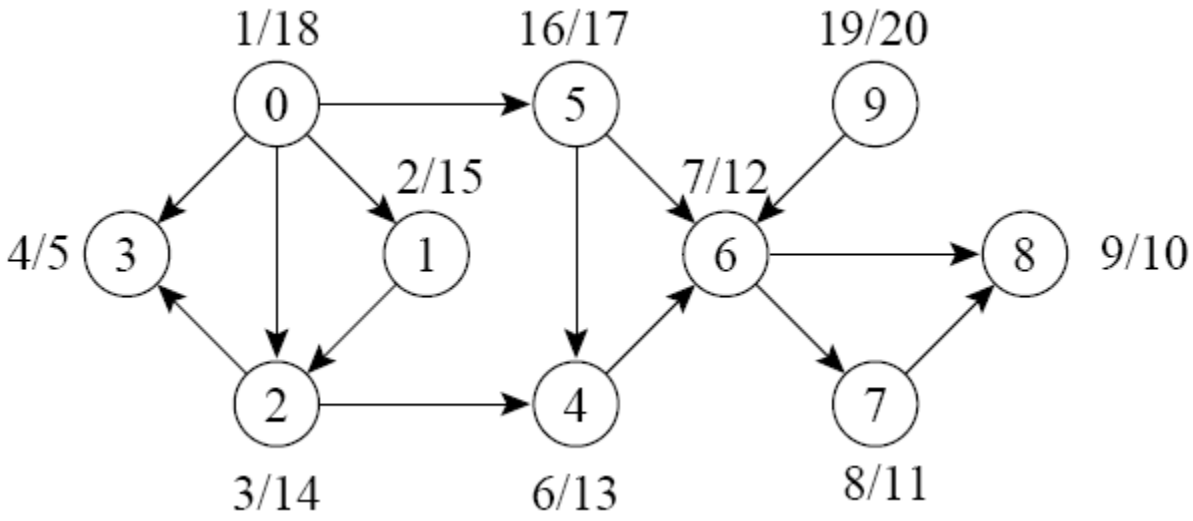
DFS

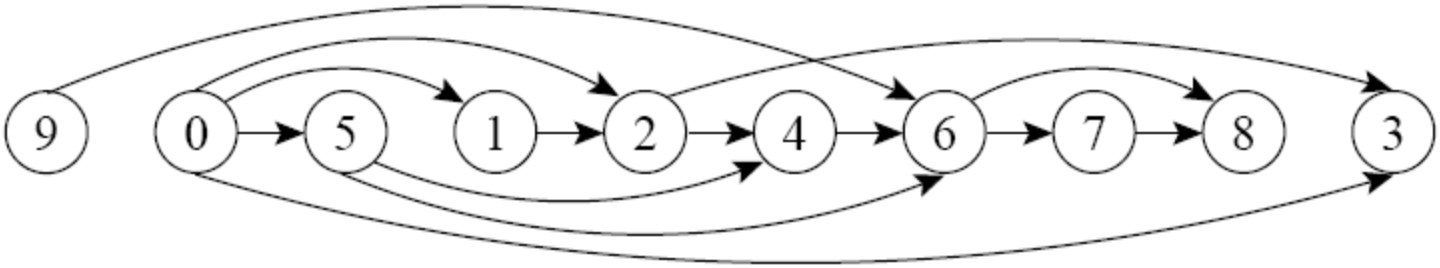
- Faça a ordenação topológica do grafo abaixo



DFS

- Faça a ordenação topológica do grafo abaixo





DFS

- Ordenação topológica de um grafo direcionado acíclico
  - Como alterar o algoritmo de busca em profundidade para realizar a ordenação topológica?
  - Qual a complexidade de tempo do algoritmo?

- Atenção
  - Não há uma única ordenação topológica
  - Não há ordenação topológica em grafos com ciclos

Exercício

- Escreva uma versão (em C) não recursiva do DFS
  - À mão e individual, para entregar na próxima aula