Grafos: Busca em Profundidade

SCC0216 Modelagem Computacional em Grafos

Thiago A. S. Pardo
Maria Cristina F. Oliveira

Percorrendo um grafo

- Há duas possibilidades
  - Busca em largura (usando uma fila)
  - **Busca em profundidade (usando uma pilha)**

Percorrendo um grafo

DFS - *Depth-First Search*

- Explora-se cada vértice do grafo em profundidade, a medida em que é visitado...
  - Procura-se avançar na busca sem olhar para todos os vértices vizinhos 'de mesmo nível'
  - Quando necessário, volta-se aos vizinhos ainda não totalmente processados antes (processo de backtracking)

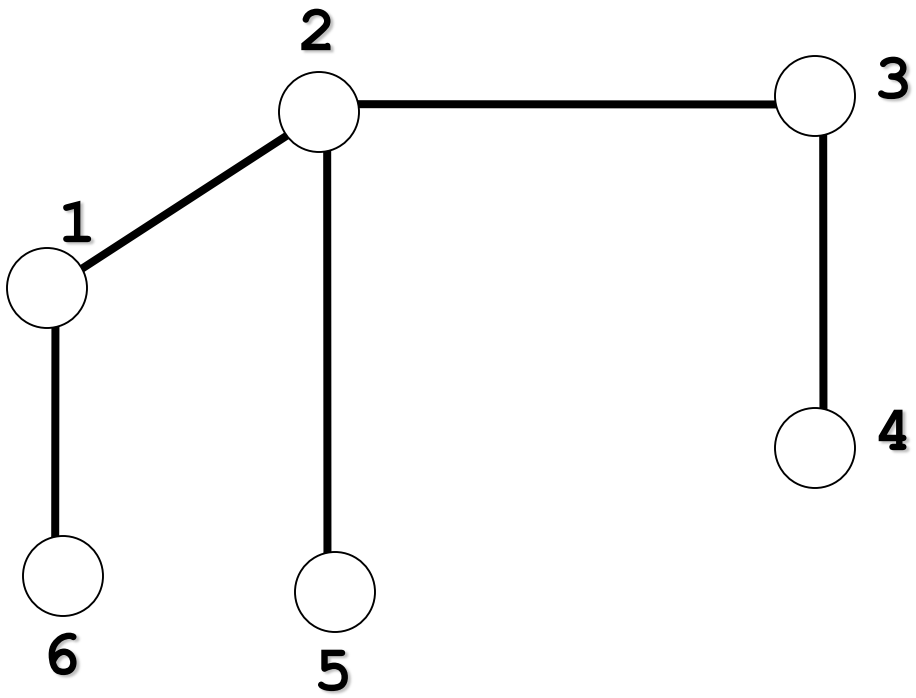
Percorrendo um grafo

DFS - *Depth-First Search*

- Pode-se adotar o mesmo esquema de rotulação por cores para guiar a busca
  - Todos os vértices são inicializados brancos
  - Quando um vértice *v* é 'descoberto' pela primeira vez, ele se torna cinza
  - Um vértice *v* torna-se preto quando todos os vértices adjacentes a ele tiverem sido 'descobertos',

