

# GRAFOS

## DCE529 - Algoritmos e Estruturas de Dados III

Atualizado em: 12 de março de 2024

Iago Carvalho

Departamento de Ciência da Computação



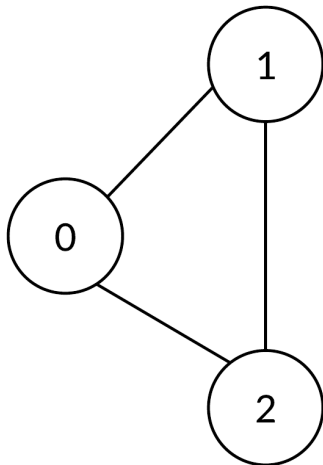
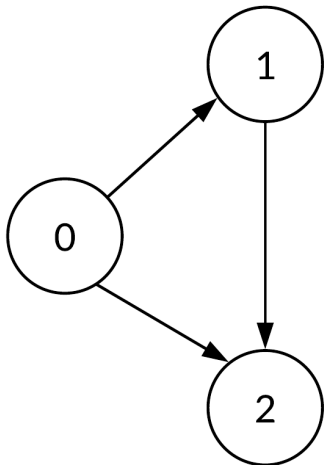
Diversos problemas computacionais podem ser representados como grafos

- Uma estrutura de dados especial
- Representação de uma rede
- Talvez seja a estrutura mais útil em toda a Ciência da Computação

Um grafo  $G$  é definido como  $G = (V, E)$

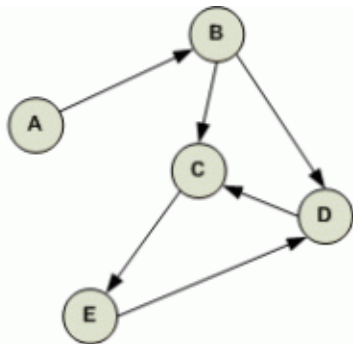
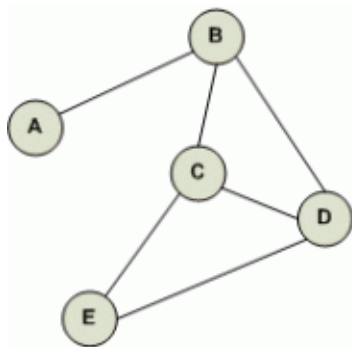
- $V = \{v_1, v_2, \dots, v_n\}$  é o conjunto de vértices
- $E = \{e_1, e_2, \dots, e_m\}$ 
  - $e_i = (u, v) \mid u, v \in V$

Um grafo pode ser direcionado ou não-direcionado

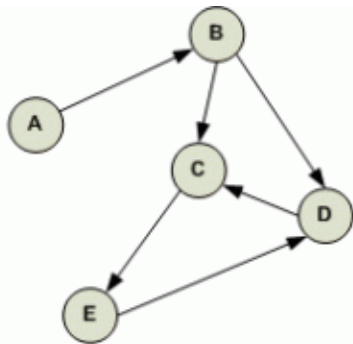
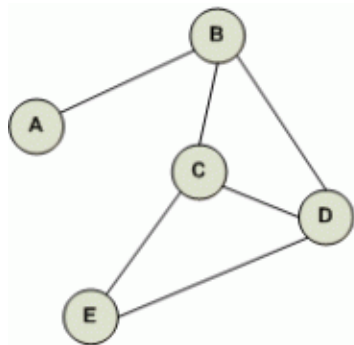


## CAMINHOS E CICLOS

Caminho  $C = \langle c, e, d, c \rangle$

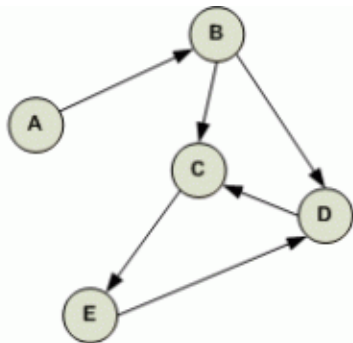
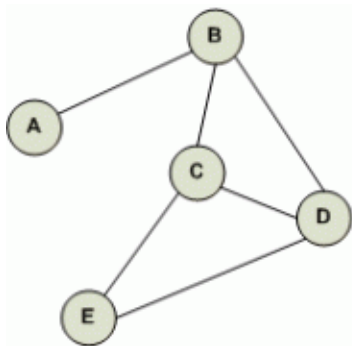


