

# ALGORITMOS EM GRAFOS

---

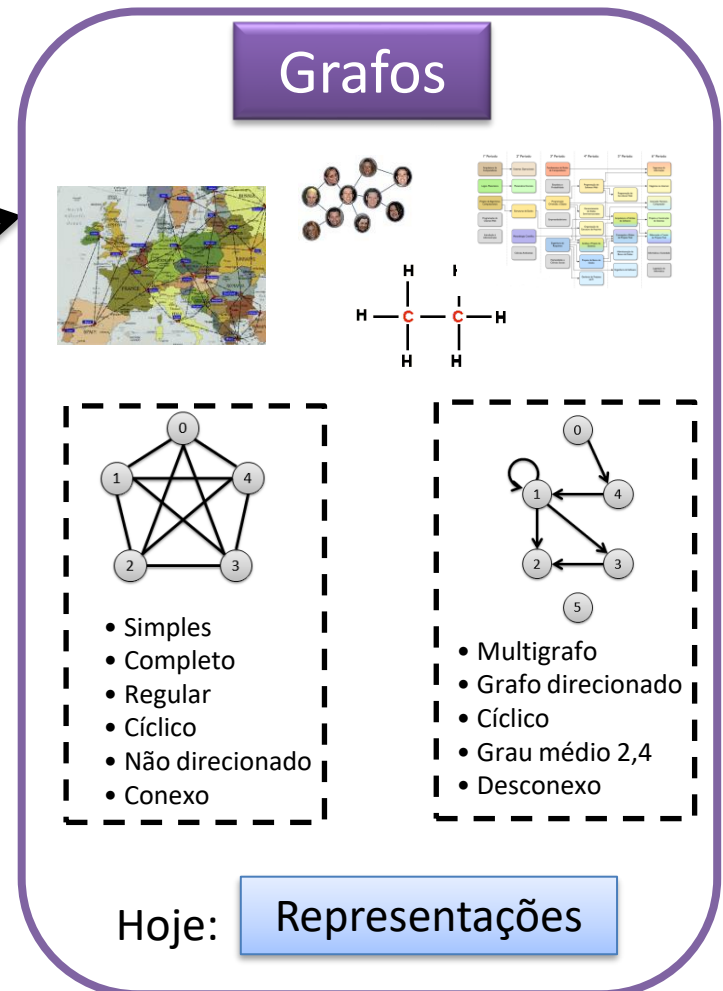
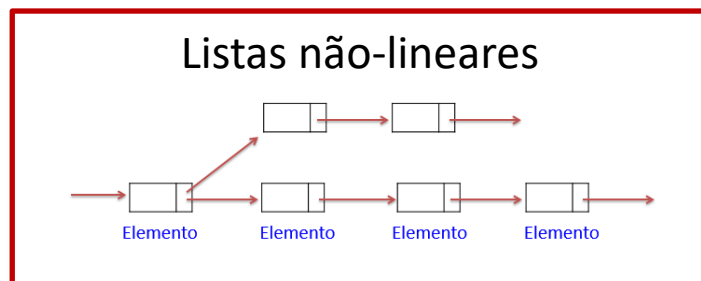
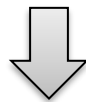
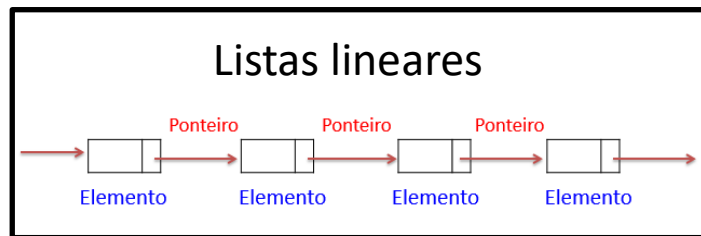
*Bacharelado em Sistemas da Informação*

*Prof. Marco André Abud Kappel*

**Aula 2 – Representações de grafos**

# Representações de grafos

- Na última aula:



# Representações de grafos

---

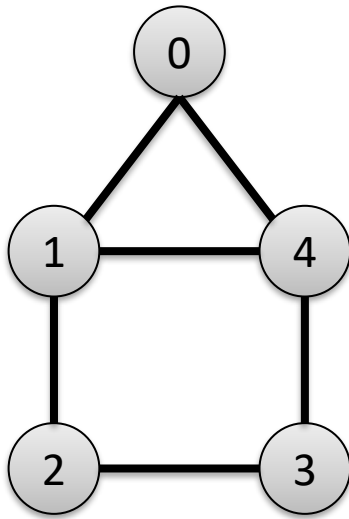
- **Matriz de Adjacências**

- Supondo que os vértices são **enumerados**, um grafo pode ser representado por sua **matriz de adjacência A**.
- Seja um grafo  $G=(V,E)$  e uma matriz  $A_{n \times n} = [a_{ij}]$ , onde  $n$  é o número de nós do grafo:
  - $A(i, j) = 1$ , se  $(i, j) \in E$
  - $A(i, j) = 0$ , caso contrário
- Informa se existe um **caminho de comprimento 1** entre dois nós.

# Representações de grafos

- **Matriz de Adjacências**

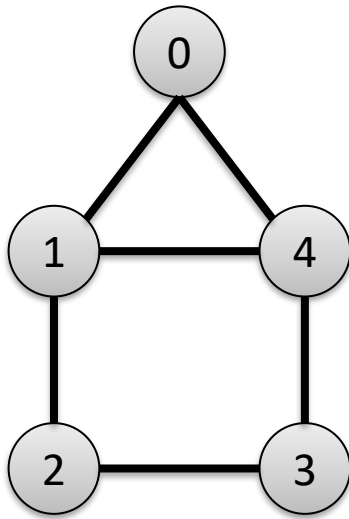
- Exemplos:



# Representações de grafos

- **Matriz de Adjacências**

– Exemplos:



Nós



	0	1	2	3	4
0					
1					
2					
3					
4					

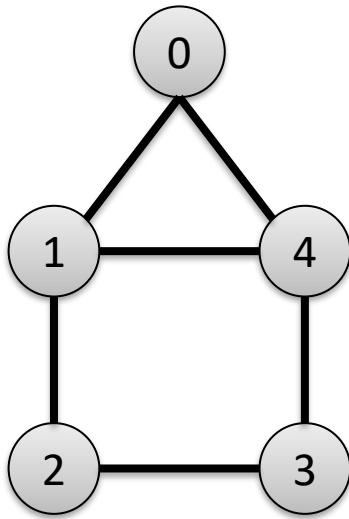
Nós



# Representações de grafos

- **Matriz de Adjacências**

– Exemplos:

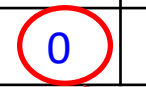


Nós



	0	1	2	3	4
0	0	1	0	0	1
1	1	0	1	0	1
2	0	1	0	1	0
3	0	0	1	0	1
4	1	1	0	1	0

Nós

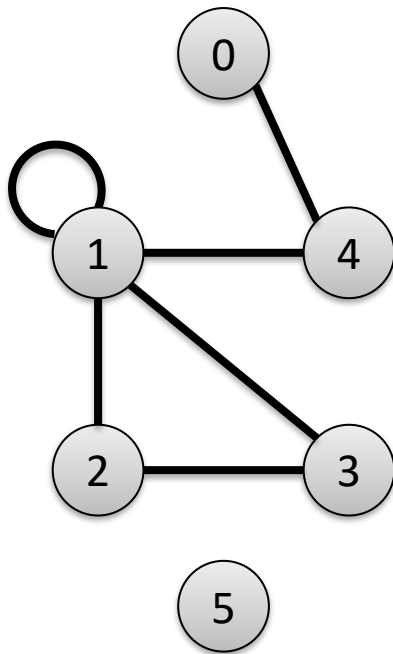


Número de  
arestas entre  
estes nós

# Representações de grafos

- **Matriz de Adjacências**

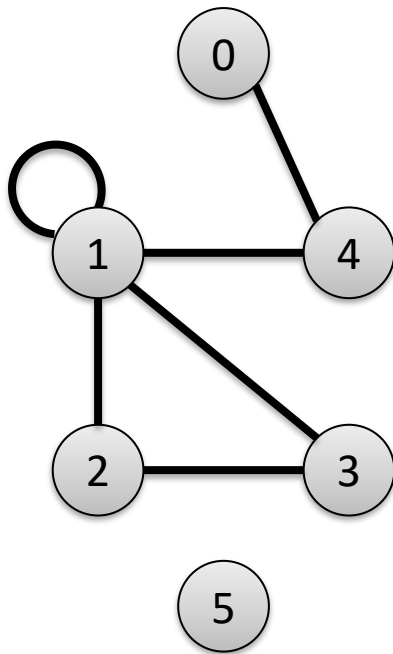
— Exemplos:



# Representações de grafos

- **Matriz de Adjacências**

– Exemplos:



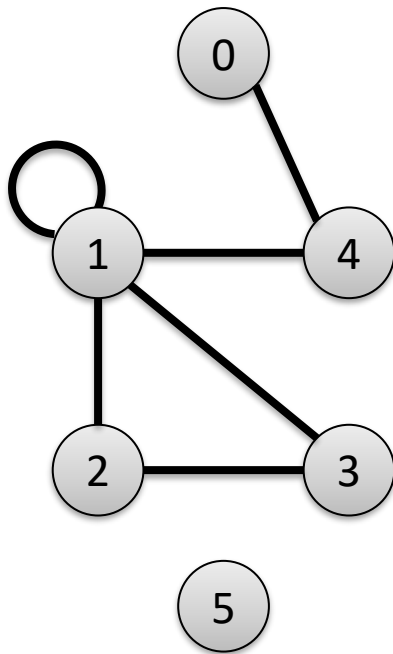
	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						



# Representações de grafos

- **Matriz de Adjacências**

– Exemplos:



	0	1	2	3	4	5
0	0	0	0	0	1	0
1	0	1	1	1	1	0
2	0	1	0	1	0	0
3	0	1	1	0	0	0
4	1	1	0	0	0	0
5	0	0	0	0	0	0

# Representações de grafos

- **Matriz de Adjacências**

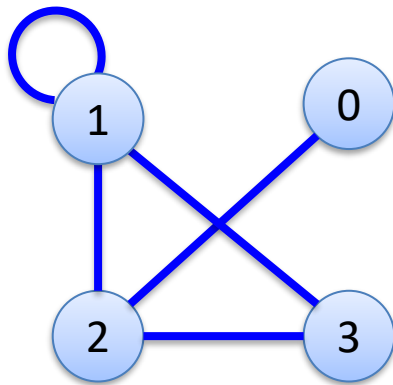
- Exemplos:

	0	1	2	3
0	0	0	1	0
1	0	1	1	1
2	1	1	0	1
3	0	1	1	0

# Representações de grafos

- **Matriz de Adjacências**

— Exemplos:

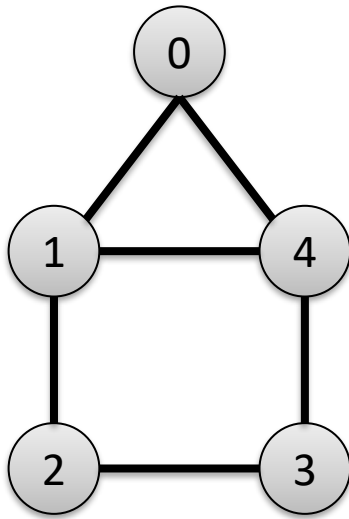


	0	1	2	3
0	0	0	1	0
1	0	1	1	1
2	1	1	0	1
3	0	1	1	0

# Representações de grafos

- **Matriz de Adjacências**

- Características em grafos simples não-direcionados:

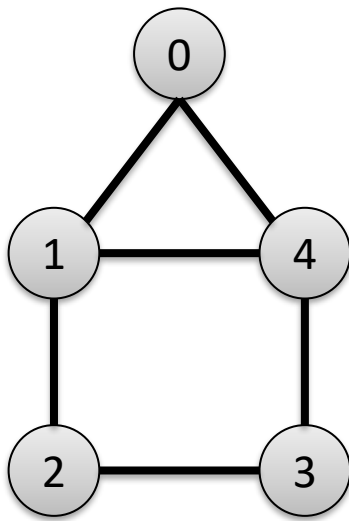


	0	1	2	3	4
0	0	1	0	0	1
1	1	0	1	0	1
2	0	1	0	1	0
3	0	0	1	0	1
4	1	1	0	1	0

# Representações de grafos

- **Matriz de Adjacências**

- Características em grafos simples não-direcionados:



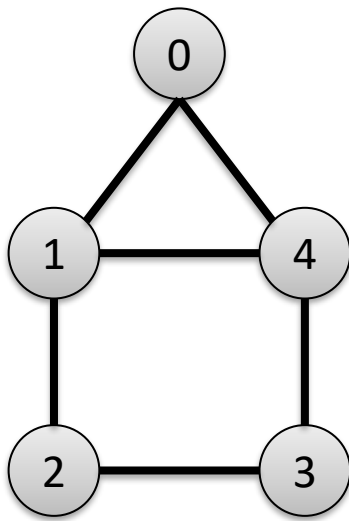
	0	1	2	3	4
0	0	1	0	0	1
1	1	0	1	0	1
2	0	1	0	1	0
3	0	0	1	0	1
4	1	1	0	1	0

- Se não tem laços, a diagonal é **nula**.

# Representações de grafos

- **Matriz de Adjacências**

- Características em grafos simples não-direcionados:



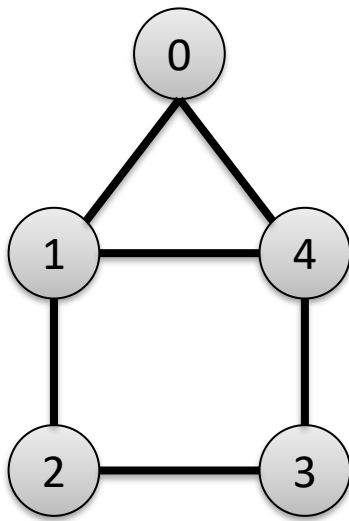
	0	1	2	3	4
0	0	1	0	0	1
1	1	0	1	0	1
2	0	1	0	1	0
3	0	0	1	0	1
4	1	1	0	1	0

- Se o grafo não é direcionado, a matriz é **simétrica**.