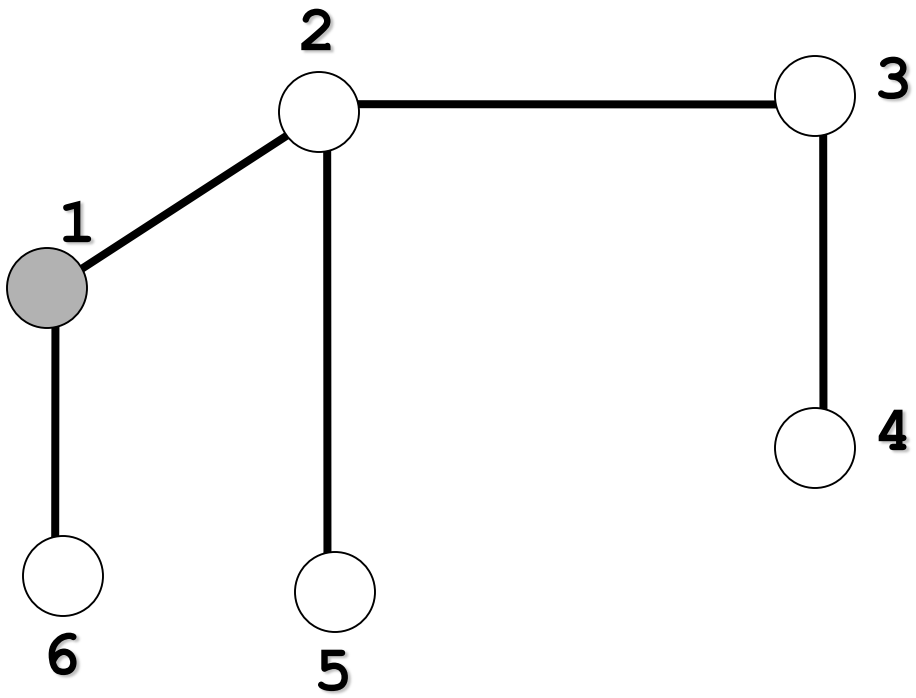
DFS – exemplo

Percorrendo um Grafo: DFS



DFS – exemplo

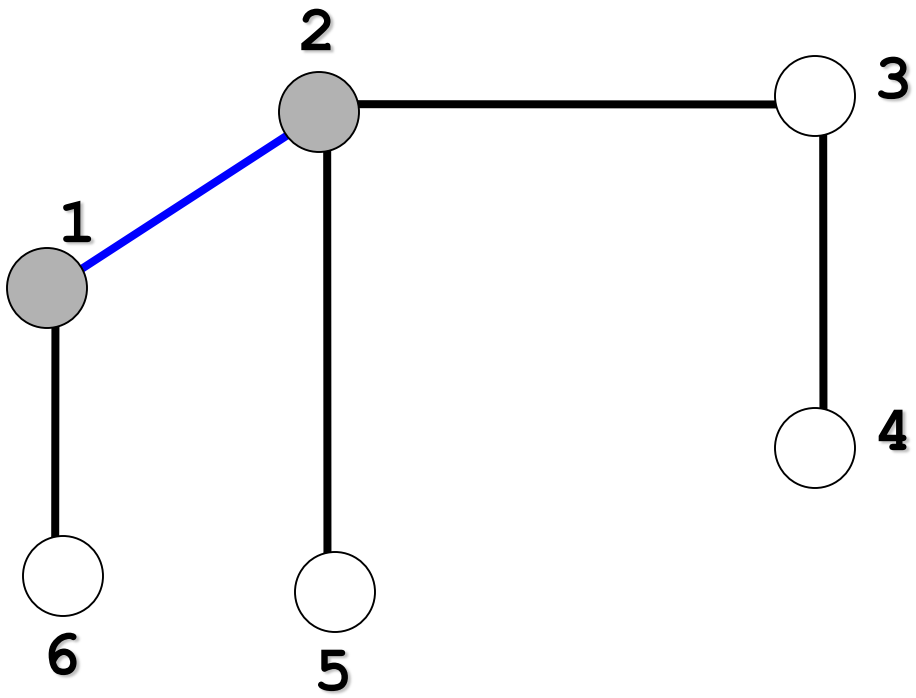
Percorrendo um Grafo: DFS



Vértice inicial: 1

DFS – exemplo

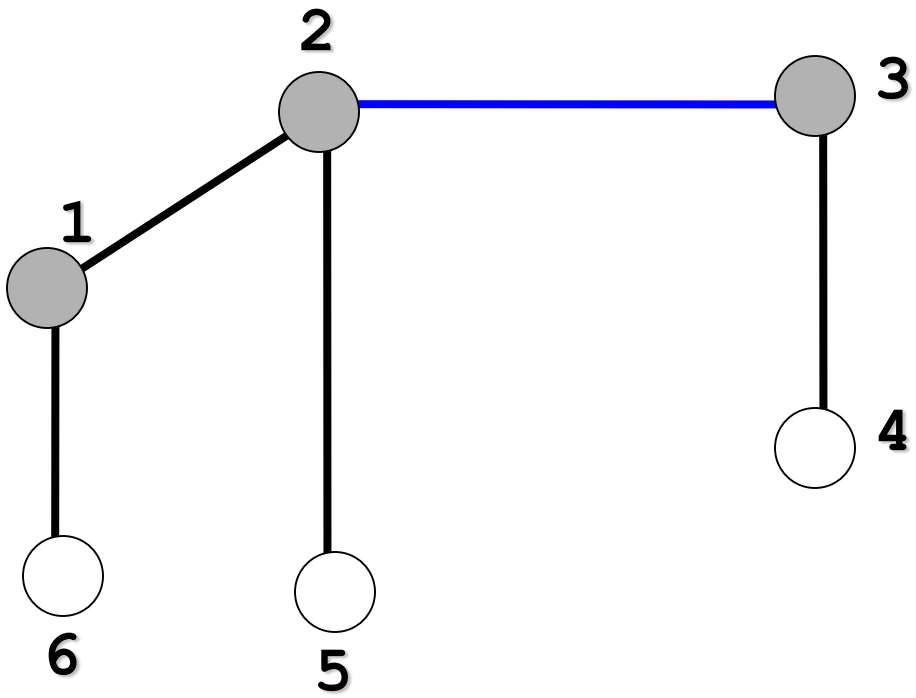
Percorrendo um Grafo: DFS



Busca-se pelo primeiro vértice adjacente a 1: 2

DFS – exemplo

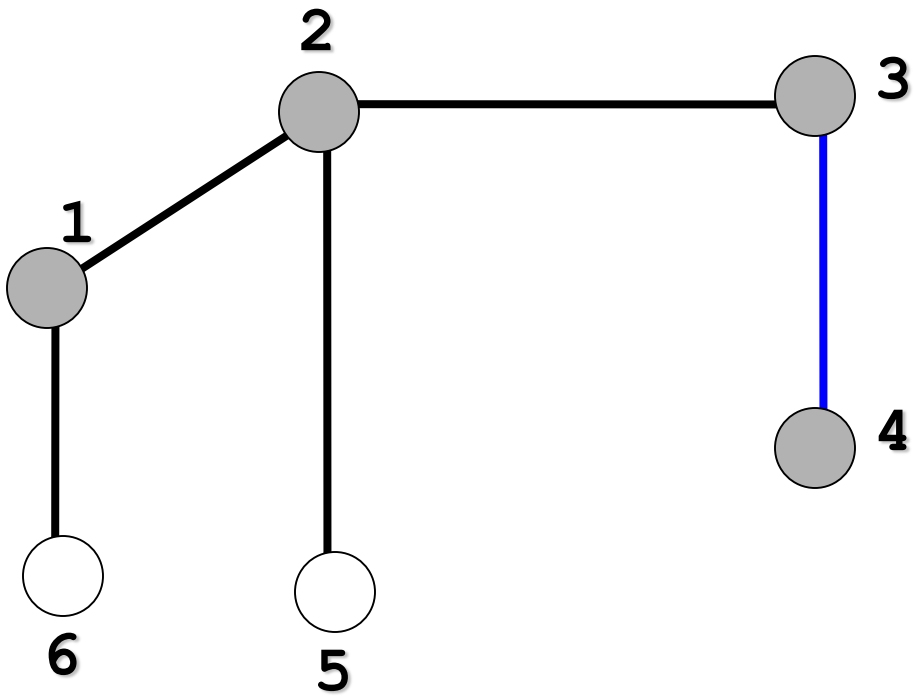
Percorrendo um Grafo: DFS



Busca-se pelo primeiro vértice adjacente a 2: 3

DFS – exemplo

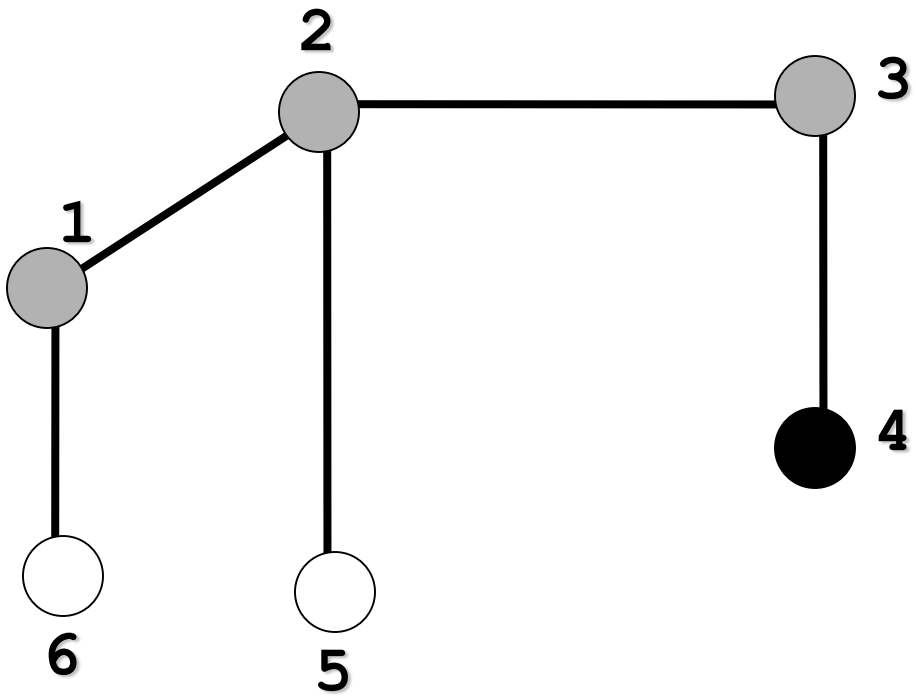
Percorrendo um Grafo: DFS



Busca-se pelo primeiro vértice adjacente a 3: 4

DFS – exemplo

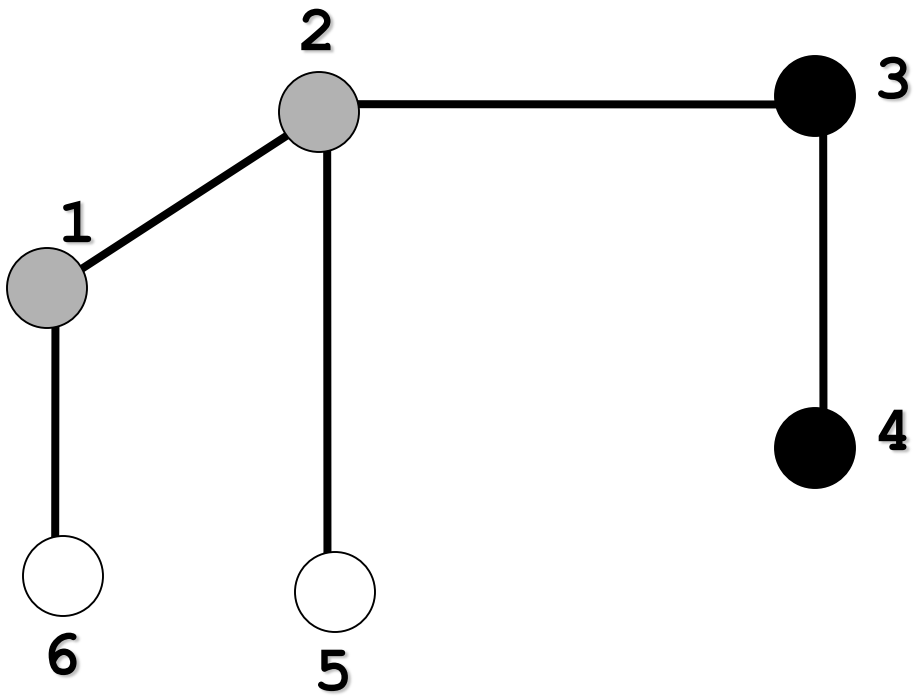
Percorrendo um Grafo: DFS



Vértice 4 não tem mais vértices adjacentes! Retorna-se ao anterior.