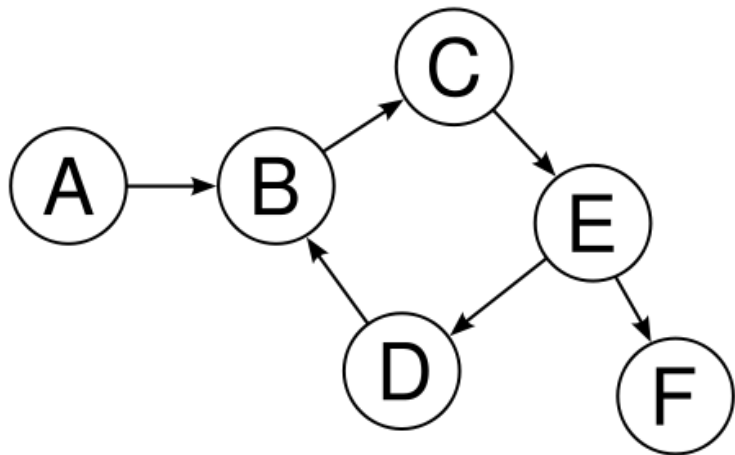
## ADJACÊNCIA E GRAU



# FECHO TRANSITIVO

Direto e inverso

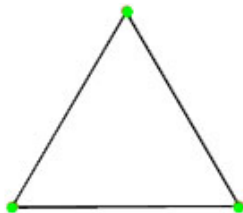




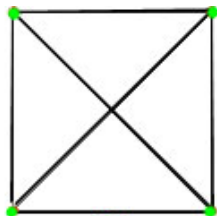
# GRAFO COMPLETO



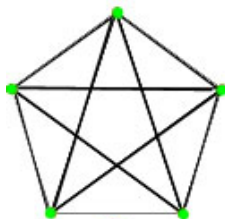
K2



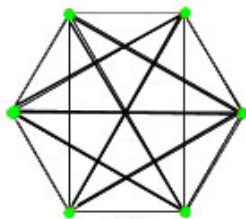
K3



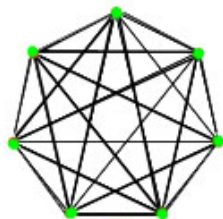
K4



K5



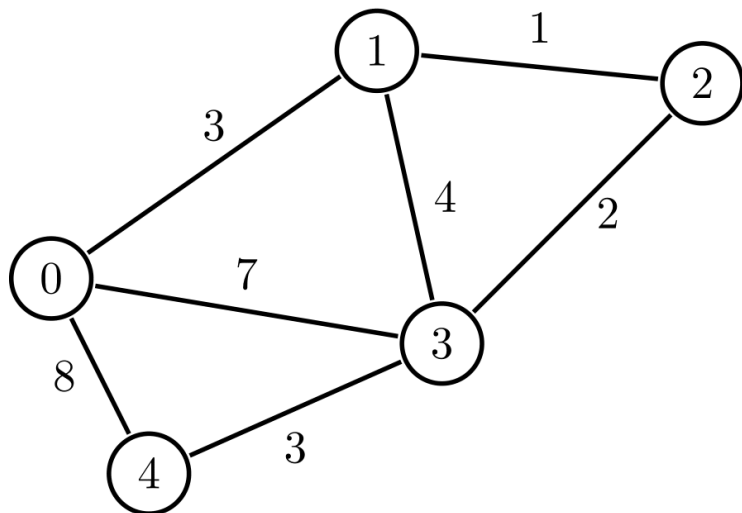
K6



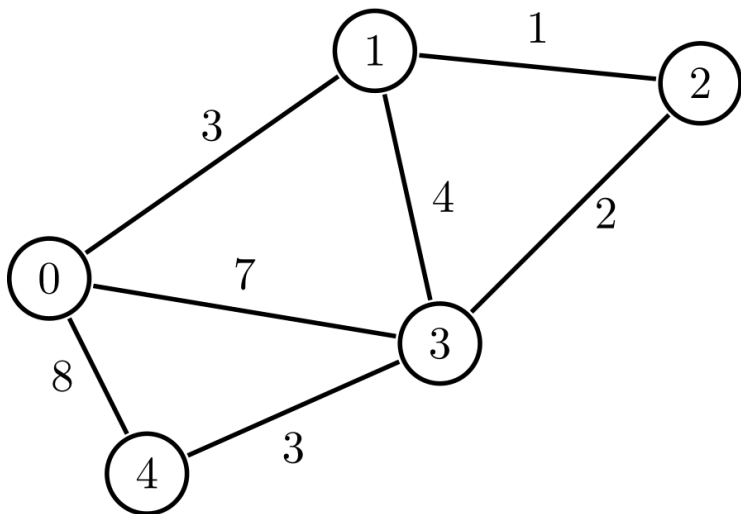
K7



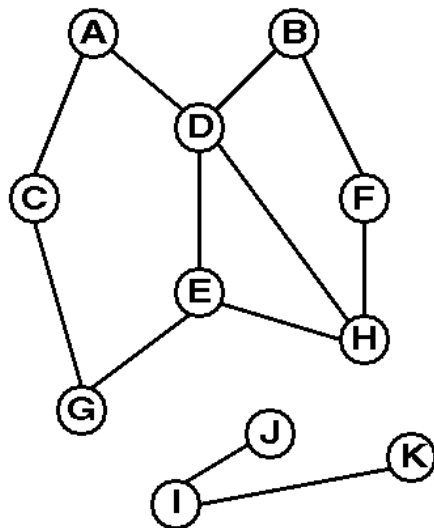
## GRAFO COM PESOS



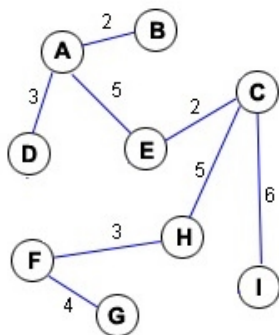