# Representações de grafos

- **Matriz de Adjacências**

- Características em grafos simples não-direcionados:

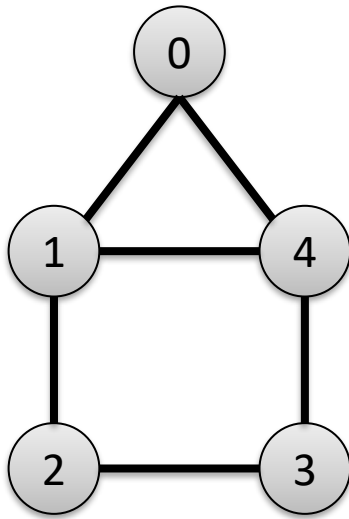


	0	1	2	3	4
0	0	1	0	0	1
1	1	0	1	0	1
2	0	1	0	1	0
3	0	0	1	0	1
4	1	1	0	1	0

# Representações de grafos

- **Matriz de Adjacências**

- Características em grafos simples não direcionados:



	0	1	2	3	4
0	0	1	0	0	1
1	1	0	1	0	1
2	0	1	0	1	0
3	0	0	1	0	1
4	1	1	0	1	0

$$1+0+1+0+1 = 3$$

**Grau do nó 1**

$$1+0+1+0+1 = 3$$

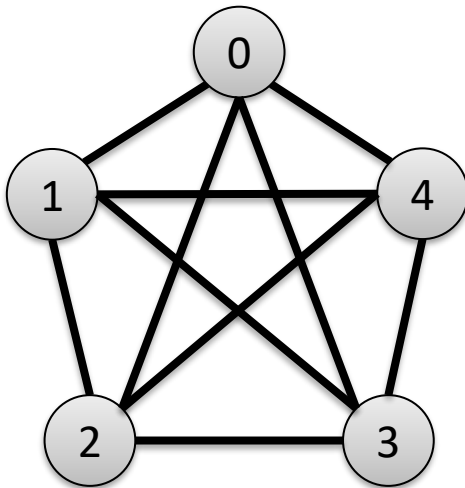
**Grau do nó 1**



# Representações de grafos

- **Matriz de Adjacências**

– Exemplos:

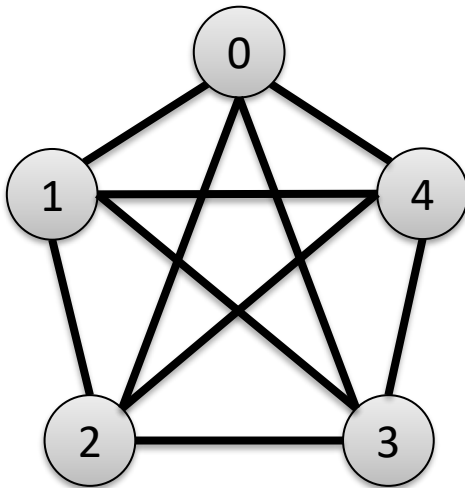


	0	1	2	3	4
0					
1					
2					
3					
4					

# Representações de grafos

- **Matriz de Adjacências**

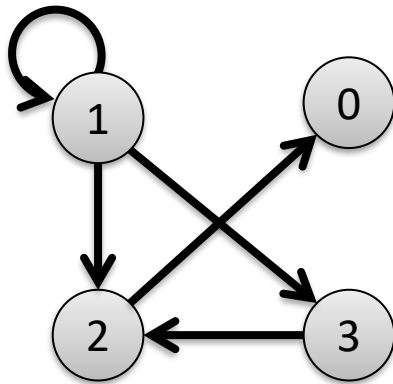
– Exemplos:



	0	1	2	3	4
0	0	1	1	1	1
1	1	0	1	1	1
2	1	1	0	1	1
3	1	1	1	0	1
4	1	1	1	1	0

# Representações de grafos

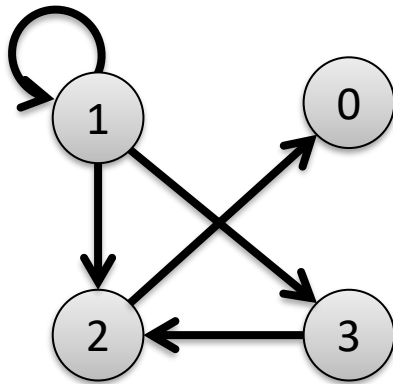
- **Matriz de Adjacências em grafos direcionados**
  - Segue o mesmo princípio, mas considerando as direções das arestas:



# Representações de grafos

- **Matriz de Adjacências em grafos direcionados**

- Segue o mesmo princípio, mas considerando as direções das arestas:

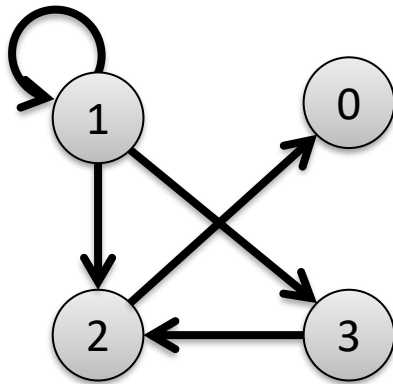


	0	1	2	3
0				
1				
2				
3				

# Representações de grafos

- **Matriz de Adjacências em grafos direcionados**

- Segue o mesmo princípio, mas considerando as direções das arestas:

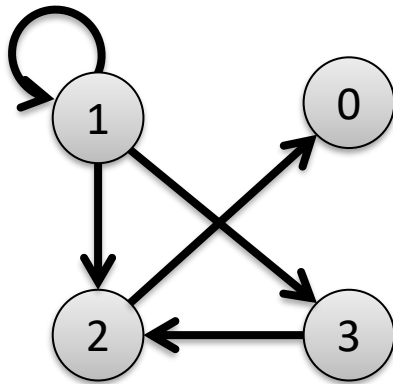


	0	1	2	3
0	0	0	0	0
1	0	1	1	1
2	1	0	0	0
3	0	0	1	0

# Representações de grafos

- **Matriz de Adjacências em grafos direcionados**

- Segue o mesmo princípio, mas considerando as direções das arestas:

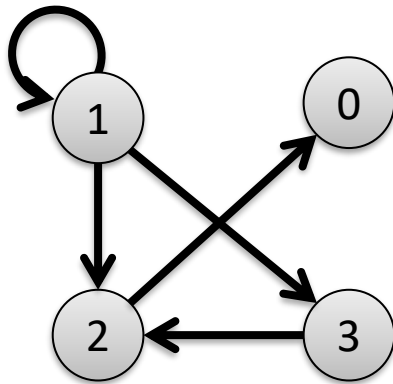


	0	1	2	3
0	0	0	0	0
1	0	1	1	1
2	1	0	0	0
3	0	0	1	0

# Representações de grafos

- **Matriz de Adjacências em grafos direcionados**

- Segue o mesmo princípio, mas considerando as direções das arestas:



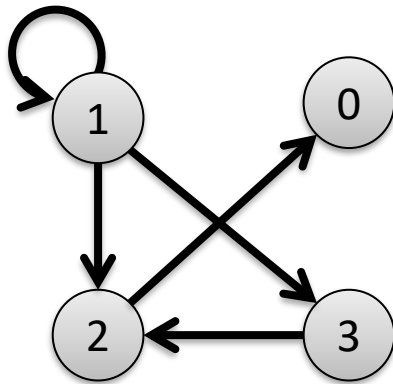
	0	1	2	3
0	0	0	0	0
1	0	1	1	1
2	1	0	0	0
3	0	0	1	0

$0+1+1+1 = 3$   
**Grau de saída**  
do nó 1

# Representações de grafos

- **Matriz de Adjacências em grafos direcionados**

- Segue o mesmo princípio, mas considerando as direções das arestas:



	0	1	2	3
0	0	0	0	0
1	0	1	1	1
2	1	0	0	0
3	0	0	1	0

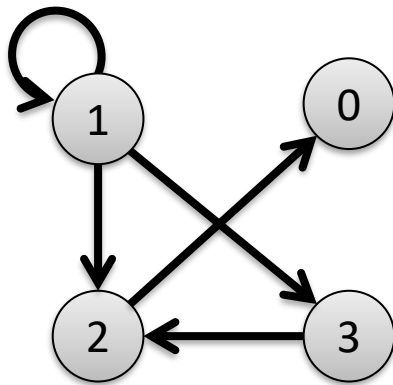
$0+1+1+1 = 3$   
**Grau de saída**  
do nó 1



# Representações de grafos

- **Matriz de Adjacências em grafos direcionados**

- Segue o mesmo princípio, mas considerando as direções das arestas:



	0	1	2	3
0	0	0	0	0
1	0	1	1	1
2	1	0	0	0
3	0	0	1	0

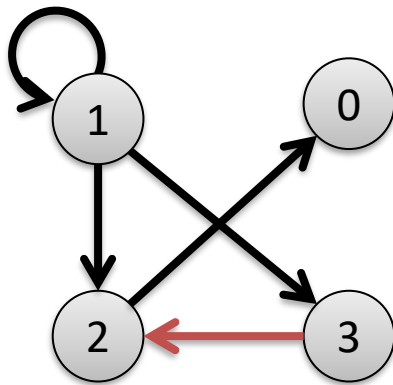
$0+1+1+1 = 3$   
**Grau de saída**  
do nó 1

$0+1+0+0 = 1$   
**Grau de entrada**  
do nó 1

# Representações de grafos

- **Matriz de Adjacências em grafos direcionados**

- Segue o mesmo princípio, mas considerando as direções das arestas:



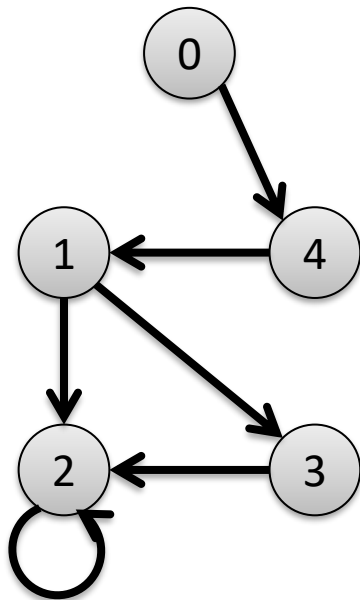
	0	1	2	3
0	0	0	0	0
1	0	1	1	1
2	1	0	0	0
3	0	0	1	0

Indica que existe um **caminho** de **comprimento 1** com **origem** no **nó 3** e **destino** no **nó 2**.

# Representações de grafos

- **Matriz de Adjacências em grafos direcionados**

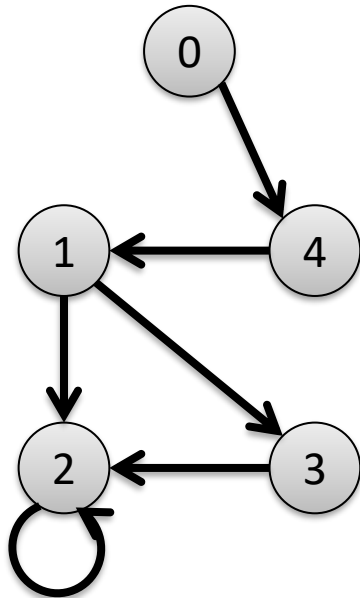
– Exemplo:



# Representações de grafos

- **Matriz de Adjacências em grafos direcionados**

– Exemplo:

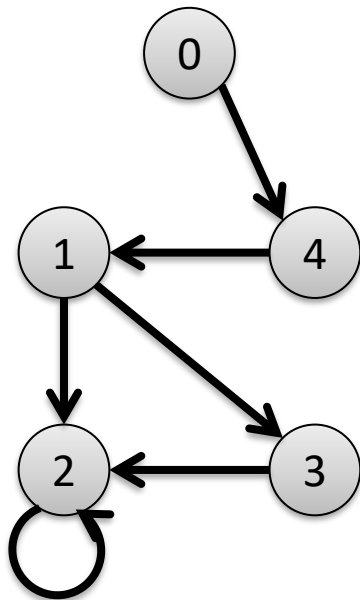


	0	1	2	3	4
0					
1					
2					
3					
4					

# Representações de grafos

- **Matriz de Adjacências em grafos direcionados**

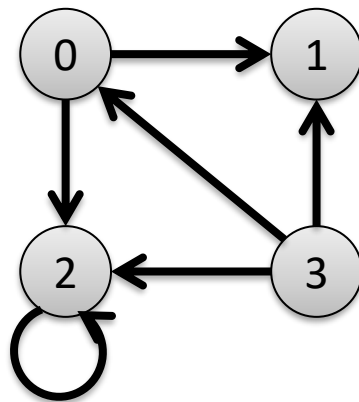
– Exemplo:



	0	1	2	3	4
0	0	0	0	0	1
1	0	0	1	1	0
2	0	0	1	0	0
3	0	0	1	0	0
4	0	1	0	0	0

# Representações de grafos

- **Matriz de Adjacências em grafos direcionados**
  - Sabemos que a matriz de adjacência indica se **existe** um **caminho de comprimento 1** entre dois nós.
  - Como saber se existe um caminho de **comprimento 2**?

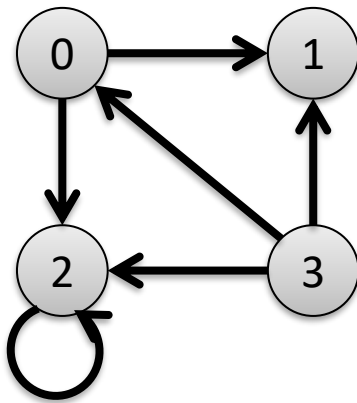


	0	1	2	3
0	0	1	1	0
1	0	0	0	0
2	0	0	1	0
3	1	1	1	0

# Representações de grafos

- **Matriz de Adjacências em grafos direcionados**

- **Solução:** realizar o **produto booleano** entre a matriz e ela mesma.



Matriz A:

$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

booleano

- $A \times A = A^{(2)} \rightarrow$  Matriz indica se existe um caminho de **comprimento 2** entre dois nós.

# Representações de grafos

- **Matriz de Adjacências em grafos direcionados**

- $A \times A = A^{(2)} \rightarrow$  Matriz indica se existe um caminho de **comprimento 2** entre dois nós.

- Lembrando:

X	Y	X && Y
0	0	0
0	1	0
1	0	0
1	1	1

X	Y	X    Y
0	0	0
0	1	1
1	0	1
1	1	1

- Assim:

$$A \times A = A^{(2)} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$



# Representações de grafos

- **Matriz de Adjacências em grafos direcionados**

$$A \times A = A^{(2)} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

# Representações de grafos

- Matriz de Adjacências em grafos direcionados

$$A \times A = A^{(2)} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

$$A_{(4,2)}^{(2)} = (A_{(4,1)} \&\& A_{(1,1)}) \parallel (A_{(4,2)} \&\& A_{(2,2)}) \parallel (A_{(4,3)} \&\& A_{(3,2)}) \parallel (A_{(4,4)} \&\& A_{(4,2)})$$

$$A_{(4,2)}^{(2)} = (1 \&\& 1) \parallel (1 \&\& 0) \parallel (1 \&\& 0) \parallel (0 \&\& 1)$$

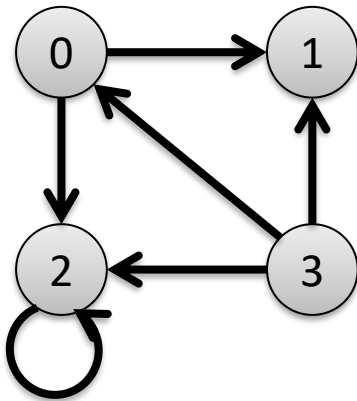
$$A_{(4,2)}^{(2)} = 1 \parallel 0 \parallel 0 \parallel 0$$

$$A_{(4,2)}^{(2)} = 1$$

# Representações de grafos

- **Matriz de Adjacências em grafos direcionados**

– Verificando:



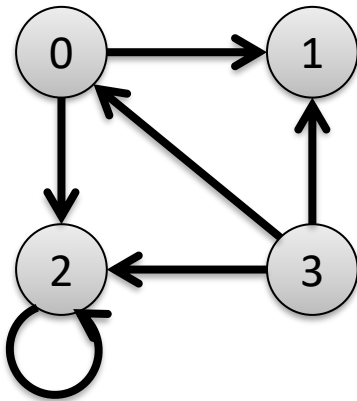
Matriz  $A^{(2)}$ :

	0	1	2	3
0	0	0	1	0
1	0	0	0	0
2	0	0	1	0
3	0	1	1	0

# Representações de grafos

- **Matriz de Adjacências em grafos direcionados**

- Verificando:



Matriz  $A^{(2)}$ :

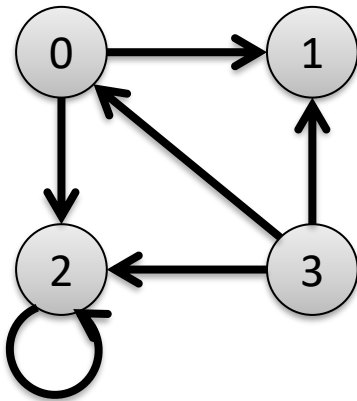
	0	1	2	3
0	0	0	1	0
1	0	0	0	0
2	0	0	1	0
3	0	1	1	0

- Existe caminho de comprimento 2 do nó 0 para o nó 1?