DFS – exemplo

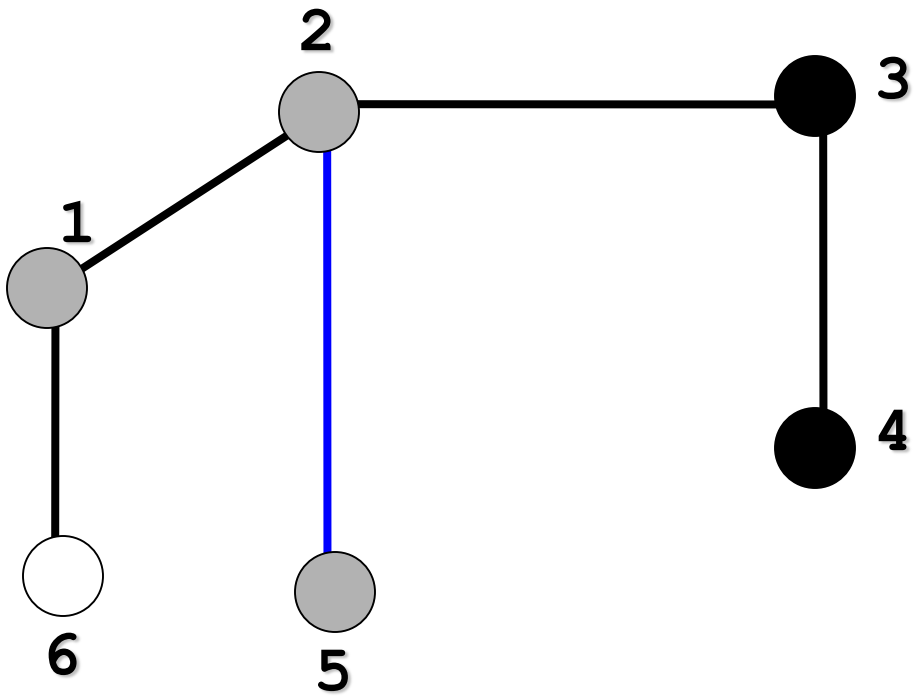
Percorrendo um Grafo: DFS



Vértice 3 não tem mais vértices adjacentes! Retorna-se ao anterior: 2

DFS – exemplo

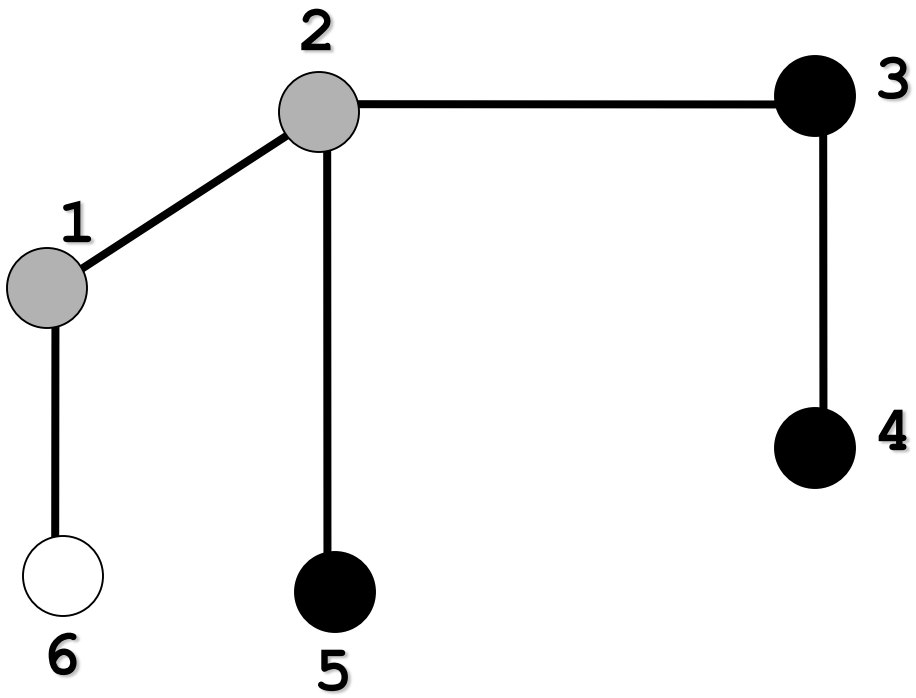
Percorrendo um Grafo: DFS



Busca-se pelo outro vértice adjacente a 2: 5

DFS – exemplo

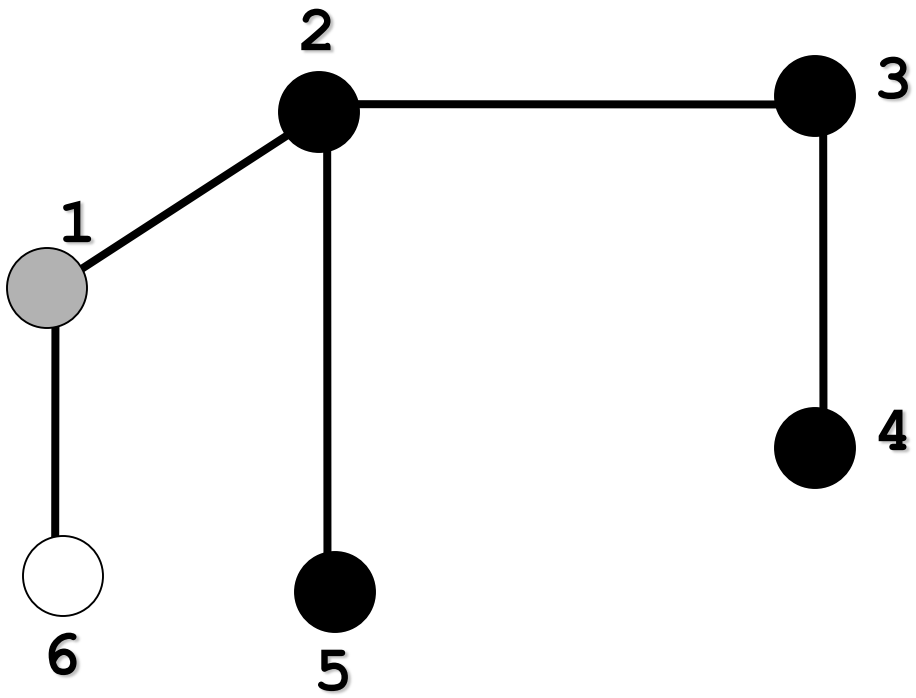
Percorrendo um Grafo: DFS



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

DFS – exemplo

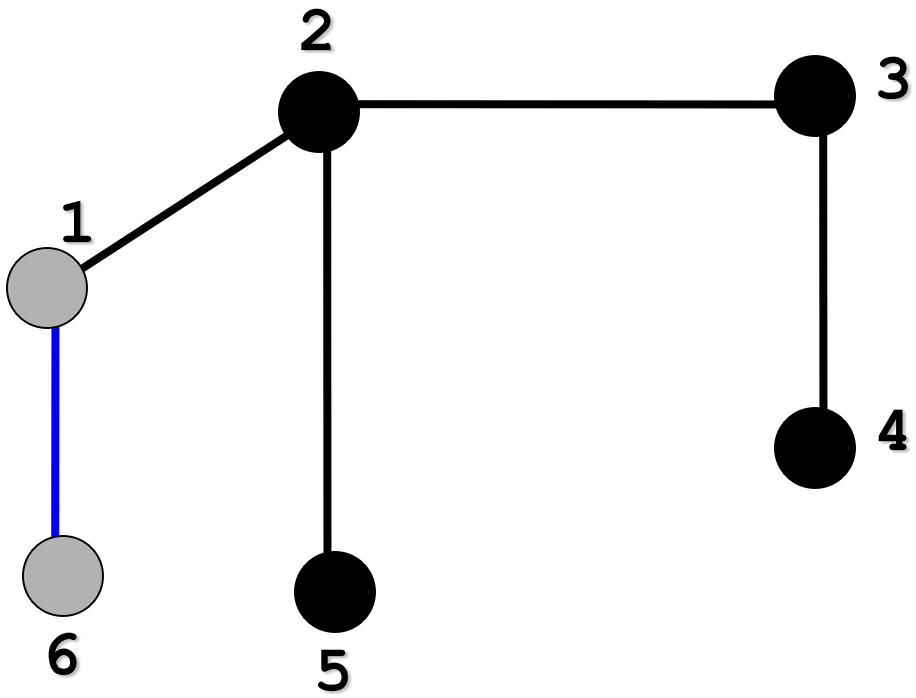
Percorrendo um Grafo: DFS



Vérrice 2 não tem mais vértices adjacentes! Retorna-se ao anterior: 1

DFS – exemplo

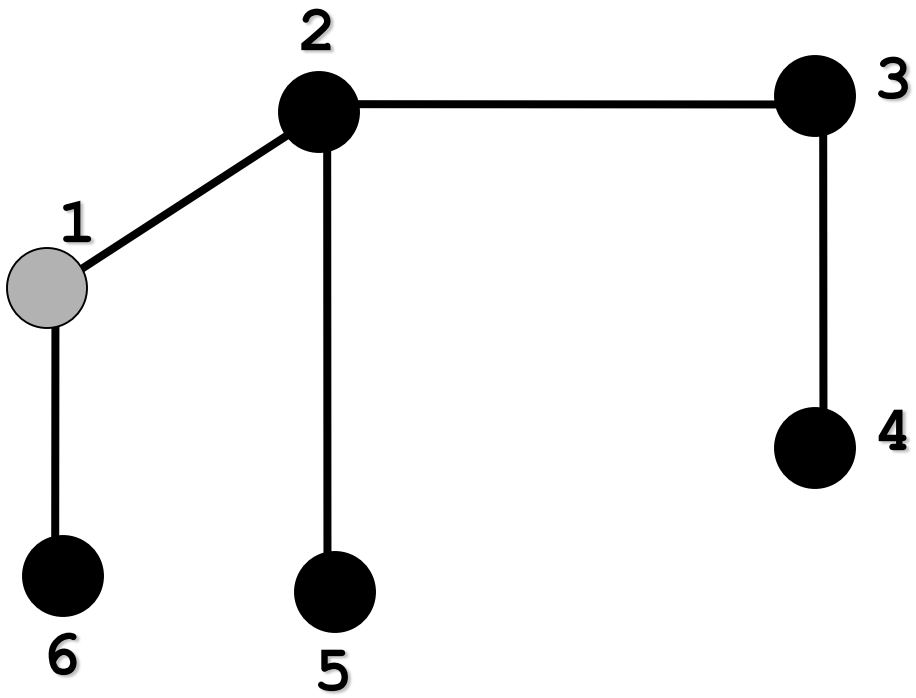
Percorrendo um Grafo: DFS



Busca-se pelo outro vértice adjacente a 1: 6

DFS – exemplo

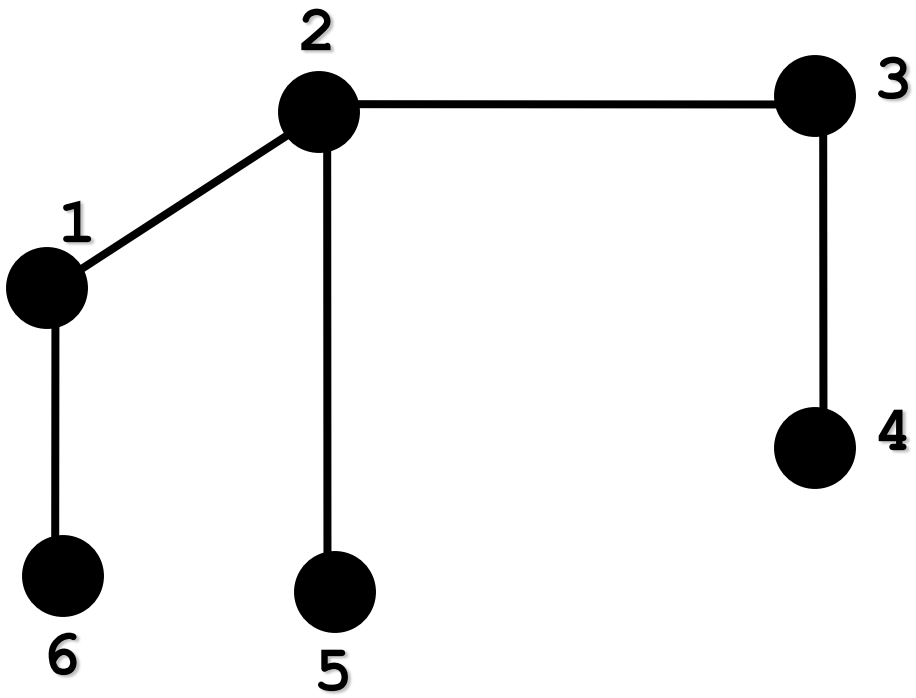
Percorrendo um Grafo: DFS



Vértice 6 não tem mais vértices adjacentes! Retorna-se ao anterior: 1

DFS – exemplo

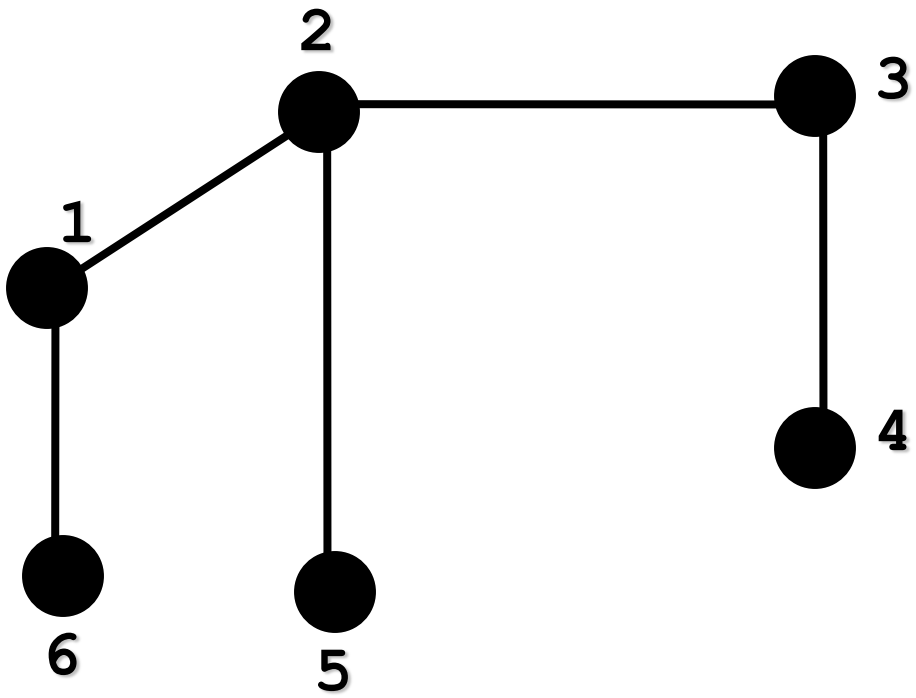
Percorrendo um Grafo: DFS



Vértice 1 não tem mais vértices adjacentes! Fim da busca!