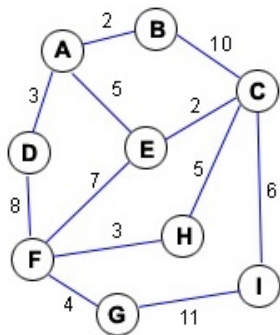
## GRAFO CONEXO



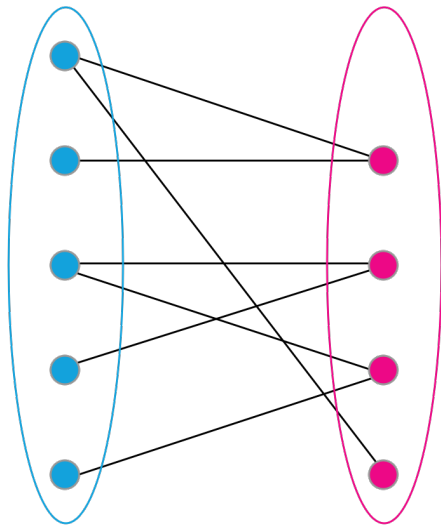
## GRAFO DESCONECTADO E COMPONENTES CONEXAS



## ÁRVORE GERADORA (MÍNIMA)



# GRAFO BIPARTIDO



Diversas destas propriedades serão utilizadas no decorrer deste curso

Grafos são uma das estruturas mais importantes em Ciência da Computação, tendo aplicações em uma infinidade de áreas

- Redes
- Biologia
- Eletrônica
- Pesquisa Operacional
- ... [▶ Link](#)

Interessados em um pouco mais de propriedades de grafos podem acessar o seguinte link [▶ Link](#)