# Representações de grafos

- **Matriz de Adjacências em grafos direcionados**

- Verificando:



Matriz  $A^{(2)}$ :

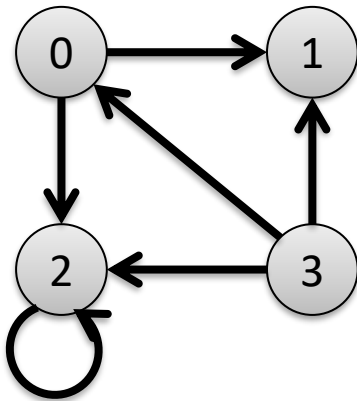
	0	1	2	3
0	0	0	1	0
1	0	0	0	0
2	0	0	1	0
3	0	1	1	0

- Existe caminho de comprimento 2 do nó 0 para o nó 1? **Não!**

# Representações de grafos

- **Matriz de Adjacências em grafos direcionados**

- Verificando:



Matriz  $A^{(2)}$ :

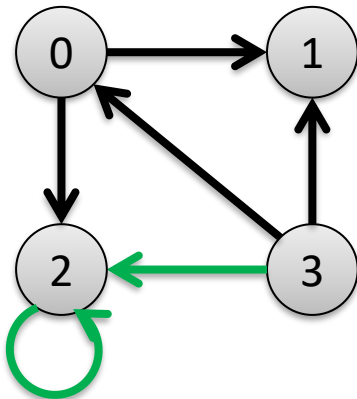
	0	1	2	3
0	0	0	1	0
1	0	0	0	0
2	0	0	1	0
3	0	1	1	0

- Existe caminho de comprimento 2 do nó 0 para o nó 1? **Não!**
  - Existe caminho de comprimento 2 do nó 3 ao nó 2?

# Representações de grafos

- **Matriz de Adjacências em grafos direcionados**

- Verificando:



Matriz  $A^{(2)}$ :

	0	1	2	3
0	0	0	1	0
1	0	0	0	0
2	0	0	1	0
3	0	1	1	0

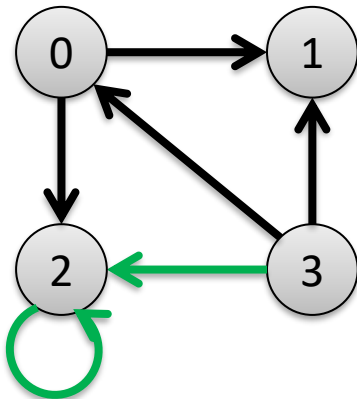
- Existe caminho de comprimento 2 do nó 0 para o nó 1? **Não!**
  - Existe caminho de comprimento 2 do nó 3 ao nó 2? **Sim!**

**Obs:** Como verificar se existe caminho de **comprimento 3**?

# Representações de grafos

- **Matriz de Adjacências em grafos direcionados**

- Verificando:



Matriz  $A^{(2)}$ :

	0	1	2	3
0	0	0	1	0
1	0	0	0	0
2	0	0	1	0
3	0	1	1	0

- Existe caminho de comprimento 2 do nó 0 para o nó 1? **Não!**
  - Existe caminho de comprimento 2 do nó 3 ao nó 2? **Sim!**

**Obs:** Para verificar se existe caminho de **comprimento 3**, basta calcular  $A^{(3)}$ , e assim por diante.



# Representações de grafos

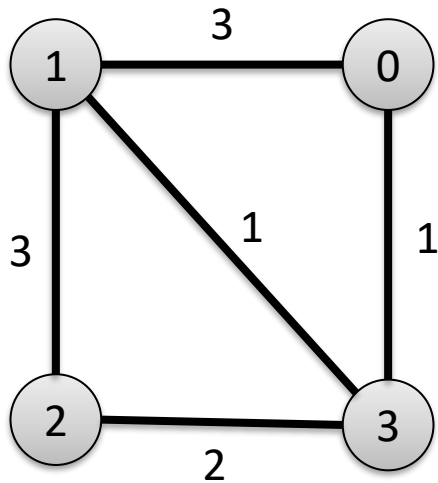
---

- **Matriz de Adjacências em grafos ponderados**
  - Quando o grafo é **ponderado**, a representação só fica **completa** quando também se indica a sua **matriz de pesos**.
  - A **matriz de pesos** é construída de maneira **semelhante** à matriz de adjacência, mas colocando os **pesos** nos locais de **arestas**.
  - Se **não existem arestas** ligando os nós, o peso que é colocado na posição correspondente é igual a **infinito** ( $\infty$ ).

# Representações de grafos

- **Matriz de Adjacências em grafos ponderados**

— Exemplo:



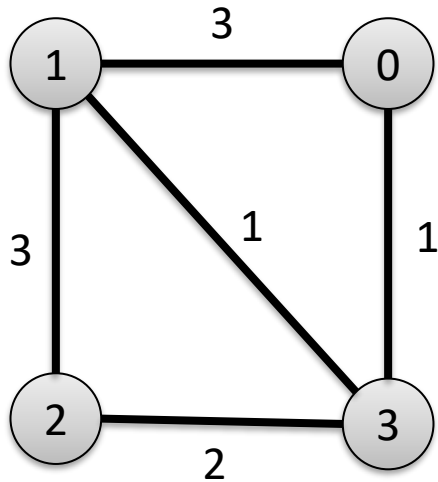
Matriz de pesos:

	0	1	2	3
0				
1				
2				
3				

# Representações de grafos

- **Matriz de Adjacências em grafos ponderados**

— Exemplo:



Matriz de pesos:

	0	1	2	3
0	$\infty$	3	$\infty$	1
1	3	$\infty$	3	1
2	$\infty$	3	$\infty$	2
3	1	1	2	$\infty$

# Representações de grafos

- **Matriz de Incidências**

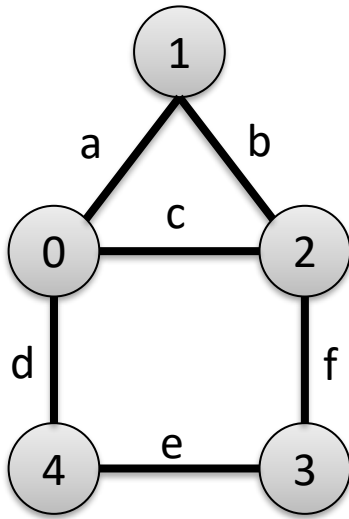
- Representa um grafo através de uma **matriz bidimensional**, onde uma das dimensões são **vértices** e a outra dimensão são **arestas**.
- Seja um grafo  $G=(V,E)$  e uma matriz  $B_{|V| \times |E|} = [b_{ij}]$ , onde  $n$  é o número de nós do grafo:
  - $B(i, j) = 1$ , se  $v_i$  pertence a  $e_j$
  - $B(i, j) = 0$ , caso contrário

**Obs:** Um laço é representado pelo valor 2.

# Representações de grafos

- **Matriz de Incidências**

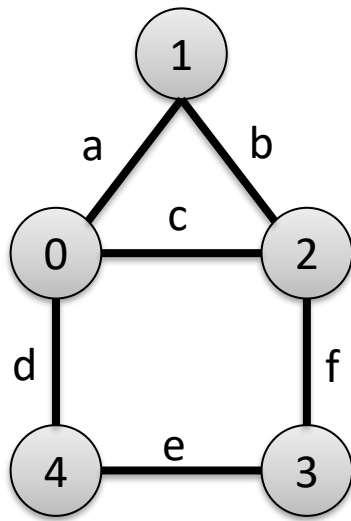
— Exemplo:



# Representações de grafos

- **Matriz de Incidências**

— Exemplo:



Nós



	a	b	c	d	e	f
0						
1						
2						
3						
4						

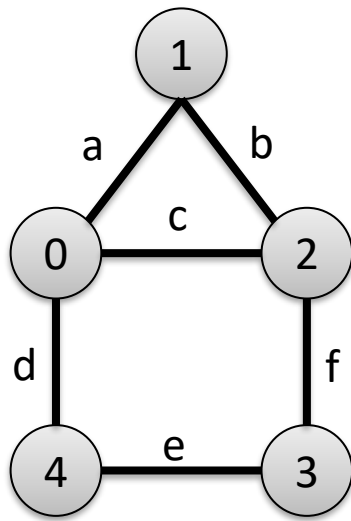
Arestas



# Representações de grafos

- **Matriz de Incidências**

— Exemplo:



Nós



	a	b	c	d	e	f
0	1	0	1	1	0	0
1	1	1	0	0	0	0
2	0	1	1	0	0	1
3	0	0	0	0	1	1
4	0	0	0	1	1	0

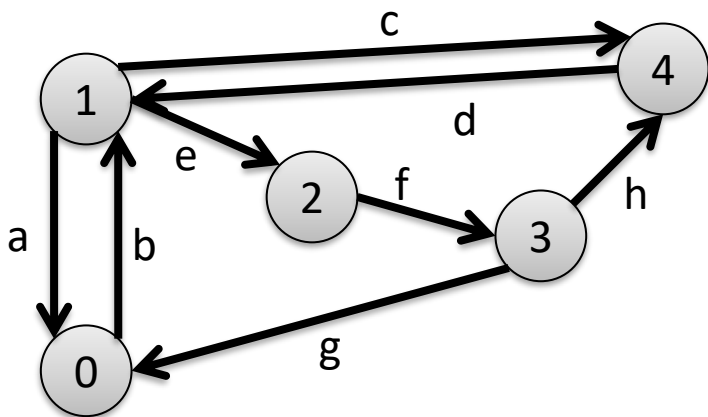
Arestas



# Representações de grafos

- **Matriz de Incidências em grafos direcionados**

- Se  $G$  é um **grafo direcionado**, então  $b_{ij} = +1$  se a aresta **chega** em  $v_i$  e  $b_{ij} = -1$ , caso a aresta **parta** de  $v_i$ .
- Exemplo:



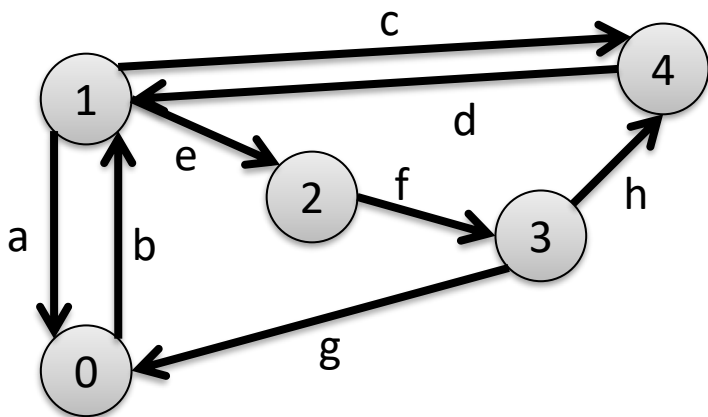
**Obs:** Um laço é representado pelo valor 1 ou -1.



# Representações de grafos

- **Matriz de Incidências em grafos direcionados**

- Se  $G$  é um **grafo direcionado**, então  $b_{ij} = +1$  se a aresta **chega** em  $v_i$  e  $b_{ij} = -1$ , caso a aresta **parta** de  $v_i$ .
- Exemplo:

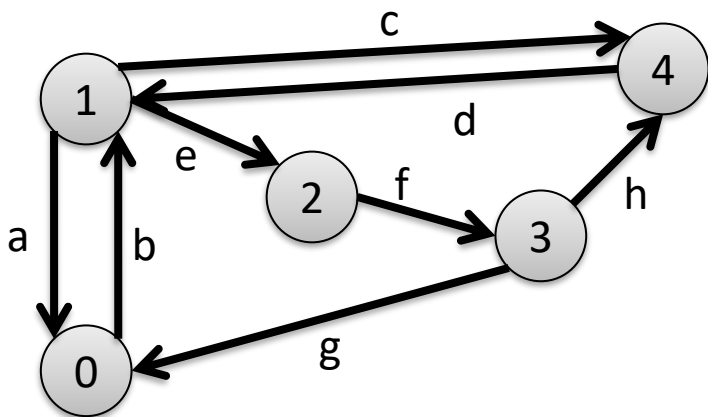


	a	b	c	d	e	f	g	h
0								
1								
2								
3								
4								

# Representações de grafos

- **Matriz de Incidências em grafos direcionados**

- Se  $G$  é um **grafo direcionado**, então  $b_{ij} = +1$  se a aresta **chega** em  $v_i$  e  $b_{ij} = -1$ , caso a aresta **parta** de  $v_i$ .
- Exemplo:



	a	b	c	d	e	f	g	h
0	+1	-1	0	0	0	0	+1	0
1	-1	+1	-1	+1	-1	0	0	0
2	0	0	0	0	+1	-1	0	0
3	0	0	0	0	0	+1	-1	-1
4	0	0	+1	-1	0	0	0	+1

# Representações de grafos

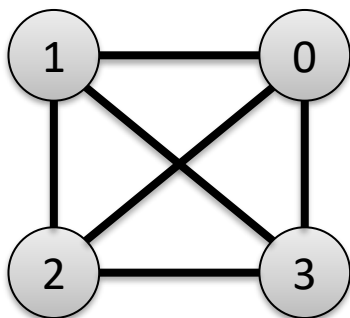
---

- **Lista de adjacências**
  - Para **cada nó** é criada uma **lista encadeada** com todos os **nós adjacentes**.
  - Em geral é a mais usada em **grafos esparsos**, por ser mais eficiente.
  - A única forma de descobrir as adjacências é **percorrendo todos** os elementos da lista.
  - Em grafos direcionados, considerar os **sentidos** das setas.