DFS – exemplo

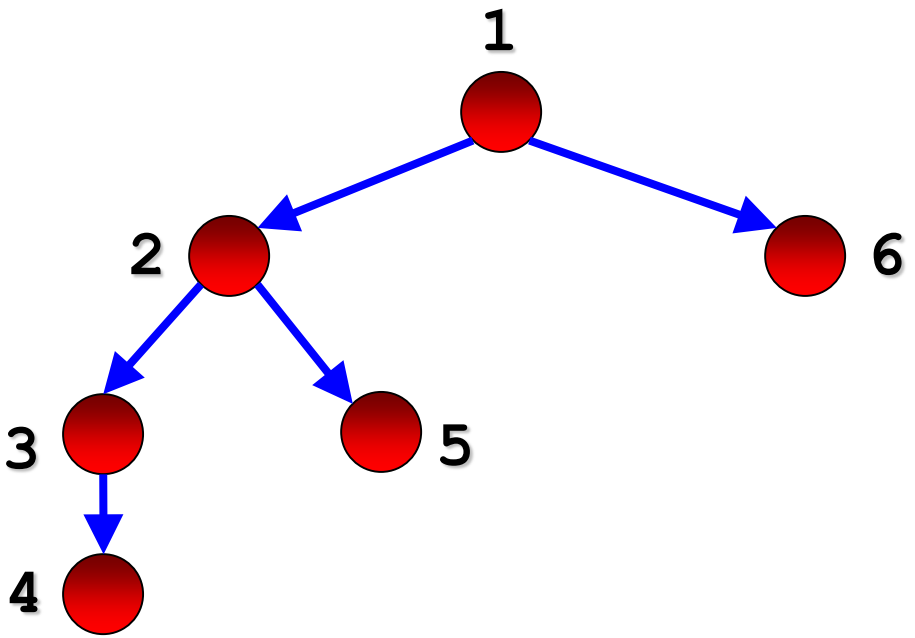
Percorrendo um Grafo: DFS



Sequencia dos vértices (processados): 4, 3, 5, 2, 6, 1

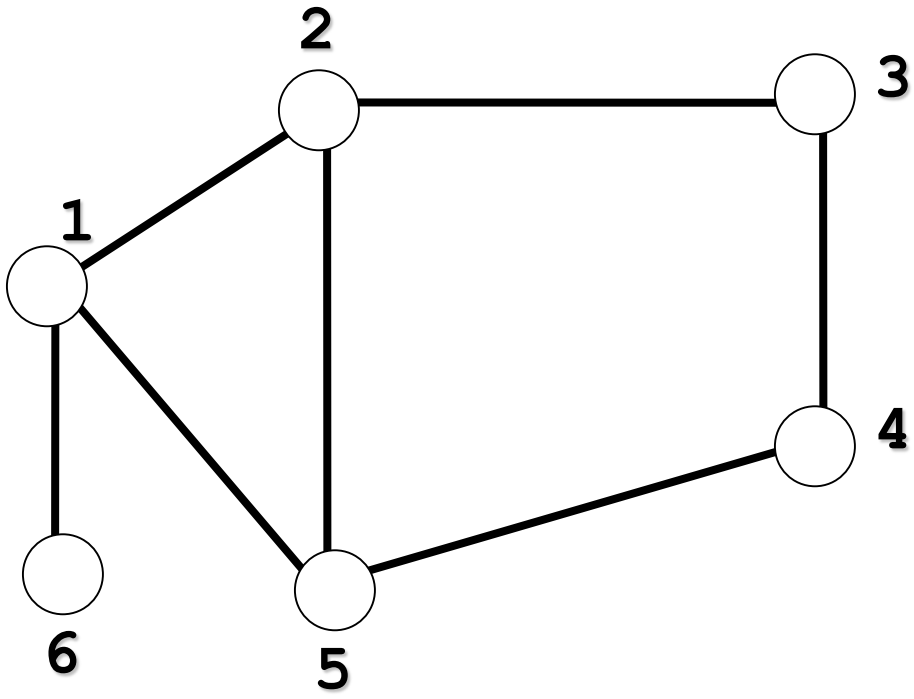
Árvore de busca em profundidade

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



Todas as arestas foram percorridas: é um mero acaso!

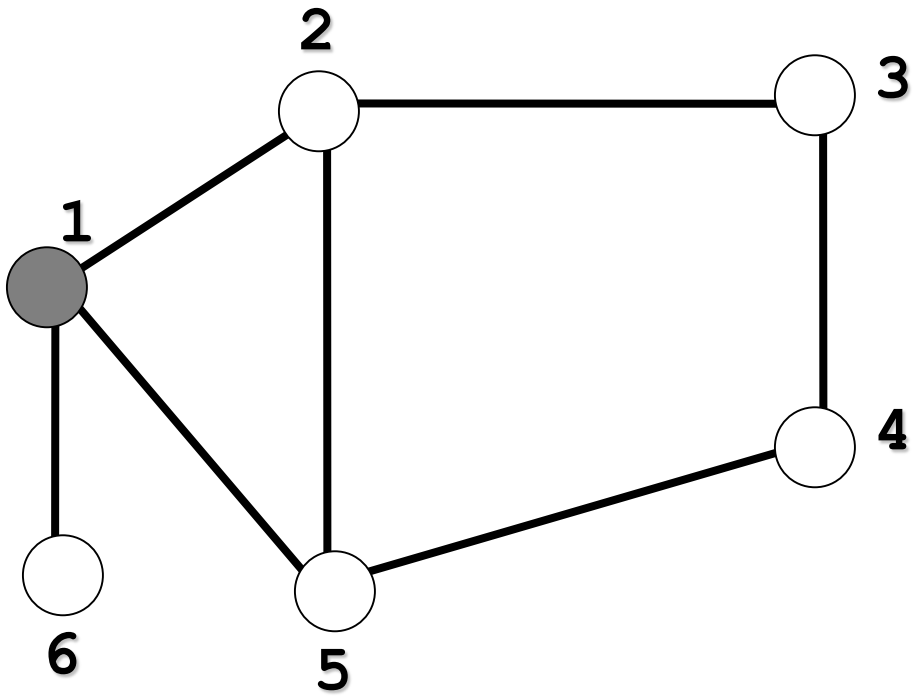
DFS – exemplo 2



Nós descobertos (cinza)

Nós processados (preto)

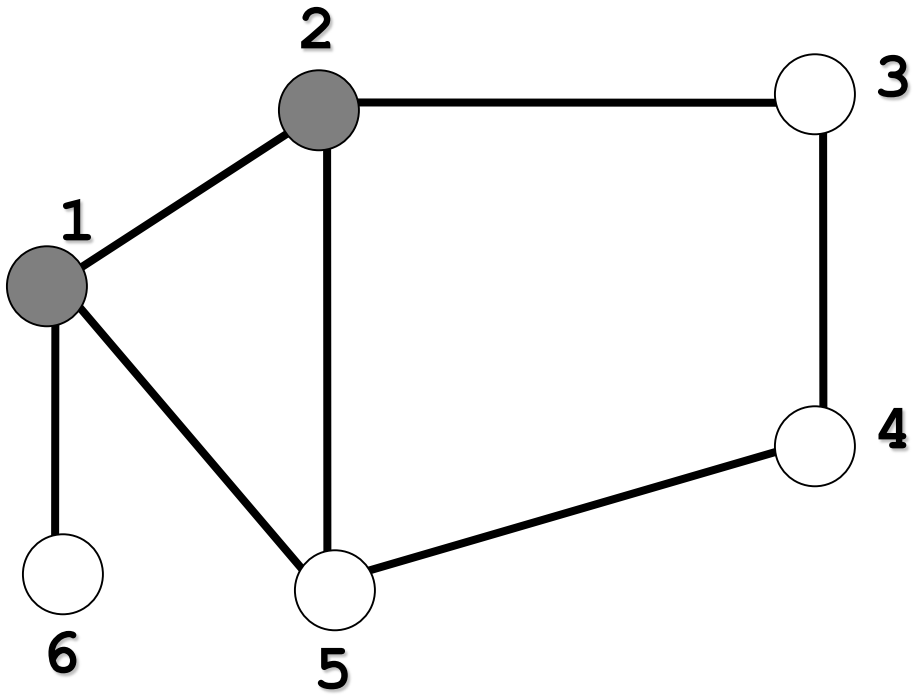
DFS – exemplo 2



Nós descobertos (cinza): 1

Nós processados (preto):