Existem duas estruturas de dados capazes de representar grafos

- Matriz de adjacência
- Lista de adjacência

Cada estrutura difere-se da outra pela complexidade de suas operações

- Complexidade de adicionar ou retirar nós
- Complexidade de inserir ou remover arestas
- Complexidade de pesquisa
  - Saber se uma aresta existe ou não
- Diferentes complexidades de espaço

# MATRIZ DE ADJACÊNCIA

Talvez seja a maneira mais natural de se representar um grafo

- Grafo com  $n$  vértices
- Matriz bi-dimensional  $n \times n$
- Complexidade de espaço:  $\mathcal{O}(n^2) = \mathcal{O}(m)$

Inserção e remoção de vértices é cara

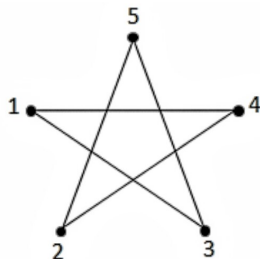
- Necessário alocar ou desalocar memória

Modificação de arestas e pesquisa é barata

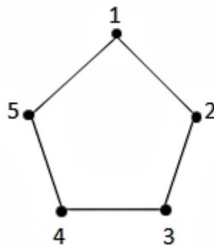
- Necessário apenas modificar (ou verificar) uma célula específica da matriz



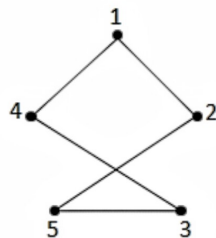
# MATRIZ DE ADJACÊNCIA



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

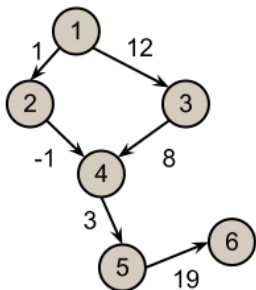


	1	2	3	4	5
1		1			1
2	1		1		
3		1		1	
4			1		1
5	1			1	



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

## Weighted Directed Graph & Adjacency Matrix



Weighted Directed Graph

	1	2	3	4	5	6
1	0	1	12	0	0	0
2	-1	0	0	-1	0	0
3	-12	0	0	8	0	0
4	0	1	-8	0	3	0
5	0	0	0	-3	0	19
6	0	0	0	0	-19	0

Adjacency Matrix

# LISTA DE ADJACÊNCIA

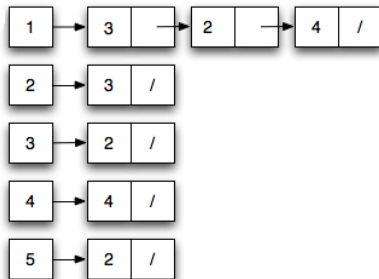
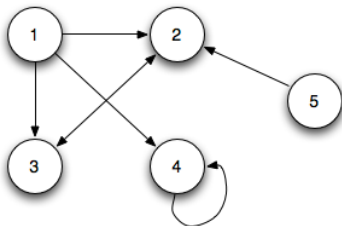
Uma lista de adjacência pode ser representada como uma lista de listas

- Uma lista que contém todos os vértices do grafo
- Cada lista contém outra lista
  - Contém todos os vértices adjacentes

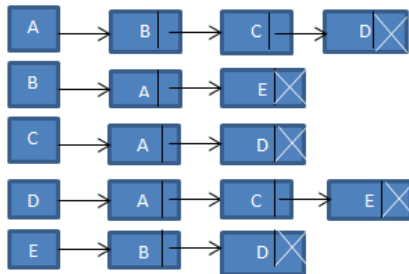
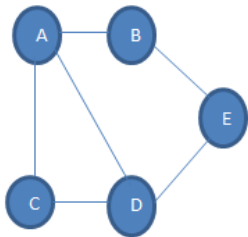
Complexidades diferem das de matriz de adjacência

- Complexidade de espaço:  $\mathcal{O}(n^2) = \mathcal{O}(m)$
- Inserção, pesquisa e remoção de arestas:  $\mathcal{O}(n)$
- Inserção e remoção de vértices:  $\mathcal{O}(1)$

# LISTA DE ADJACÊNCIA



# LISTA DE ADJACÊNCIA



# LISTA DE ADJACÊNCIA