# Representações de grafos

- **Lista de adjacências**

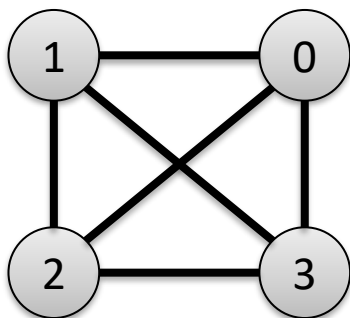
- Exemplo:



# Representações de grafos

- **Lista de adjacências**

— Exemplo:

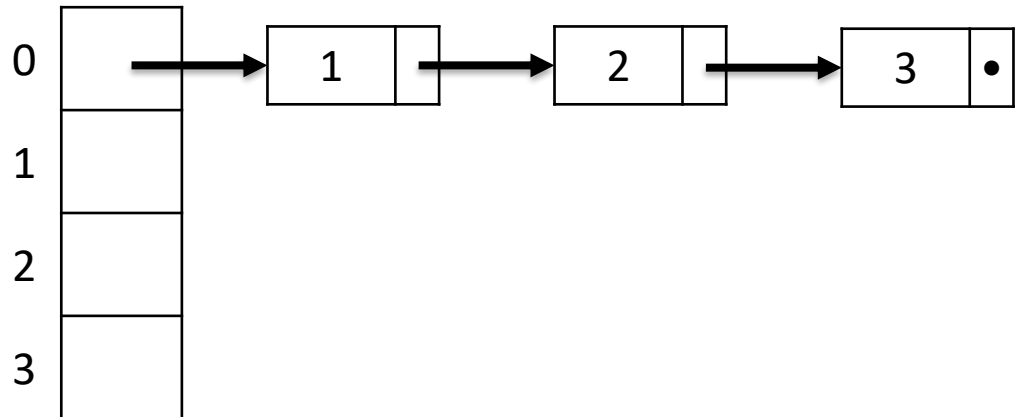
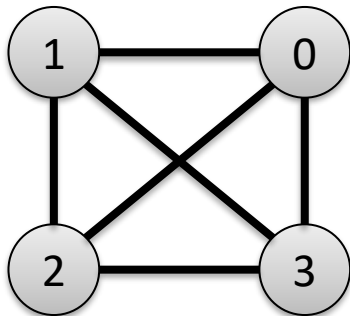


0	
1	
2	
3	

# Representações de grafos

- **Lista de adjacências**

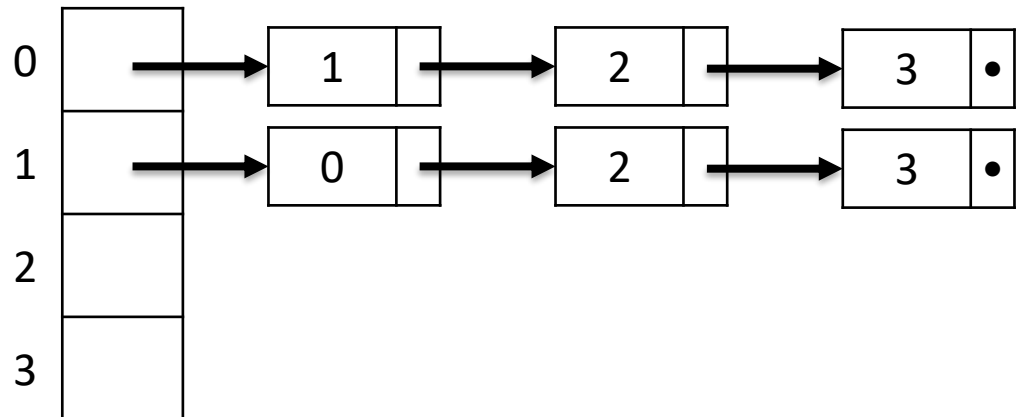
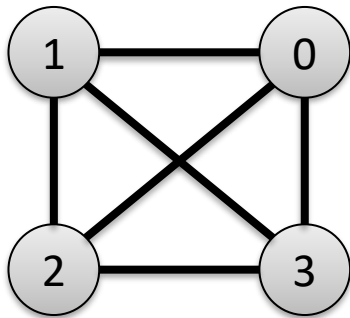
– Exemplo:



# Representações de grafos

- **Lista de adjacências**

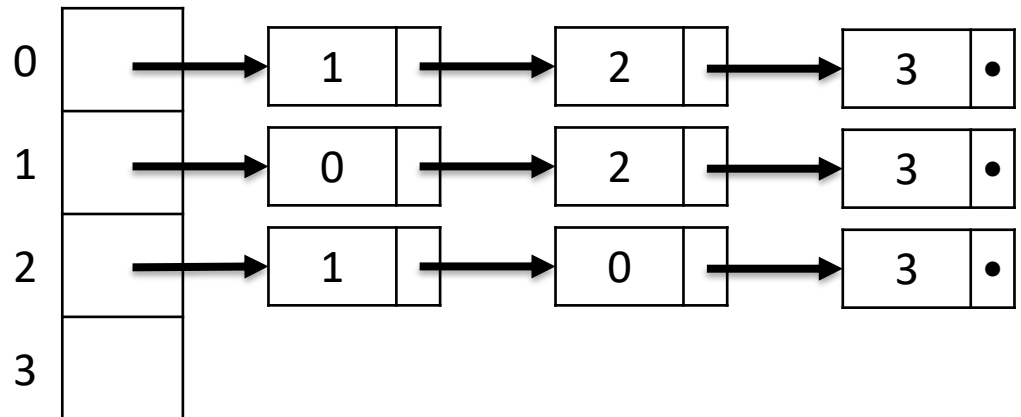
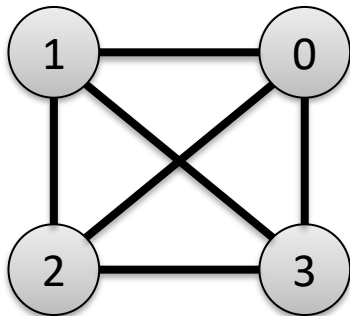
— Exemplo:



# Representações de grafos

- **Lista de adjacências**

– Exemplo:

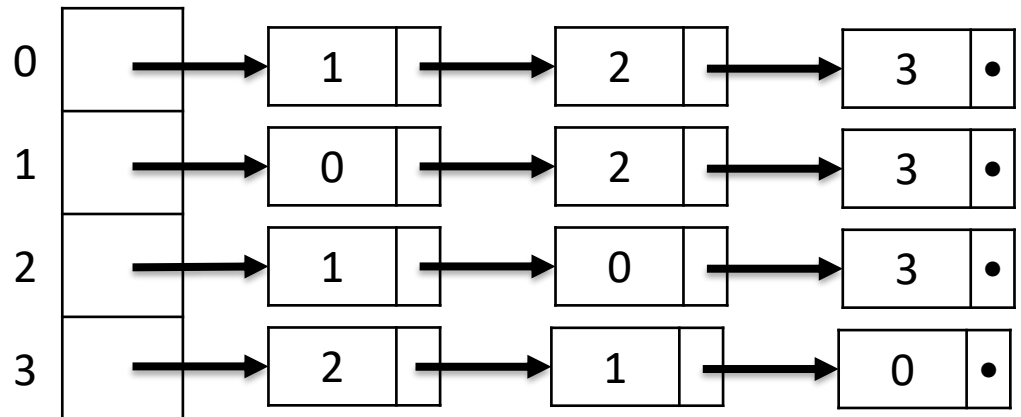
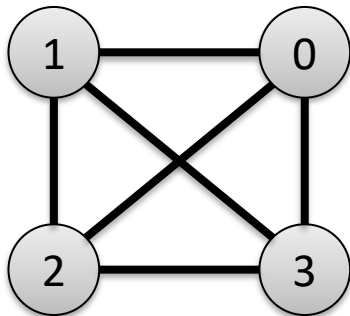




# Representações de grafos

- **Lista de adjacências**

– Exemplo:



# Representações de grafos

- **Lista de adjacências**

- Exemplo:



# Representações de grafos

- **Lista de adjacências**

— Exemplo:

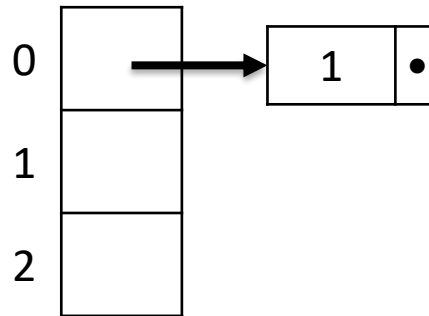


0	
1	
2	

# Representações de grafos

- **Lista de adjacências**

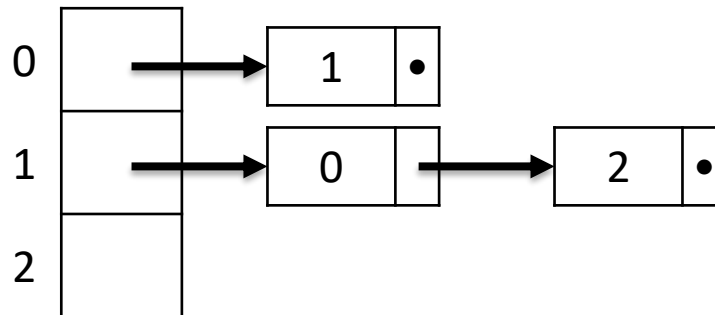
– Exemplo:



# Representações de grafos

- **Lista de adjacências**

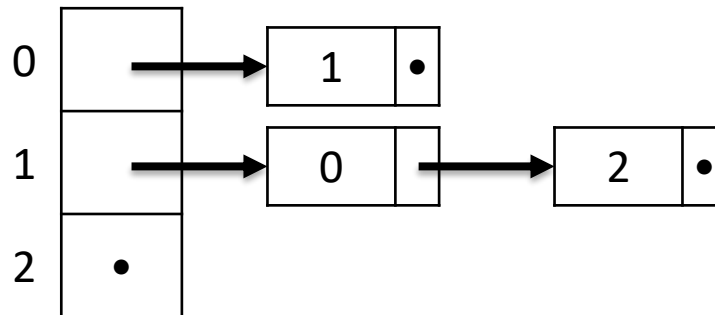
— Exemplo:



# Representações de grafos

- **Lista de adjacências**

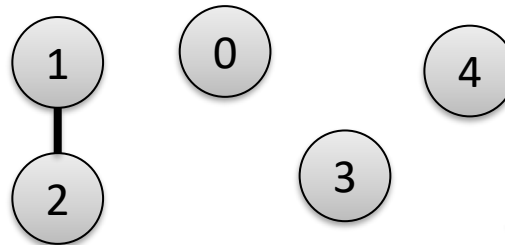
– Exemplo:



# Representações de grafos

- **Lista de adjacências**

– Exemplo:



Matriz de adjacências



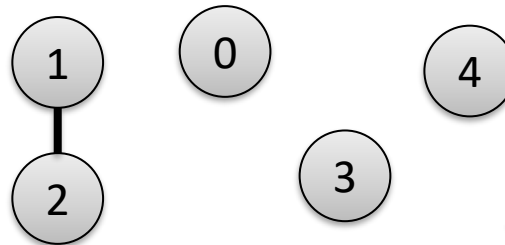
Lista de adjacências



# Representações de grafos

- **Lista de adjacências**

– Exemplo:



Matriz de adjacências

Lista de adjacências

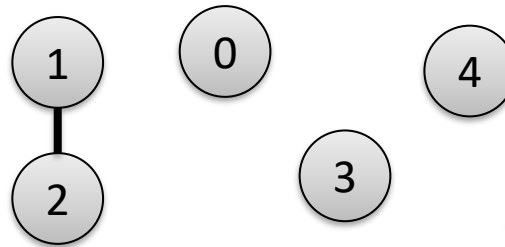
	0	1	2	3	4
0					
1					
2					
3					
4					



# Representações de grafos

- **Lista de adjacências**

– Exemplo:



Matriz de adjacências

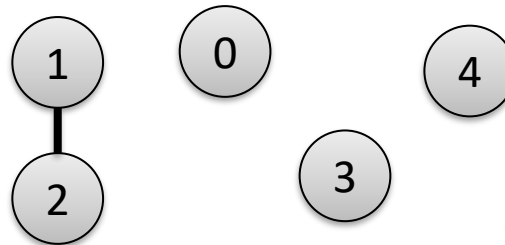
Lista de adjacências

	0	1	2	3	4
0	0	0	0	0	0
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

# Representações de grafos

- **Lista de adjacências**

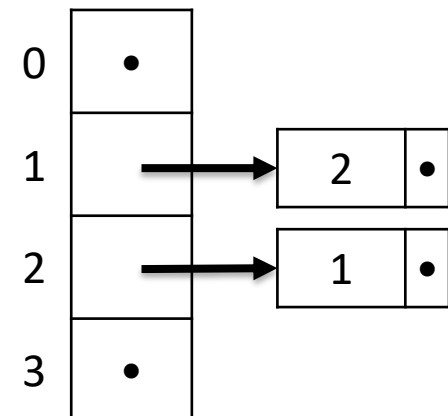
– Exemplo:



Matriz de adjacências

	0	1	2	3	4
0	0	0	0	0	0
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

Lista de adjacências

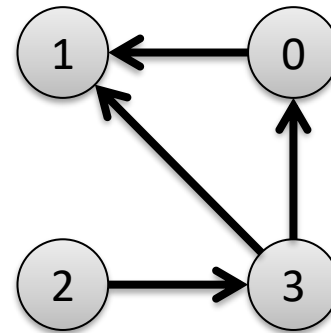


# Representações de grafos

- **Implementação de grafos**

- Matriz de adjacência:

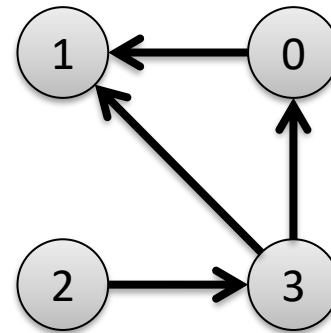
	0	1	2	3
0	0	1	0	0
1	0	0	0	0
2	0	0	0	1
3	1	1	0	0



# Representações de grafos

- Implementação de grafos
  - Matriz de adjacência:

	0	1	2	3
0	0	1	0	0
1	0	0	0	0
2	0	0	0	1
3	1	1	0	0



```
// Matriz de Adjacências
int G[MAX_NOS][MAX_NOS];

inicializaGrafo(G);

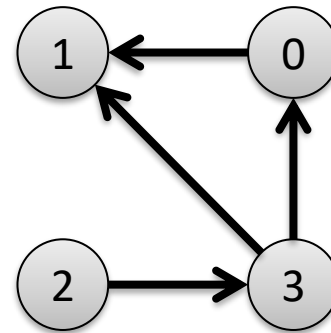
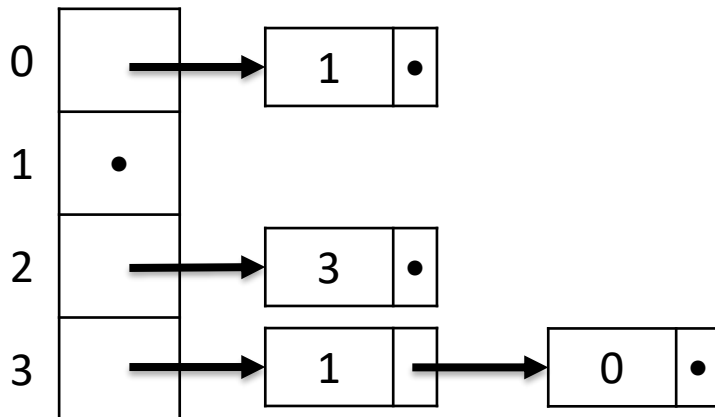
// Marcando adjacências
G[0][1] = 1;
G[2][3] = 1;
G[3][1] = 1;
G[3][0] = 1;

imprimeMatriz(G);
```

# Representações de grafos

- **Implementação de grafos**

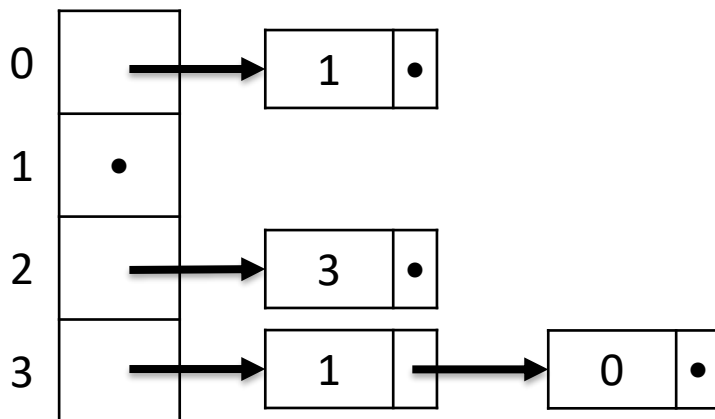
- Lista de adjacência:



# Representações de grafos

- Implementação de grafos

- Lista de adjacência:



```
typedef struct NODE{  
    int x;  
    struct NODE *next;  
} NODE;  
  
typedef NODE* LISTNODEPTR;
```

```
// Grafo é um vetor de listas encadeadas  
LISTNODEPTR G[MAX_NOS];  
  
inicializaGrafo(G);  
  
// Colocando adjacências  
G[0] = criaNo(1);  
G[2] = criaNo(3);  
G[3] = criaNo(1);  
insereNoFim(G[3], 0);  
  
// Imprimindo grafo  
imprimeGrafo(G);  
  
// Libera a memória  
liberaGrafo(G);
```