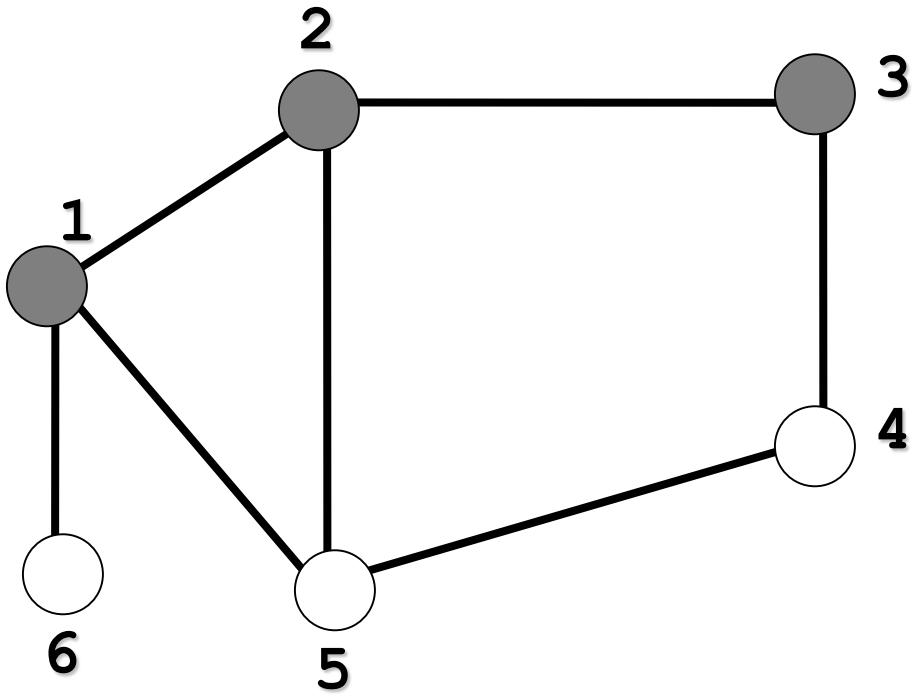
DFS – exemplo 2



Nós descobertos (cinza): 1 2

Nós processados (preto):

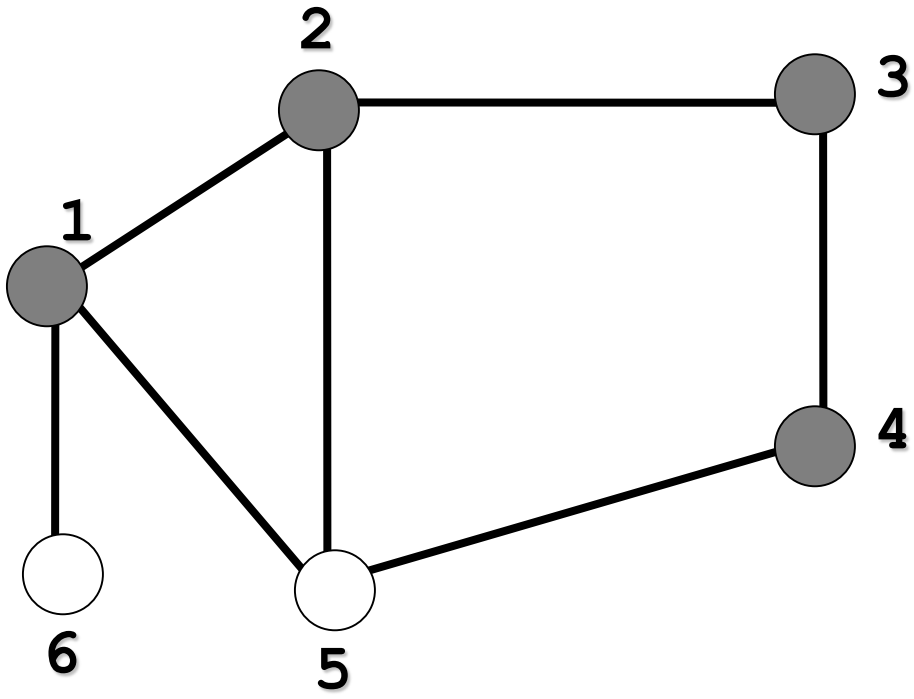
DFS – exemplo 2



Nós descobertos (cinza): 1 2 3

Nós processados (preto):

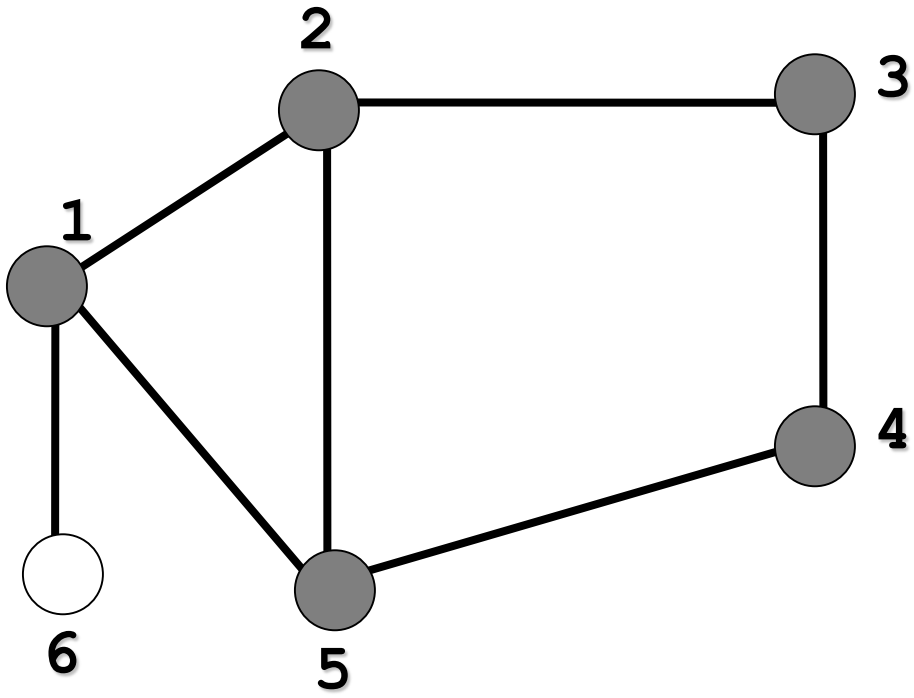
DFS – exemplo 2



Nós descobertos (cinza): 1 2 3 4

Nós processados (preto):

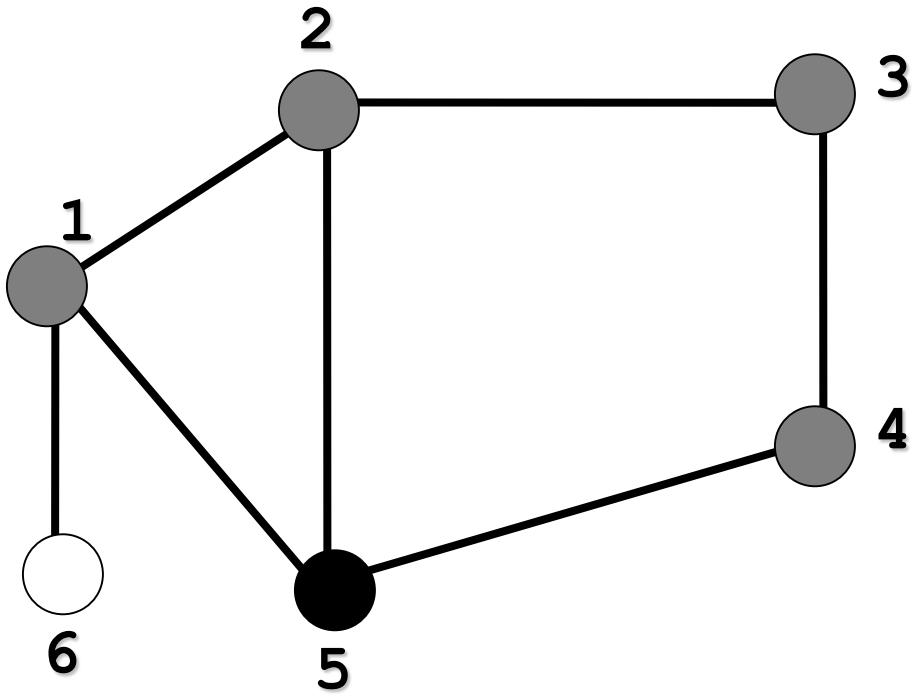
DFS – exemplo 2



Nós descobertos (cinza): 1 2 3 4 5

Nós processados (preto):