# Percursos em grafos

---

# Percursos em grafos

---

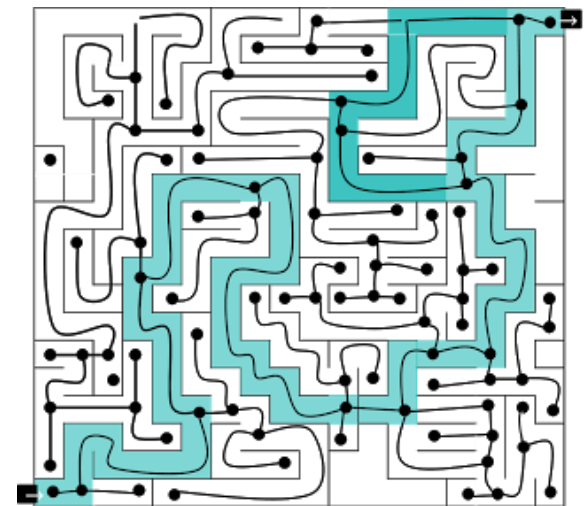
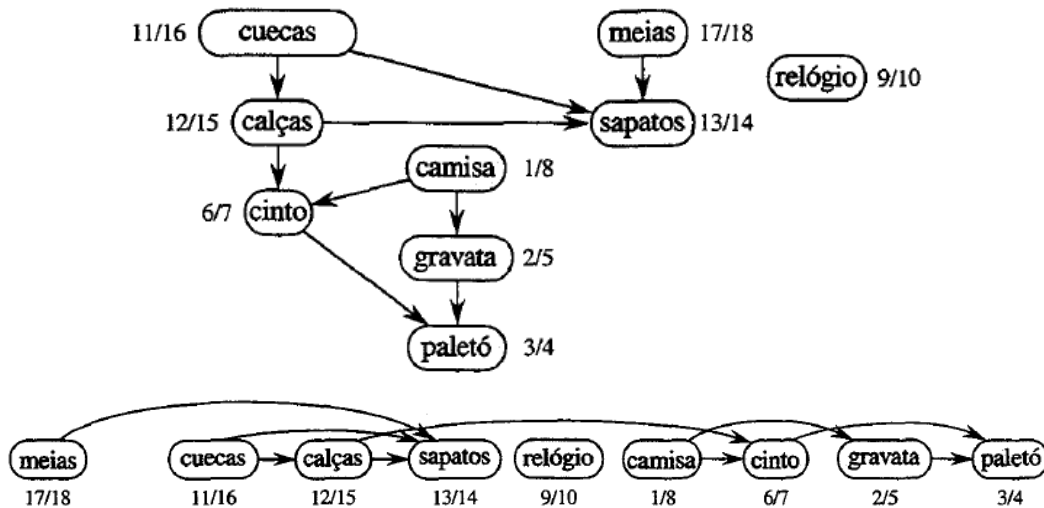
- **Definição**
  - **Algoritmos de percurso** (ou busca) em grafos tem o **objetivo** de **percorrer** e **explorar** todos os **nós acessíveis** do grafo, partindo de um **nó inicial**.
  - Cada vértice examinado deve ser marcado como visitado.
  - Inicialmente todos estão como não visitados.
  - Existem dois algoritmos **principais** de percurso:
    - **Em profundidade** (depth-first search)
    - **Em largura** (breadth-first search)



# Percursos em grafos

- Busca em profundidade

- O algoritmo é base para a resolução de muitos problemas, como a **ordenação topológica**, identificação de **componentes fortemente conectados**, labirintos e **quebra-cabeças**.



# Percursos em grafos

- **Busca em profundidade**

- Passo-a-passo:

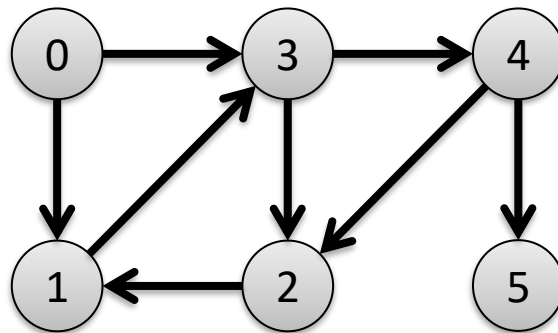
1. Escolha um vértice inicial **v não visitado**.
2. Visite e marque **todos os vértices acessíveis** a partir de **v**, que ainda **não estejam marcados**, percorrendo um **caminho único o mais profundo** possível.
3. Quando o último nó do caminho não tiver mais sucessores não visitados, visitar os **caminhos laterais não visitados**.

**Convenção:** Sempre que se esgotarem os caminhos laterais, o próximo vértice a ser visitado será o de **menor índice**.

# Percursos em grafos

- **Busca em profundidade**

— Exemplo:



# Percursos em grafos

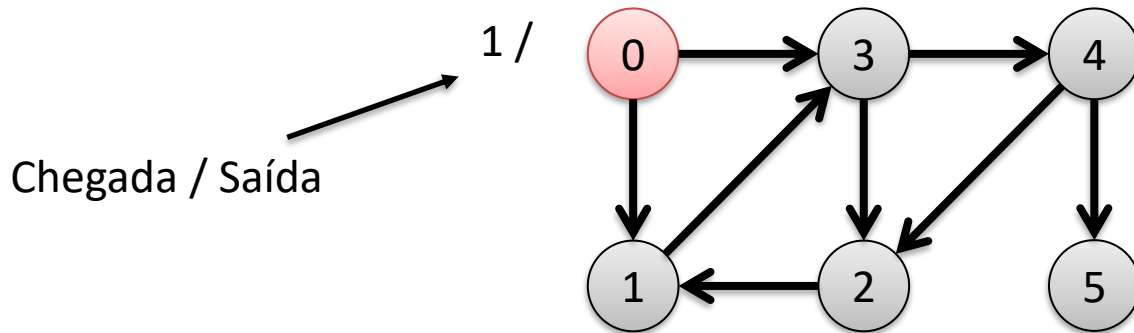
- **Busca em profundidade**

Nó inicial:



→ Marcado como visitado

— Exemplo:



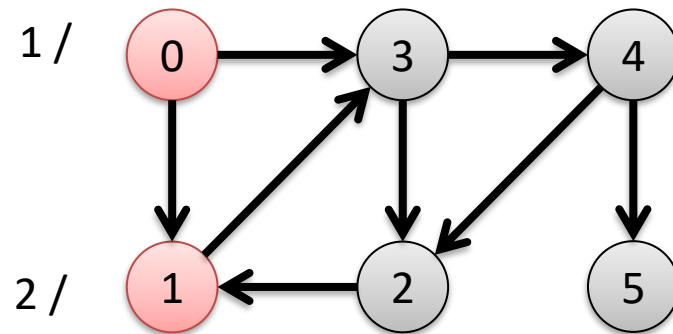
Ordem da busca em profundidade: 0

# Percursos em grafos

- **Busca em profundidade**

Nó inicial: 0

— Exemplo:



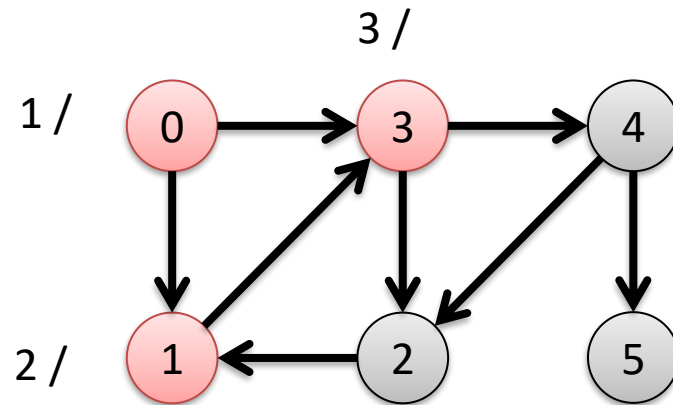
Ordem da busca em profundidade: 0, 1

# Percursos em grafos

- **Busca em profundidade**

Nó inicial: 0

— Exemplo:



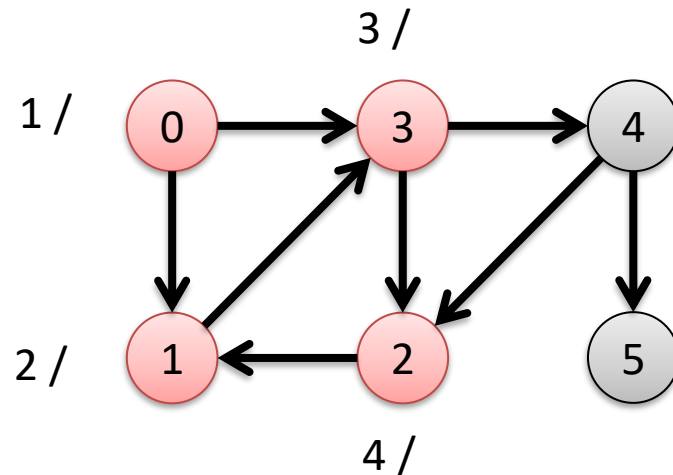
Ordem da busca em profundidade: 0, 1, 3

# Percursos em grafos

- **Busca em profundidade**

Nó inicial: 0

— Exemplo:



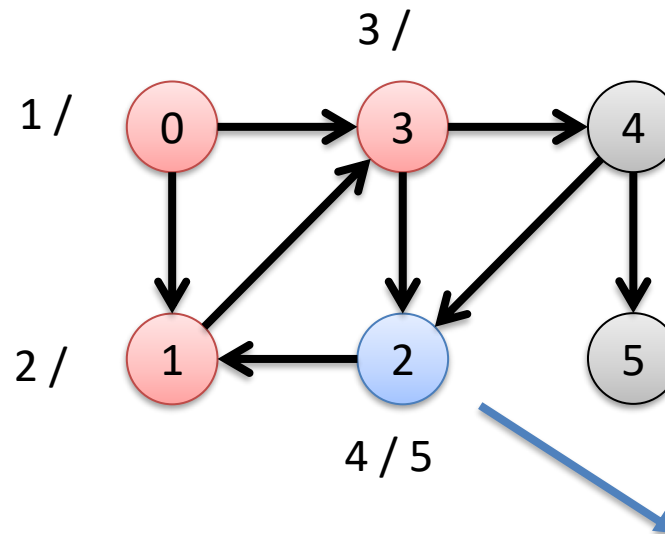
Ordem da busca em profundidade: 0, 1, 3, 2

# Percursos em grafos

- **Busca em profundidade**

Nó inicial: 0

— Exemplo:



Terminou de visitar todos os nós acessíveis por 2.

Ordem da busca em profundidade: 0, 1, 3, 2

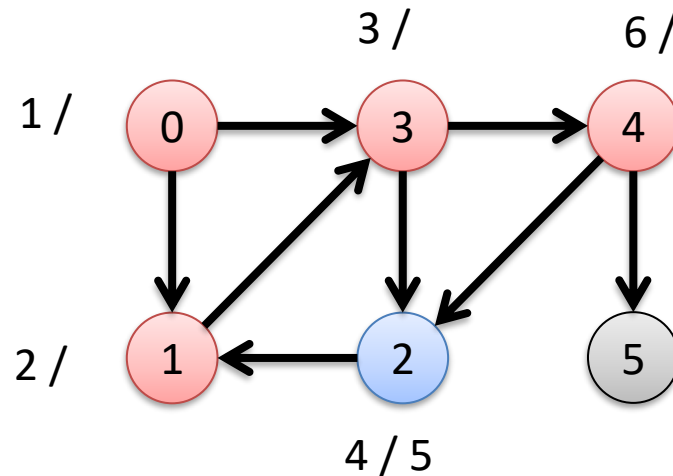


# Percursos em grafos

- **Busca em profundidade**

Nó inicial: 0

— Exemplo:



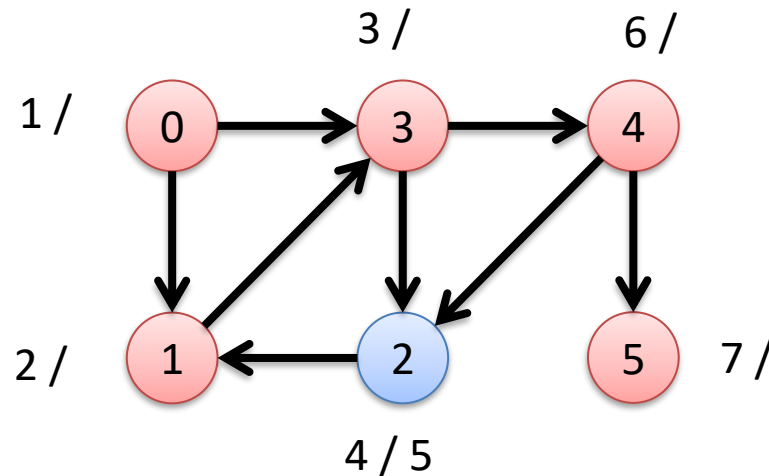
Ordem da busca em profundidade: 0, 1, 3, 2, 4

# Percursos em grafos

- **Busca em profundidade**

Nó inicial: 0

— Exemplo:



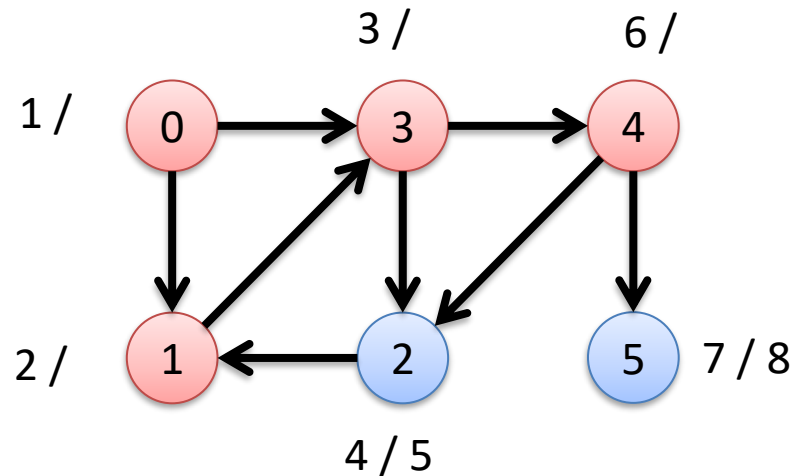
Ordem da busca em profundidade: 0, 1, 3, 2, 4, 5

# Percursos em grafos

- **Busca em profundidade**

Nó inicial: 0

— Exemplo:



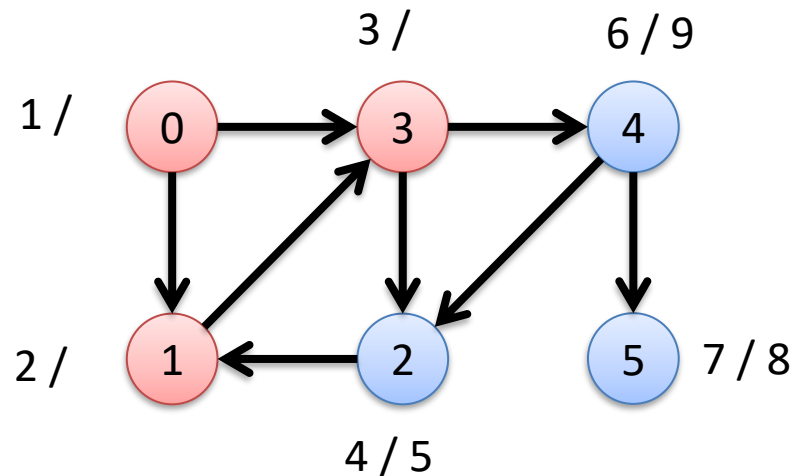
Ordem da busca em profundidade: 0, 1, 3, 2, 4, 5

# Percursos em grafos

- **Busca em profundidade**

Nó inicial: 0

— Exemplo:



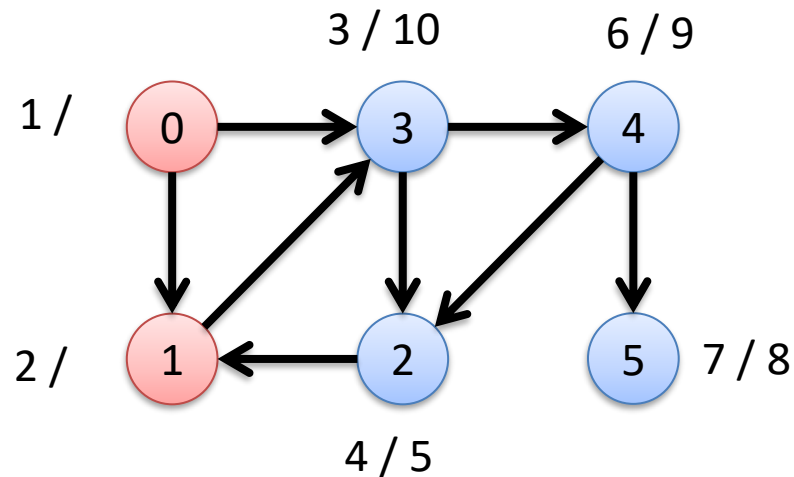
Ordem da busca em profundidade: 0, 1, 3, 2, 4, 5

# Percursos em grafos

- Busca em profundidade

Nó inicial: 0

— Exemplo:



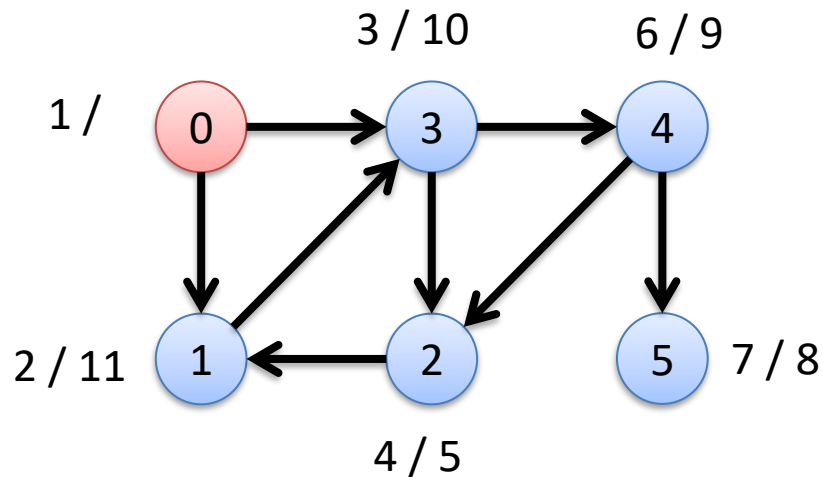
Ordem da busca em profundidade: 0, 1, 3, 2, 4, 5

# Percursos em grafos

- Busca em profundidade

Nó inicial: 0

— Exemplo:



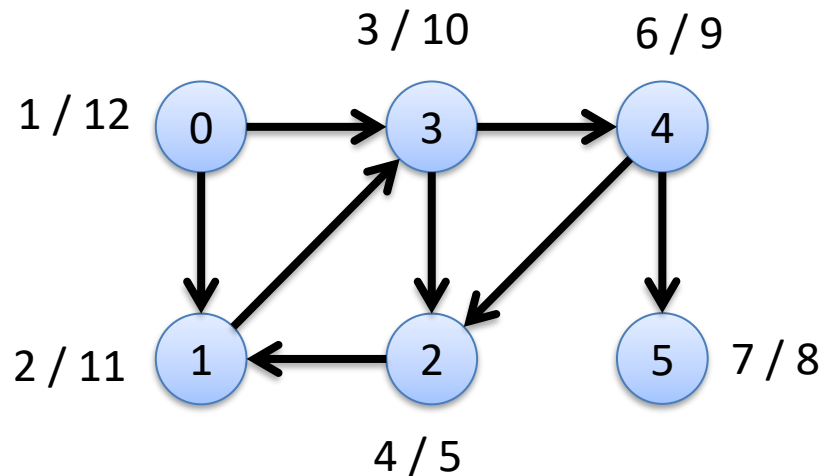
Ordem da busca em profundidade: 0, 1, 3, 2, 4, 5

# Percursos em grafos

- Busca em profundidade

Nó inicial: 0

— Exemplo:

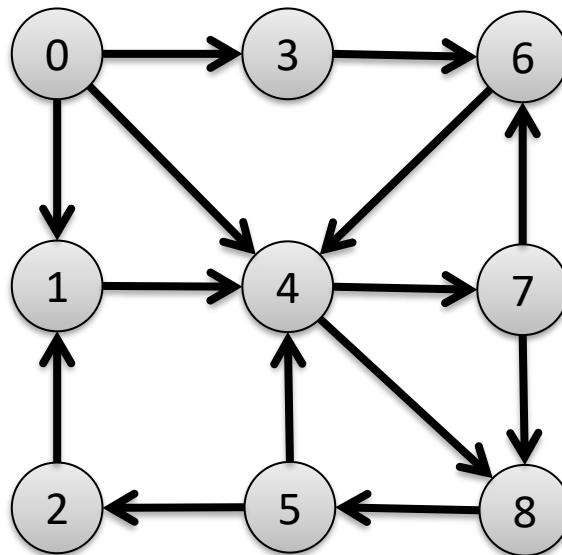


Ordem da busca em profundidade: 0, 1, 3, 2, 4, 5

# Percursos em grafos

- **Busca em profundidade**

- Exercício:

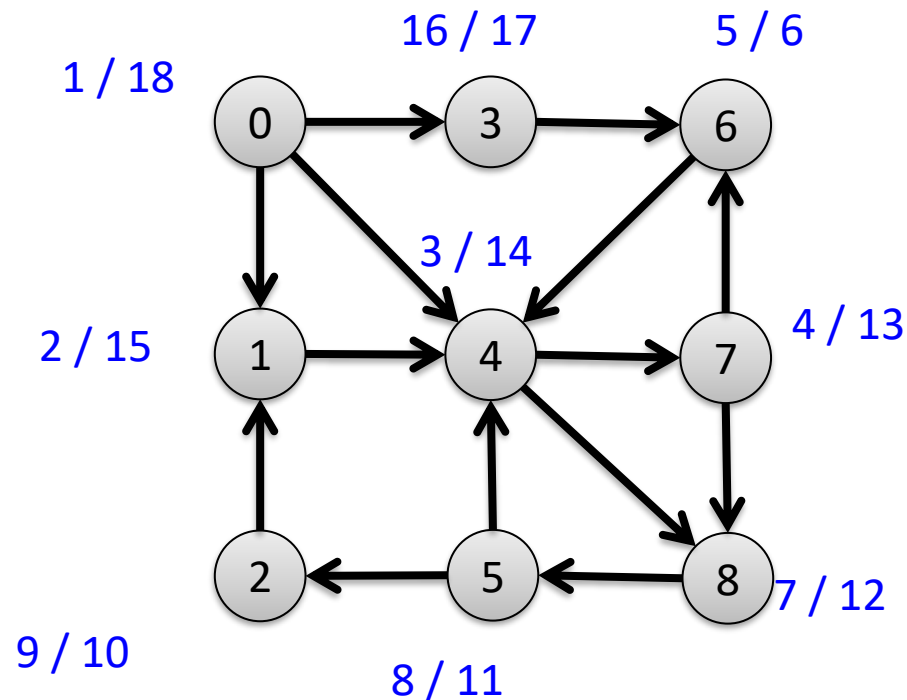




# Percursos em grafos

- Busca em profundidade

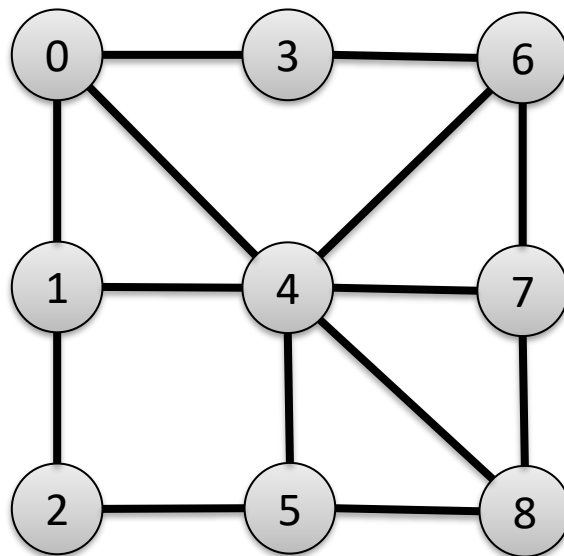
— Exercício:      Ordem da busca em profundidade: 0, 1, 4, 7, 6, 8, 5, 2, 3



# Percursos em grafos

- **Busca em profundidade**

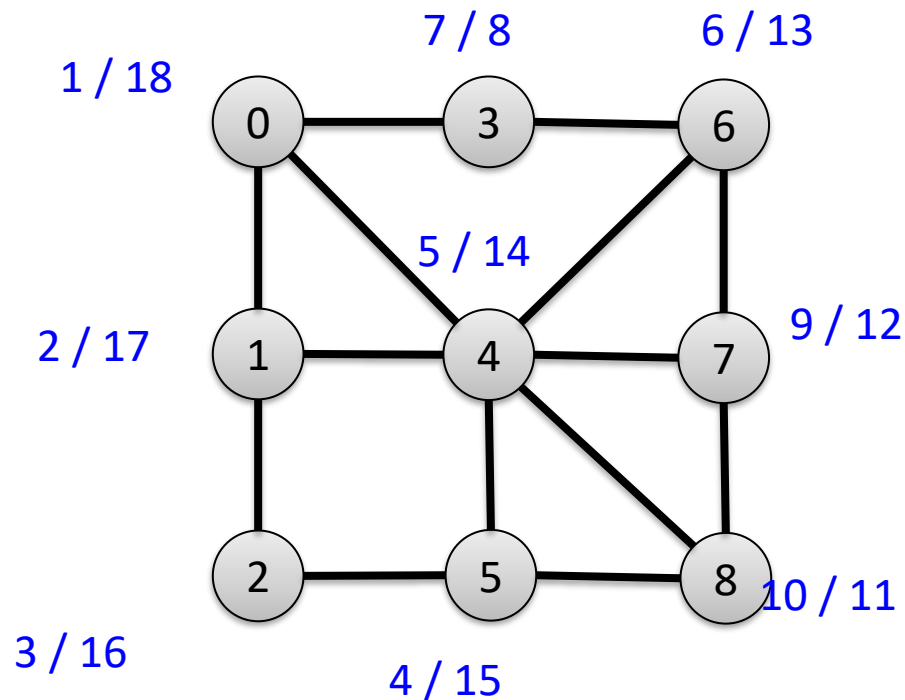
- Exercício:



# Percursos em grafos

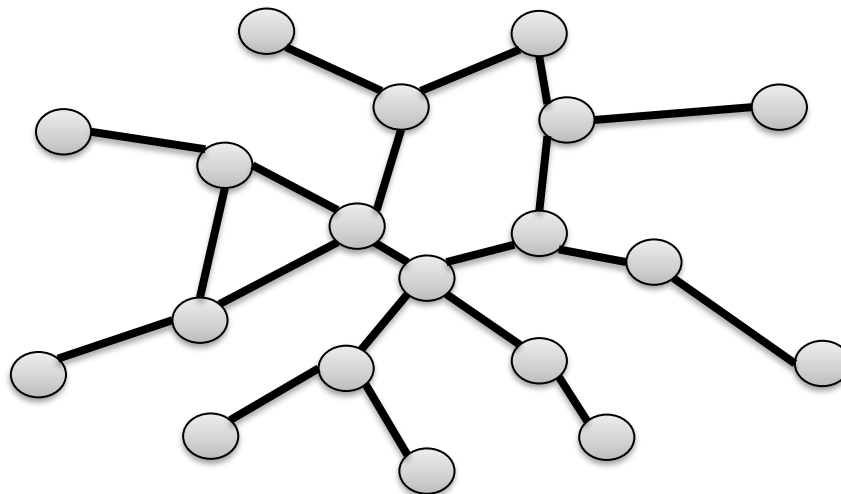
- Busca em profundidade

— Exercício:      Ordem da busca em profundidade: 0, 1, 2, 5, 4, 6, 3, 7, 8



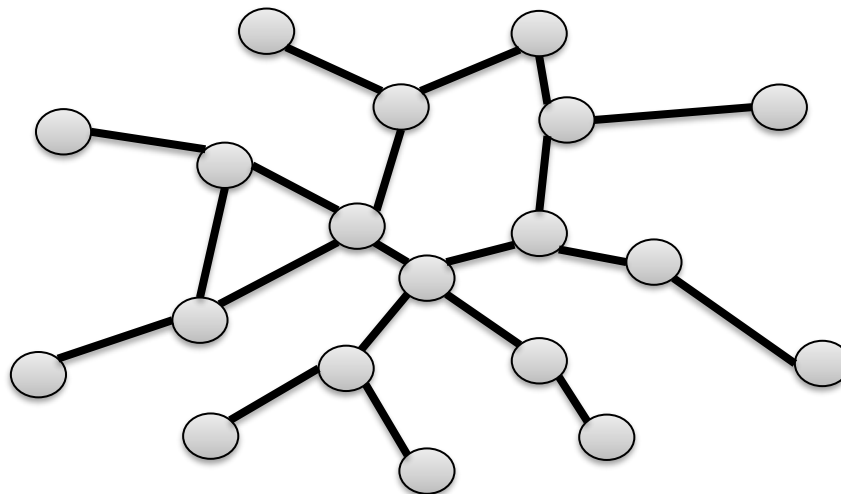
# Percursos em grafos

- Busca em largura
  - O percurso visita **todos os nós** a uma **distância k** do vértice origem **antes** de descobrir qualquer vértice a uma **distância k+1**.
  - A lista de vértices obedece a política FIFO (First-In-First-Out).
  - Assim, é utilizada uma estrutura de dados auxiliar:



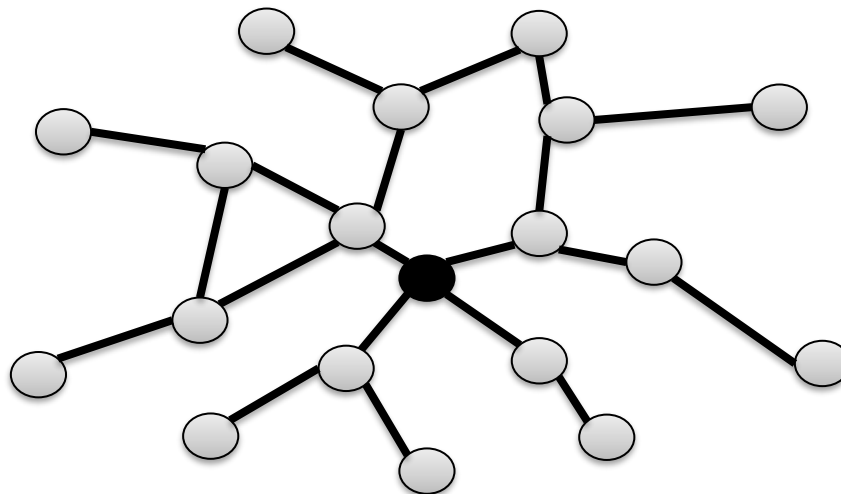
# Percursos em grafos

- Busca em largura
  - O percurso visita **todos os nós** a uma **distância k** do vértice origem **antes** de descobrir qualquer vértice a uma **distância k+1**.
  - A lista de vértices obedece a política FIFO (First-In-First-Out).
  - Assim, é utilizada uma estrutura de dados auxiliar: uma fila.