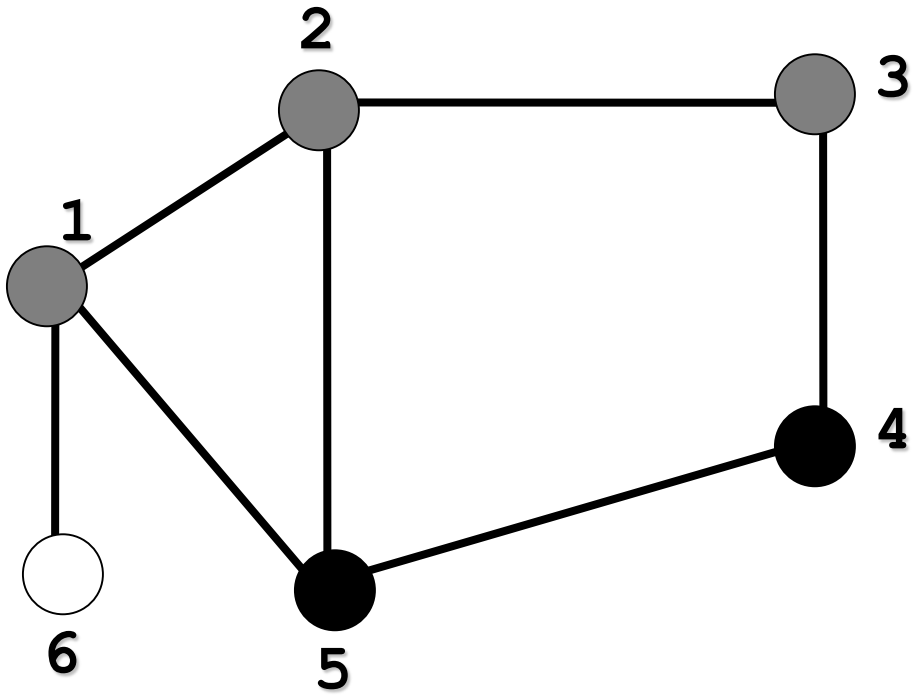
DFS – exemplo 2



Nós descobertos (cinza): 1 2 3 4 5

Nós processados (preto): 5

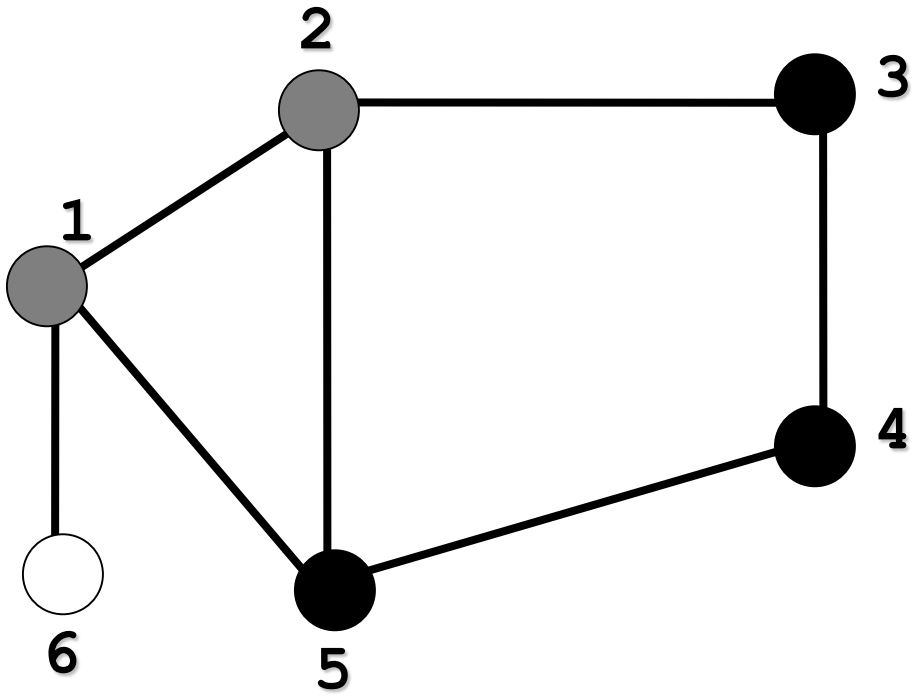
DFS – exemplo 2



Nós descobertos (cinza): 1 2 3 4 5

Nós processados (preto): 5 4

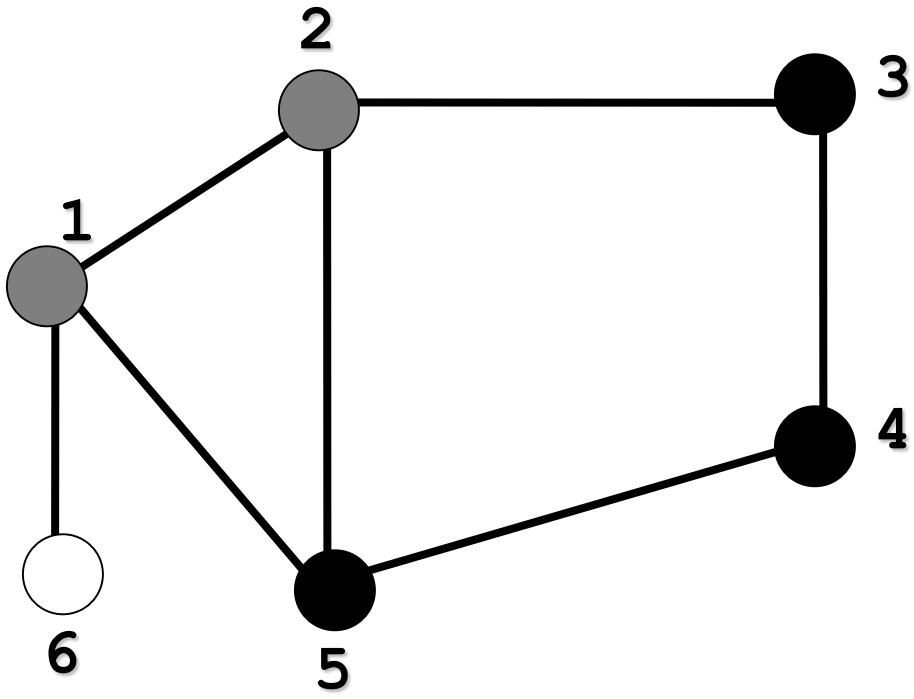
DFS – exemplo 2



Nós descobertos (cinza): 1 2 3 4 5

Nós processados (preto): 5 4 3

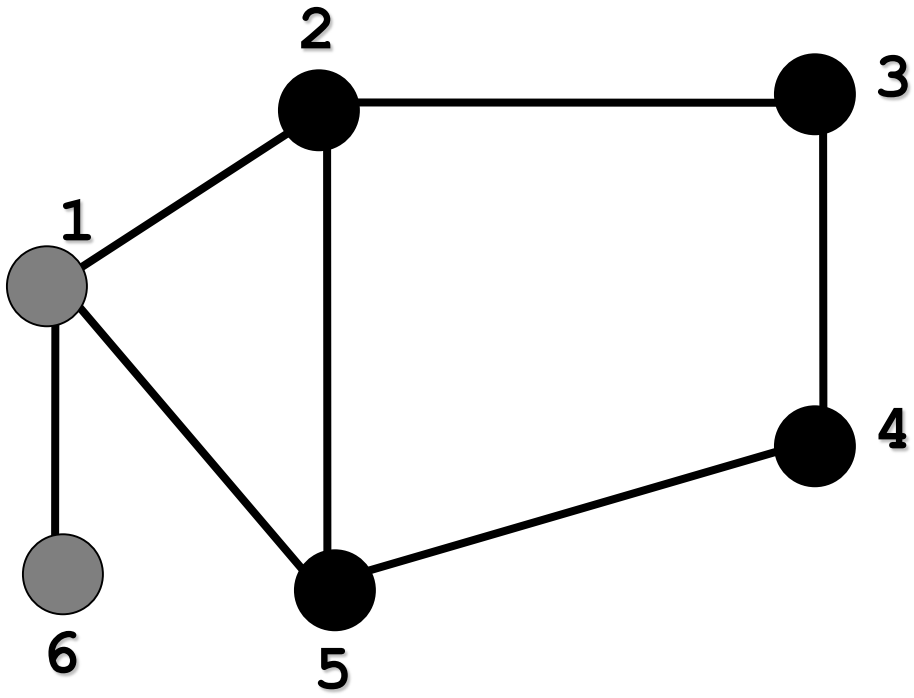
DFS – exemplo 2



Nós descobertos (cinza): 1 2 3 4 5

Nós processados (preto): 5 4 3

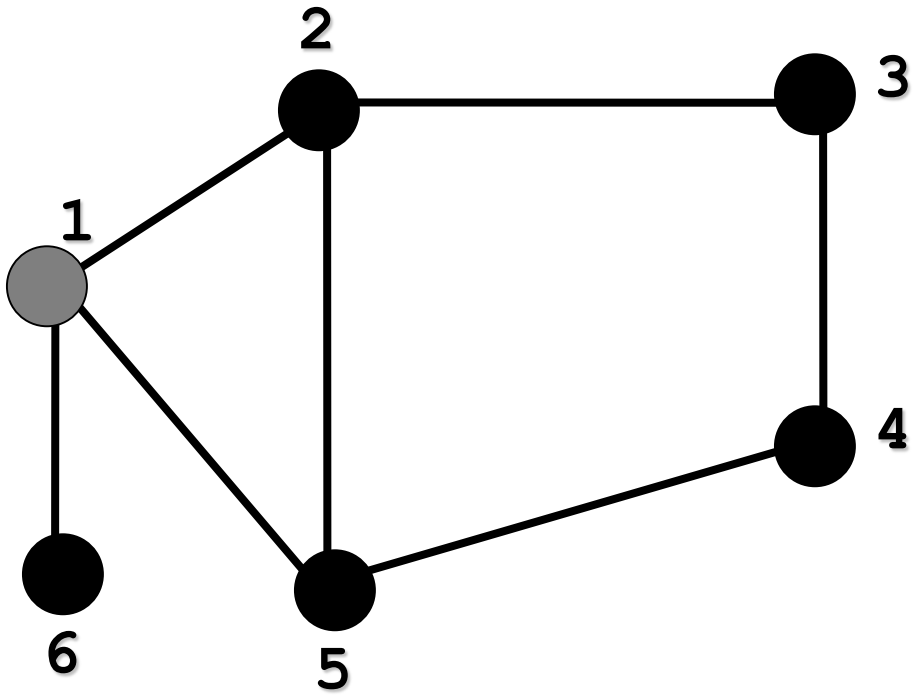
DFS – exemplo 2



Nós descobertos (cinza): 1 2 3 4 5 6

Nós processados (preto): 5 4 3 2

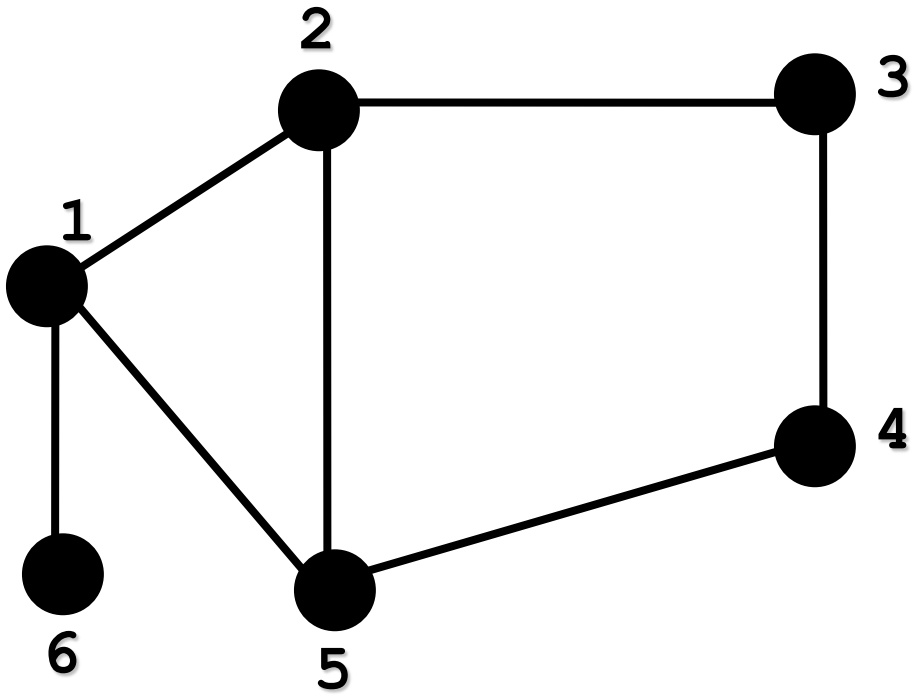
DFS – exemplo 2



Nós descobertos (cinza): 1 2 3 4 5 6

Nós processados (preto): 5 4 3 2 1

DFS – exemplo 2

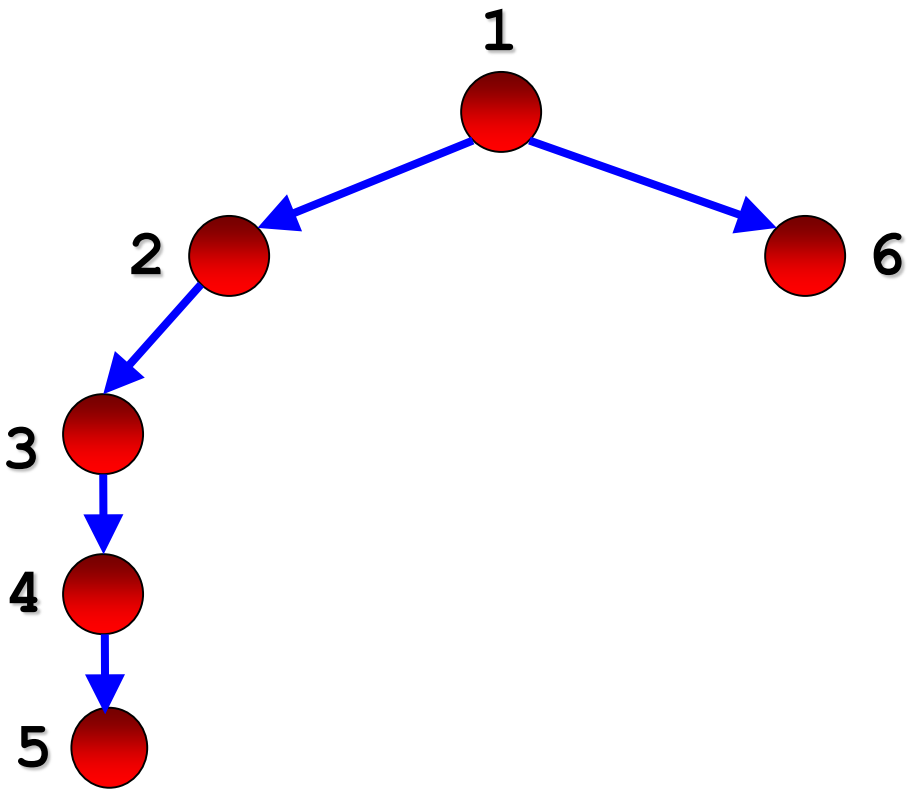


Nós descobertos (cinza): 1 2 3 4 5 6

Nós processados (preto): 5 4 3 2 6 1

Árvore de busca em profundidade

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



Nem todas as arestas foram percorridas!

DFS

- Algoritmo
  - Usa uma pilha para organizar os vértices a serem percorridos
    - Pilha pode ser implícita (via recursão) ou explícita

DFS - algoritmo

```
procedure DFS(G, v) is
    label v as gray
    for all edges (v,w) in G.adjacentEdges(v) do
        if vertex w not labeled as gray then
            call DFS(G, w)
    label v as black
```

DFS

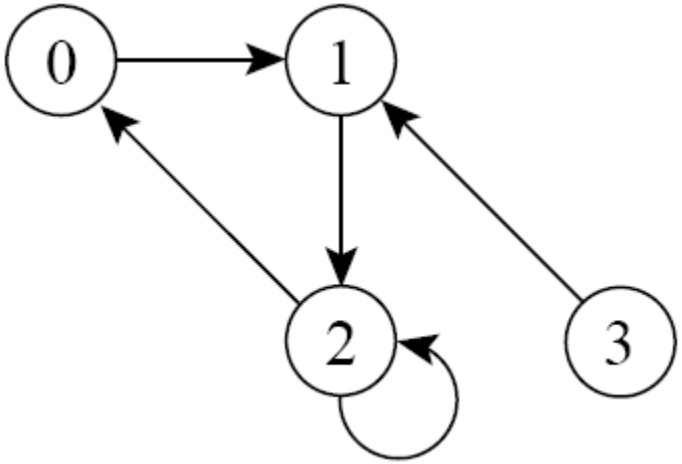
- Algoritmo
  - Usa uma pilha para organizar os vértices a serem percorridos
    - Pilha pode ser implícita (via recursão) ou explícita

  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

  - Costuma-se registrar o tempo de descoberta (cinza) e o tempo de término da busca (preto) de cada vértice