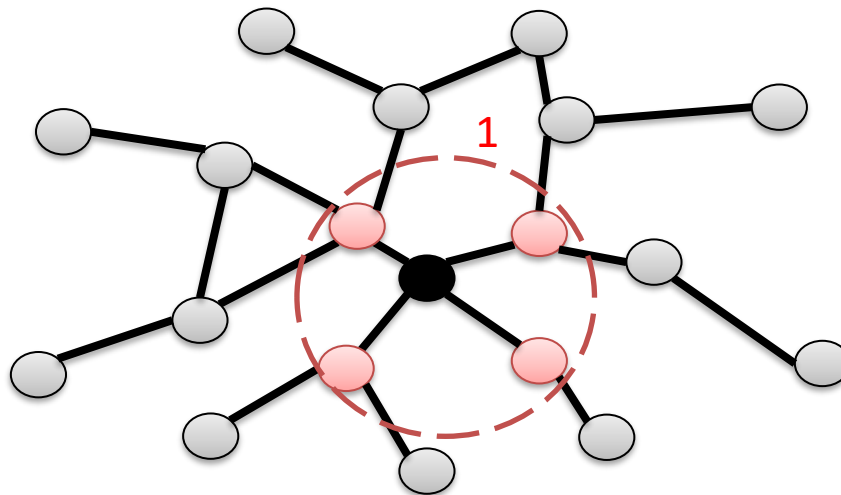
# Percursos em grafos

- Busca em largura
  - O percurso visita **todos os nós** a uma **distância k** do vértice origem **antes** de descobrir qualquer vértice a uma **distância k+1**.
  - A lista de vértices obedece a política FIFO (First-In-First-Out).
  - Assim, é utilizada uma estrutura de dados auxiliar: uma fila.



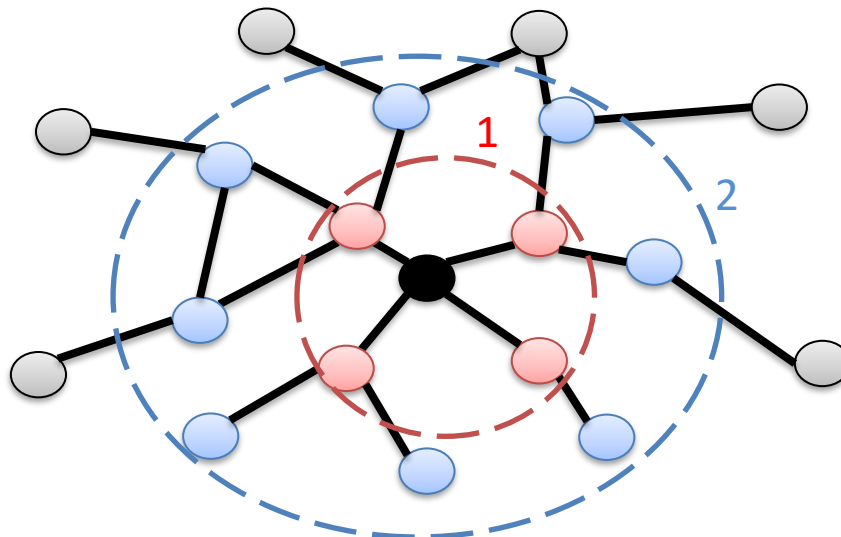
# Percursos em grafos

- Busca em largura
  - O percurso visita **todos os nós** a uma **distância k** do vértice origem **antes** de descobrir qualquer vértice a uma **distância k+1**.
  - A lista de vértices obedece a política FIFO (First-In-First-Out).
  - Assim, é utilizada uma estrutura de dados auxiliar: uma fila.



# Percursos em grafos

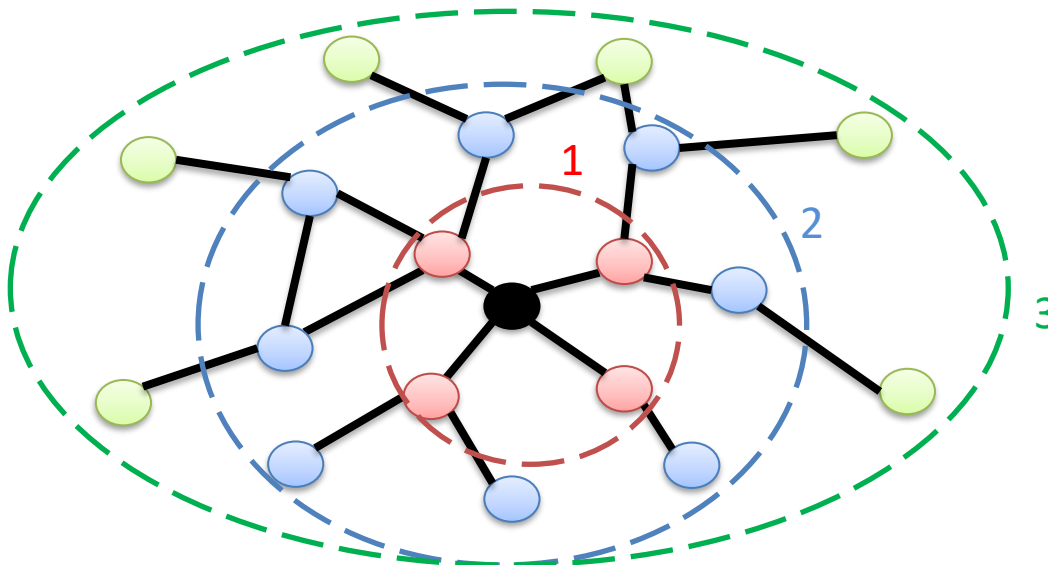
- Busca em largura
  - O percurso visita **todos os nós** a uma **distância k** do vértice origem **antes** de descobrir qualquer vértice a uma **distância k+1**.
  - A lista de vértices obedece a política FIFO (First-In-First-Out).
  - Assim, é utilizada uma estrutura de dados auxiliar: uma fila.





# Percursos em grafos

- Busca em largura
  - O percurso visita **todos os nós** a uma **distância k** do vértice origem **antes** de descobrir qualquer vértice a uma **distância k+1**.
  - A lista de vértices obedece a política FIFO (First-In-First-Out).
  - Assim, é utilizada uma estrutura de dados auxiliar: uma fila.



# Percursos em grafos

- Busca em largura

- Passo-a-passo:

1. Escolha um vértice inicial **v não visitado** e o insira em uma fila.
2. Pegue um elemento da **fila**, marque como **visitado** e acrescente seus **adjacentes** na fila.
3. Enquanto a fila **não está vazia**, volte ao passo 2.

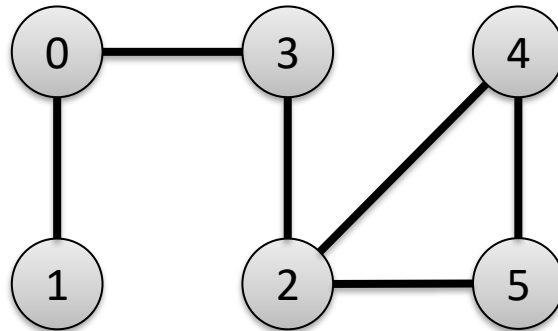
**Convenção:** Por convenção, a ordem de se adicionar os nós adjacentes na fila será **crescente** em relação aos índices.

# Percursos em grafos

- **Busca em largura**

Nó inicial: 3

— Exemplo:

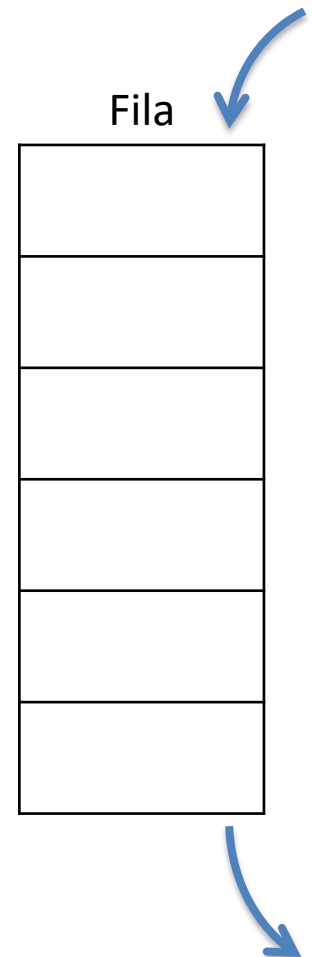
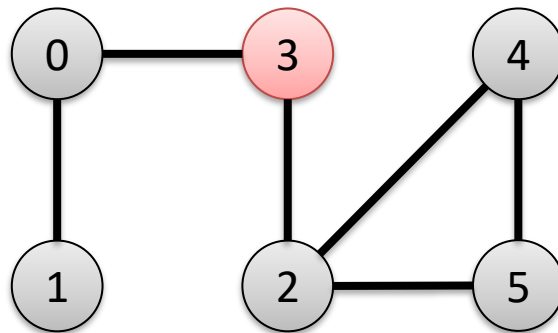


# Percursos em grafos

- **Busca em largura**

— Exemplo:

Nó inicial: 3  Marcado como visitado

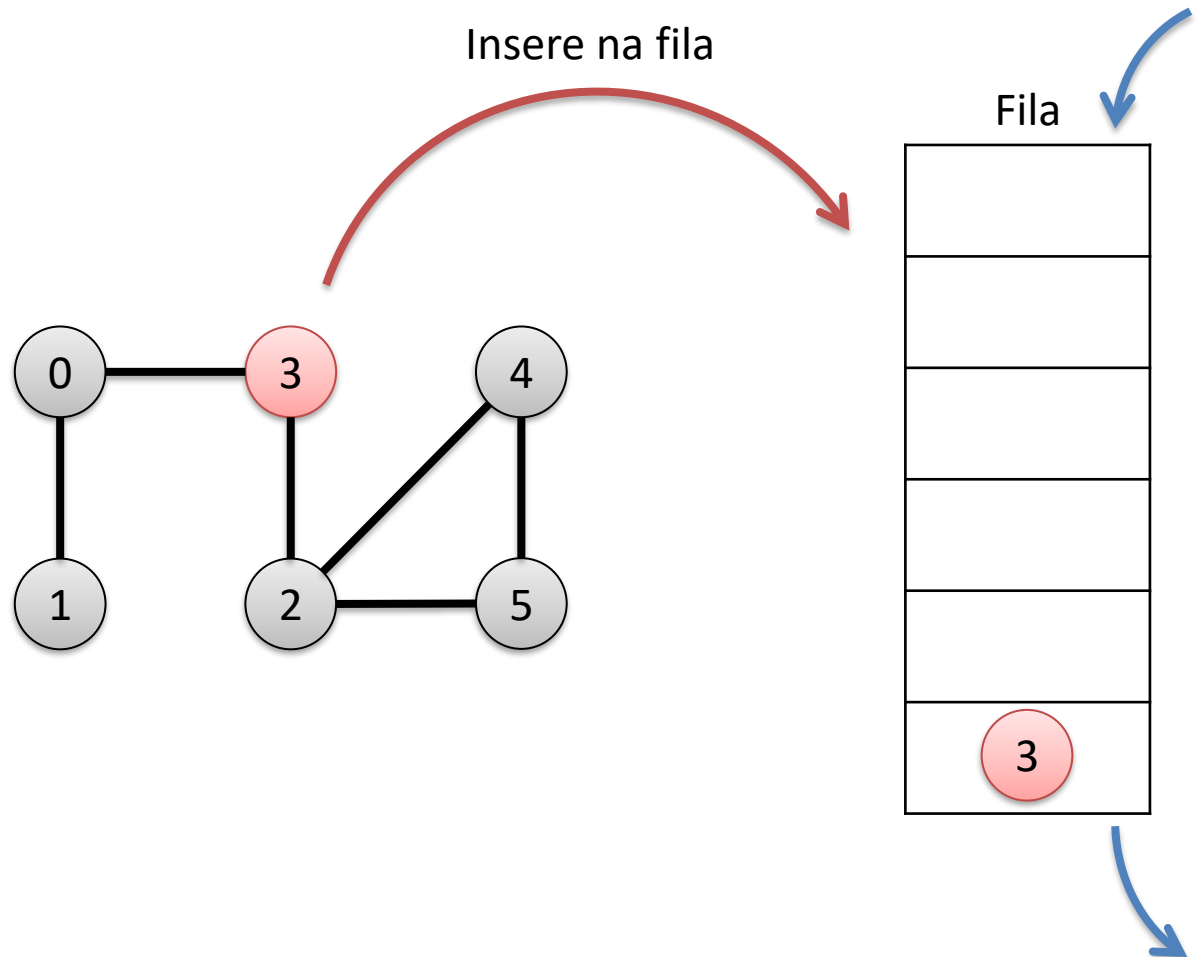


Ordem da busca em largura:

# Percursos em grafos

- **Busca em largura**

— Exemplo:

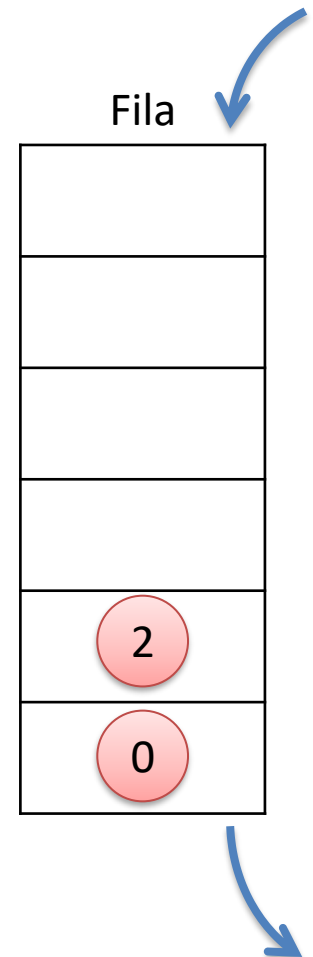
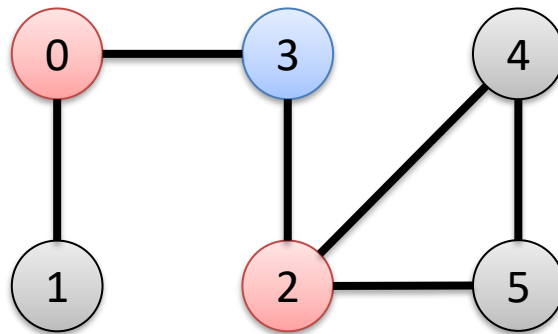


Ordem da busca em largura:

# Percursos em grafos

- Busca em largura

– Exemplo:

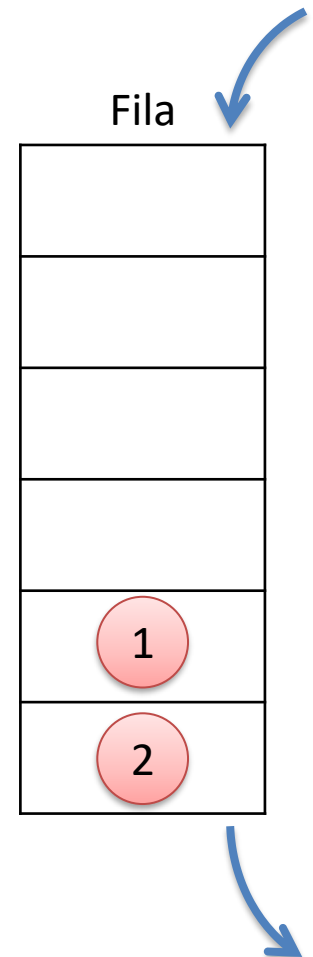
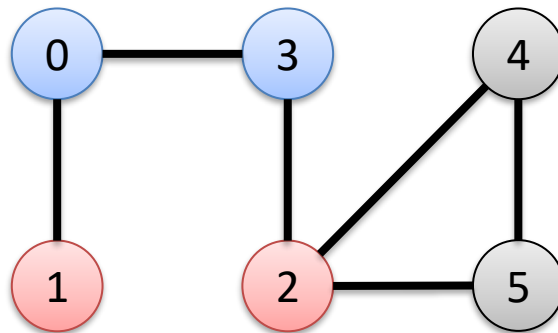


Ordem da busca em largura: 3

# Percursos em grafos

- Busca em largura

– Exemplo:

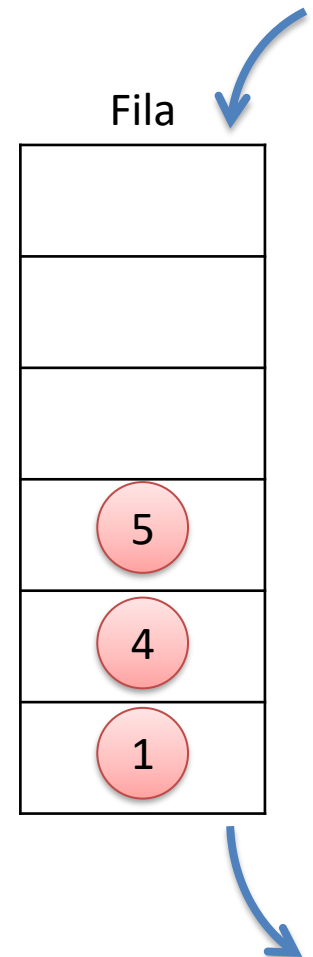
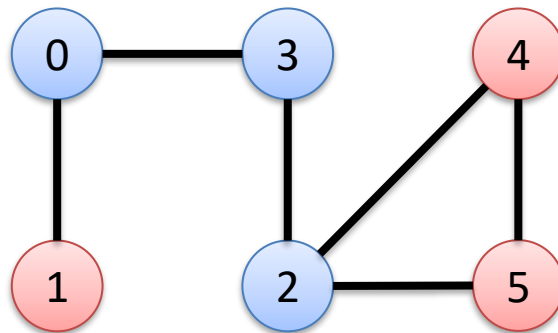


Ordem da busca em largura: 3, 0

# Percursos em grafos

- **Busca em largura**

- Exemplo:



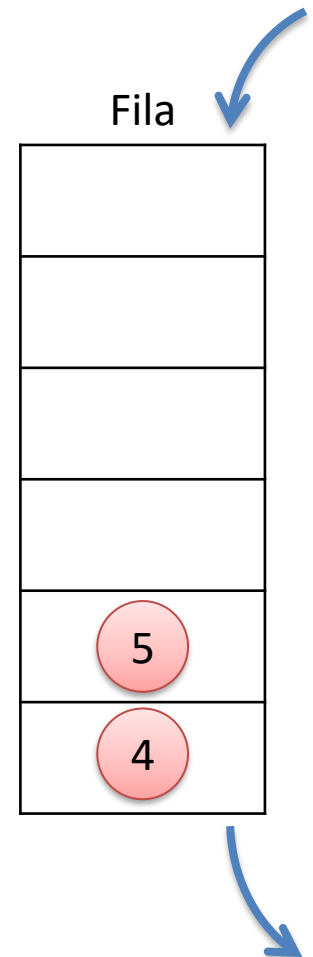
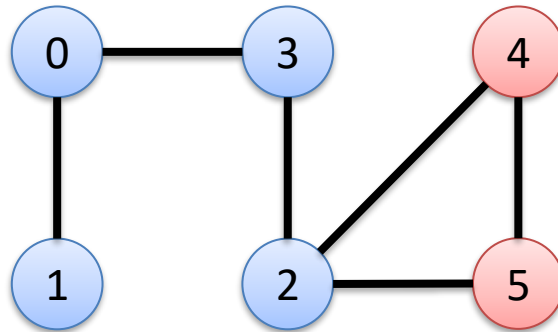
Ordem da busca em largura: 3, 0, 2



# Percursos em grafos

- **Busca em largura**

- Exemplo:

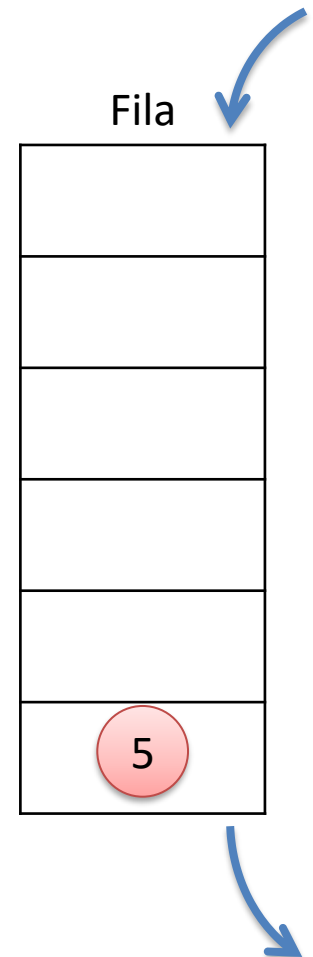
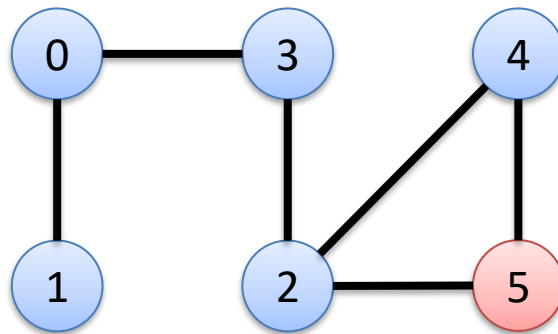


Ordem da busca em largura: 3, 0, 2, 1

# Percursos em grafos

- **Busca em largura**

– Exemplo:

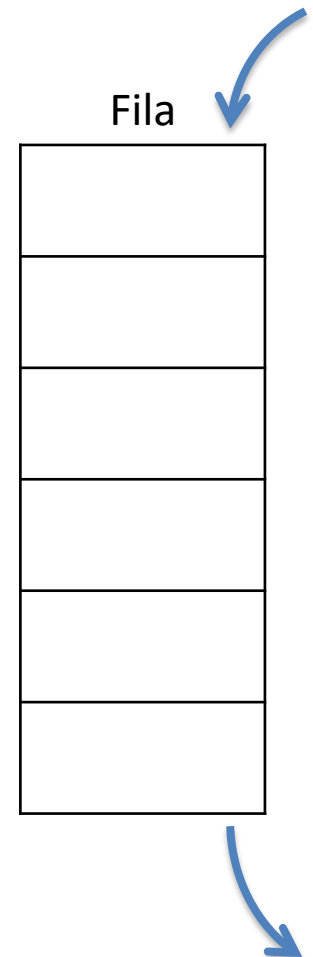
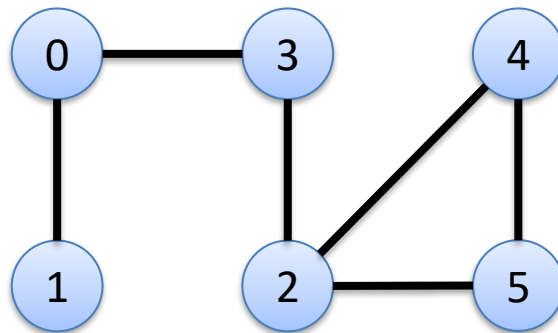


Ordem da busca em largura: 3, 0, 2, 1, 4

# Percursos em grafos

- **Busca em largura**

- Exemplo:

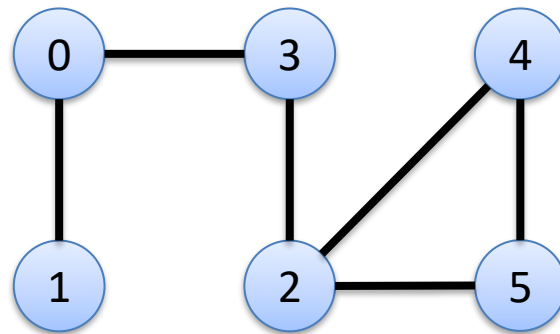


Ordem da busca em largura: 3, 0, 2, 1, 4, 5

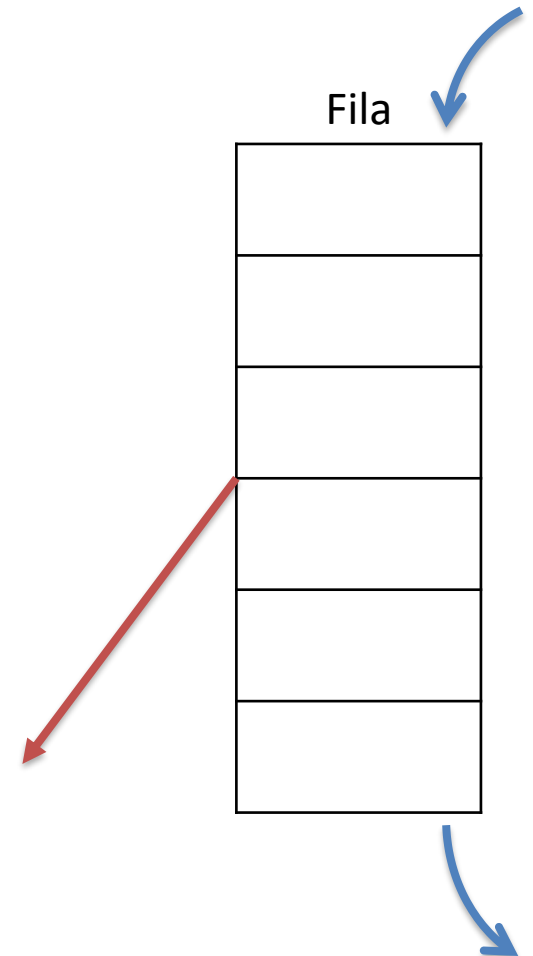
# Percursos em grafos

- **Busca em largura**

- Exemplo:



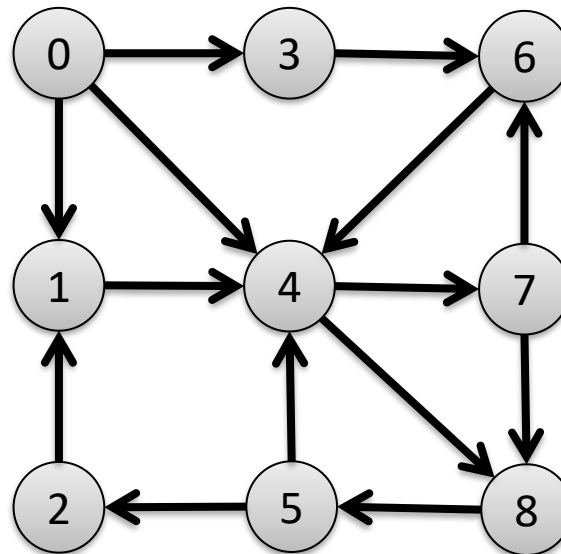
Fila vazia: Fim!



Ordem da busca em largura: 3, 0, 2, 1, 4, 5

# Percursos em grafos

- **Busca em largura**
  - Exercício:



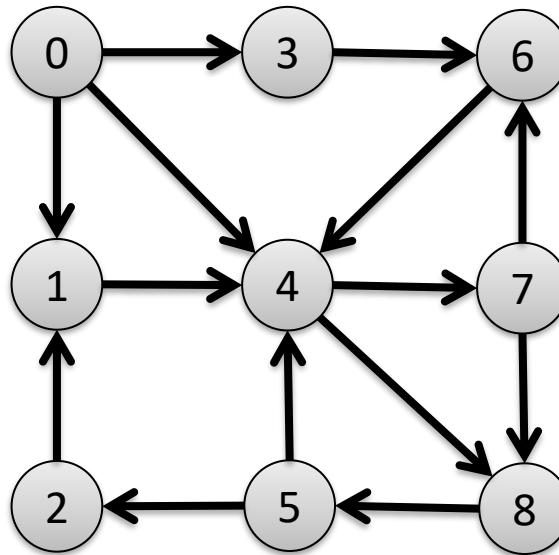
Fila

[illegible]

# Percursos em grafos

- **Busca em largura**

— Exercício:

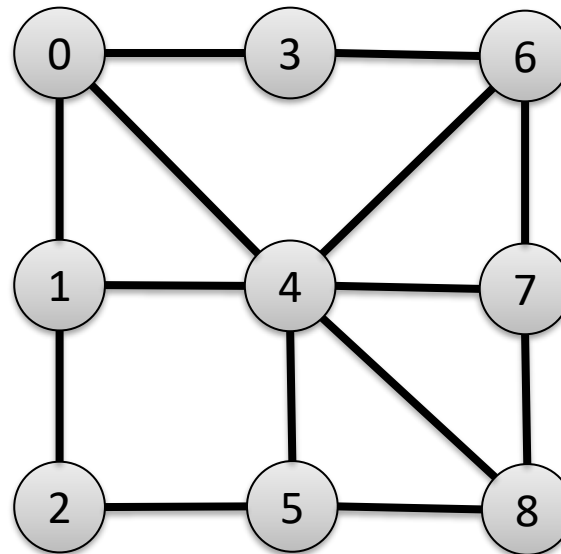


Fila


Ordem da busca em largura: 0, 1, 3, 4, 6, 7, 8, 5, 2

# Percursos em grafos

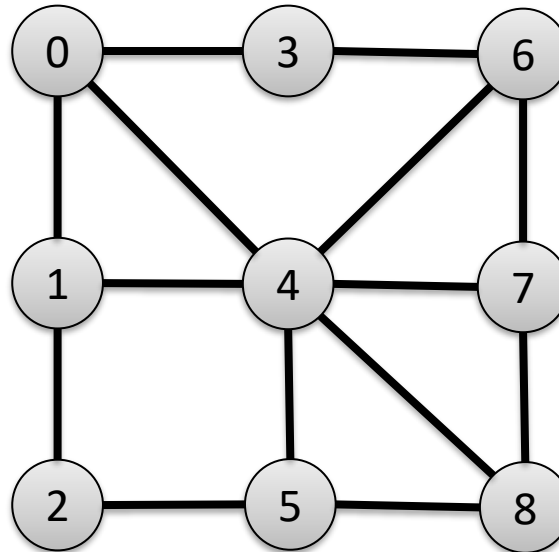
- **Busca em largura**
  - Exercício:

[illegible]

# Percursos em grafos

- **Busca em largura**

— Exercício:



Fila


Ordem da busca em largura: 0, 1, 3, 4, 2, 6, 5, 7, 8



# Percursos em grafos

---

FIM