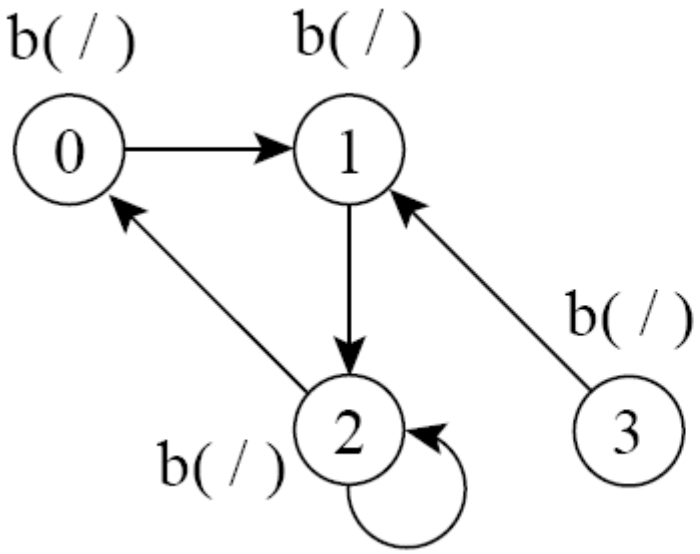
DFS

- Exemplo detalhado
  - Status de cada vértice, tempo de descoberta e de término



DFS

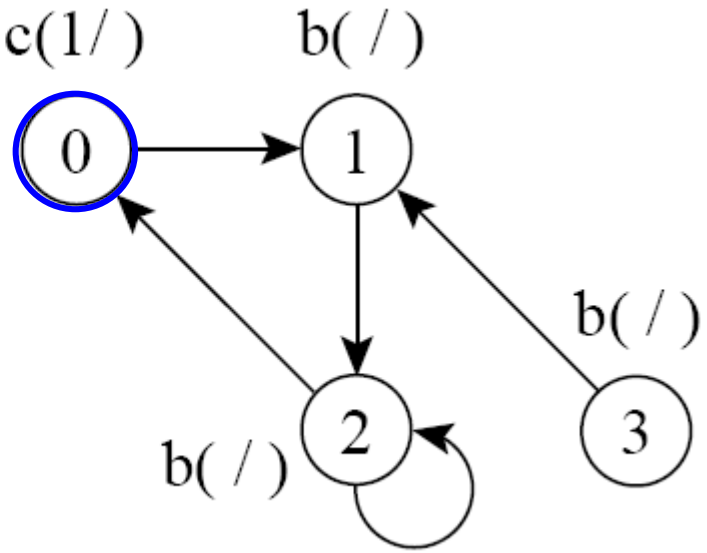
- Exemplo detalhado



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

DFS

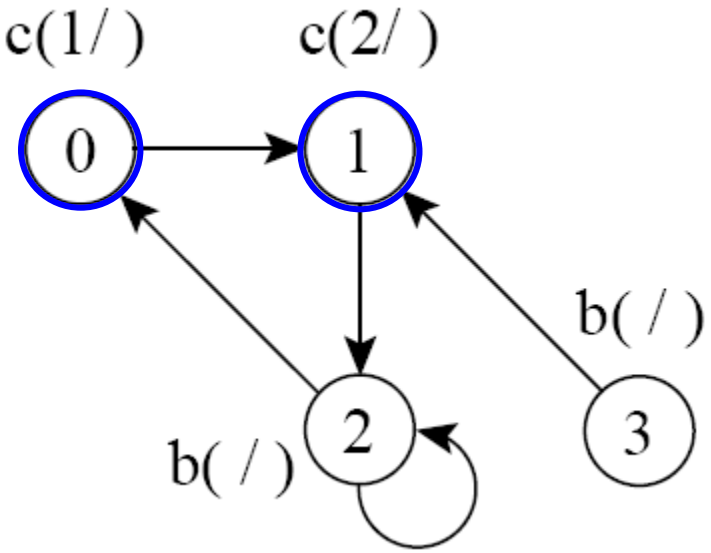
- Exemplo detalhado



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

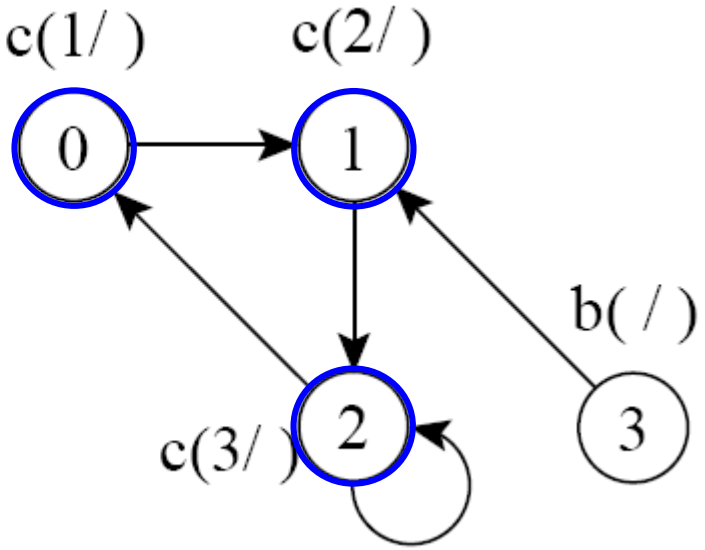
DFS

- Exemplo detalhado



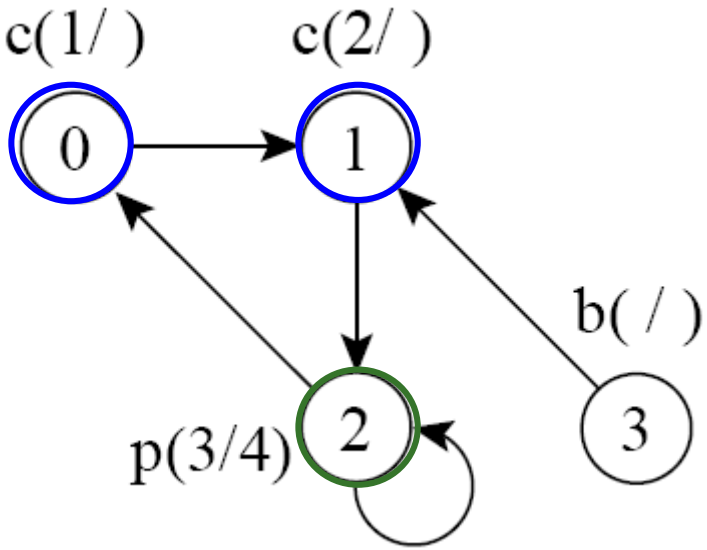
DFS

- Exemplo detalhado



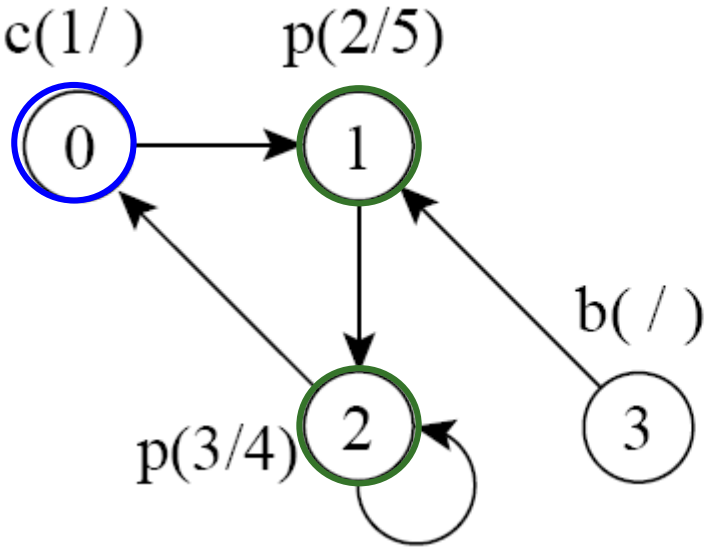
DFS

- Exemplo detalhado



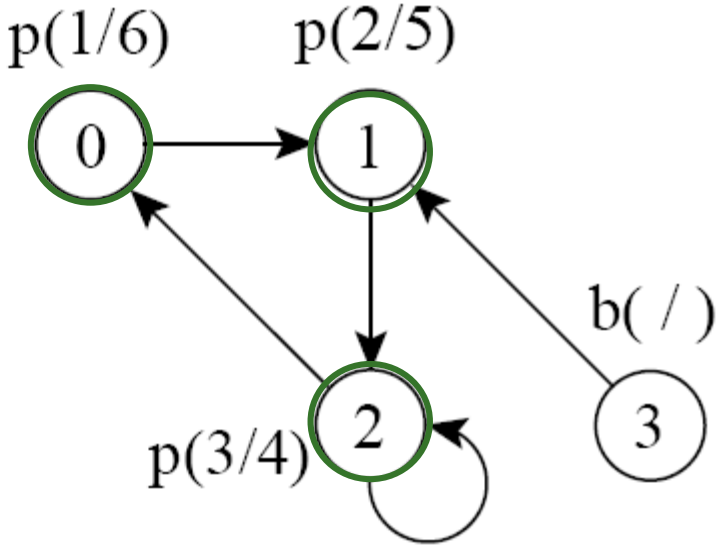
DFS

- Exemplo detalhado



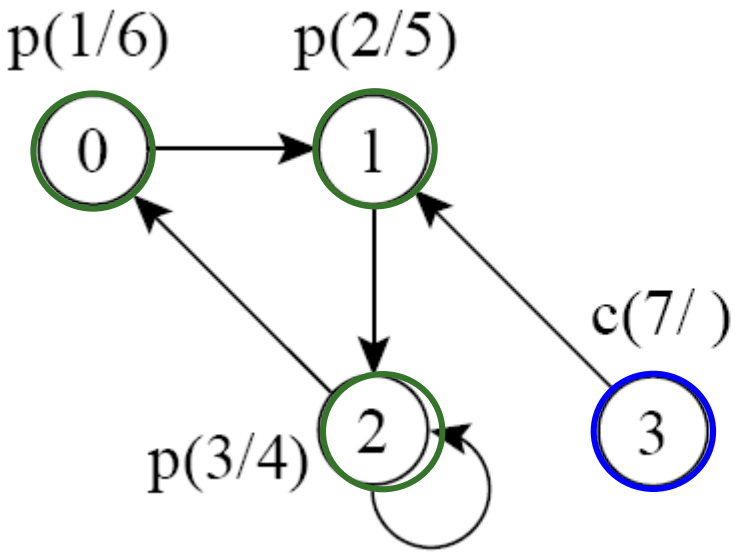
DFS

- Exemplo detalhado



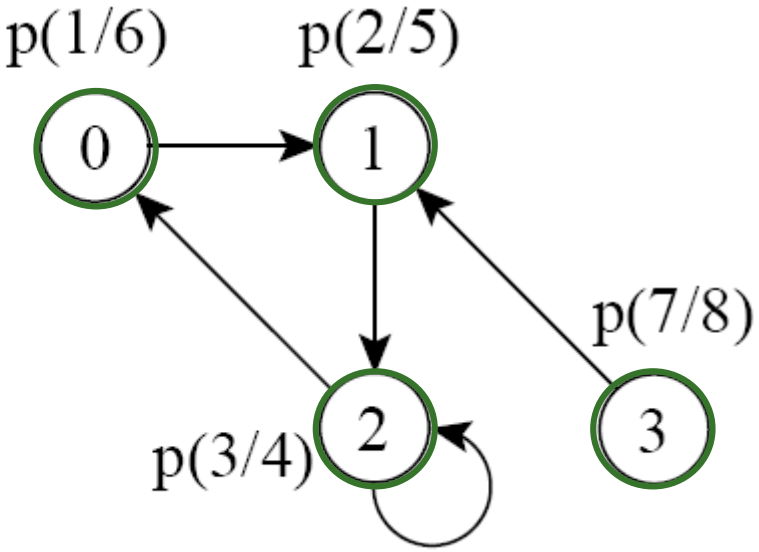
DFS

- Exemplo detalhado



DFS

- Exemplo detalhado



Complexidade do DFS

O(|V| + |A|)

- A função DFS é chamada exatamente uma vez para cada vértice de V
- A cada chamada da função, o laço é executado |adj[v]| vezes, i.e., ele será executado O(|A|) vezes no total

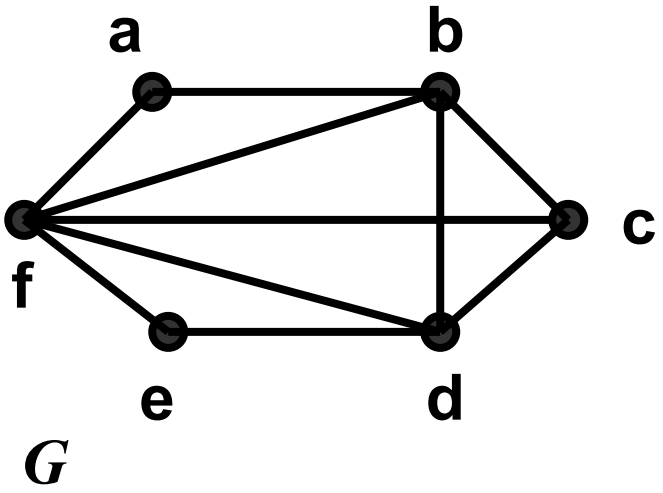
DFS - algoritmo

```
procedure DFS(G, v) is
    label v as gray
    for all edges (v,w) in G.adjacentEdges(v) do
        if vertex w not labeled as gray then
            call DFS(G, w)
    label v as black
```

Exercícios

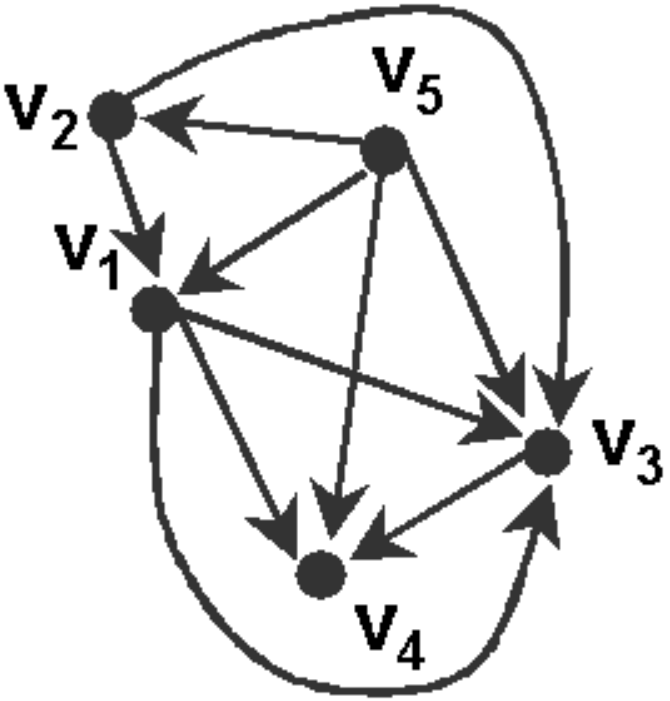
DFS

- Faça a busca em profundidade no grafo abaixo



DFS

- Faça a busca em profundidade no grafo abaixo



DFS

- Implemente a busca em profundidade