

EDCO3A

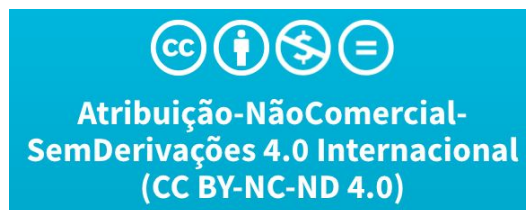
ESTRUTURAS DE DADOS 1

Aula 08 - Grafos

Profa Tamara Baldo

Licença

Este trabalho está licenciado com uma Licença CC BY-NC-ND 4.0:



maiores informações:

https://creativecommons.org/licenses/by-nc-nd/4.0/deed.pt_BR

Roteiro

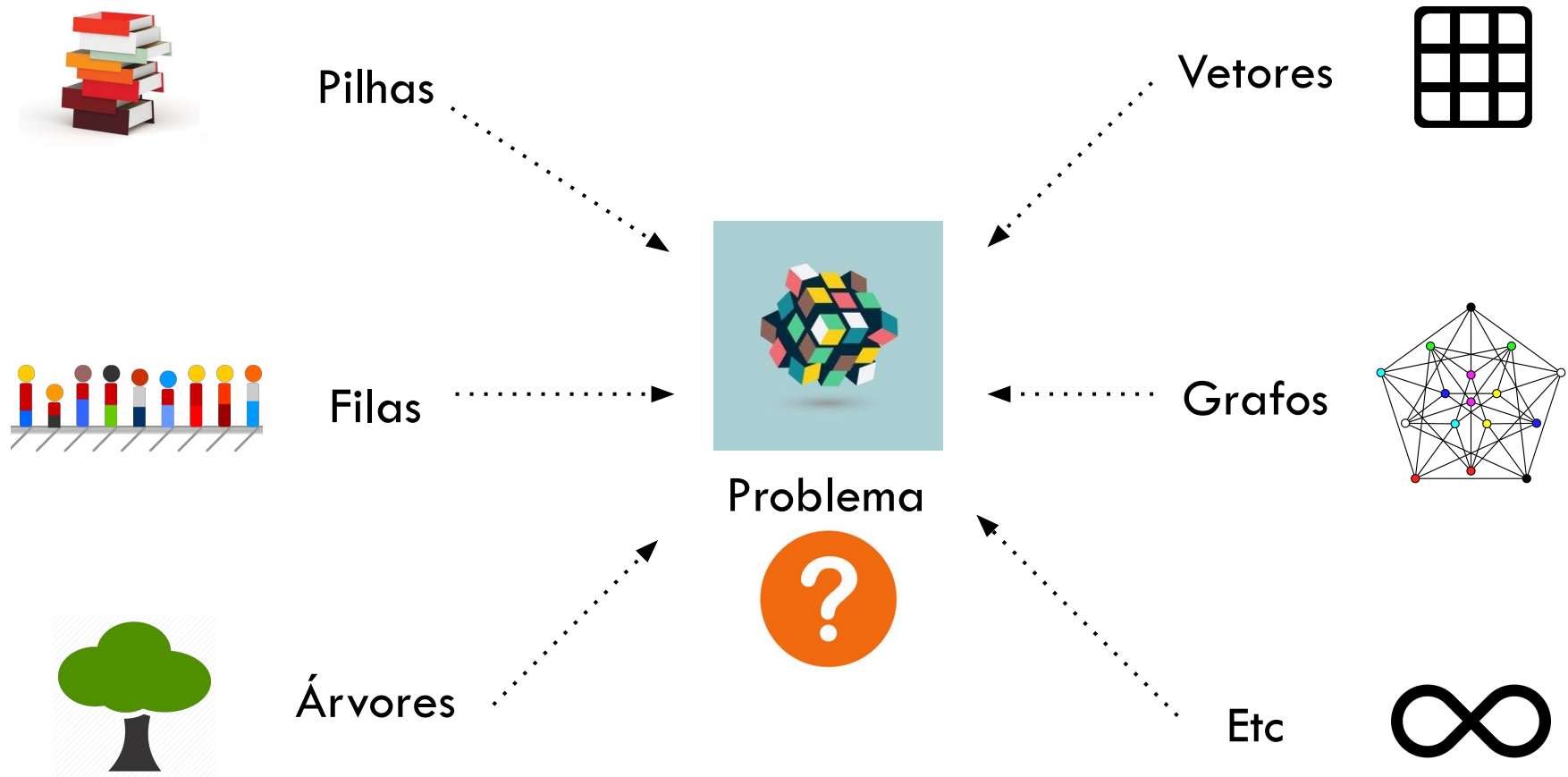


- 1** Introdução
- 2** Grafos
- 3** Definições
- 4** Representações
- 5** Referências

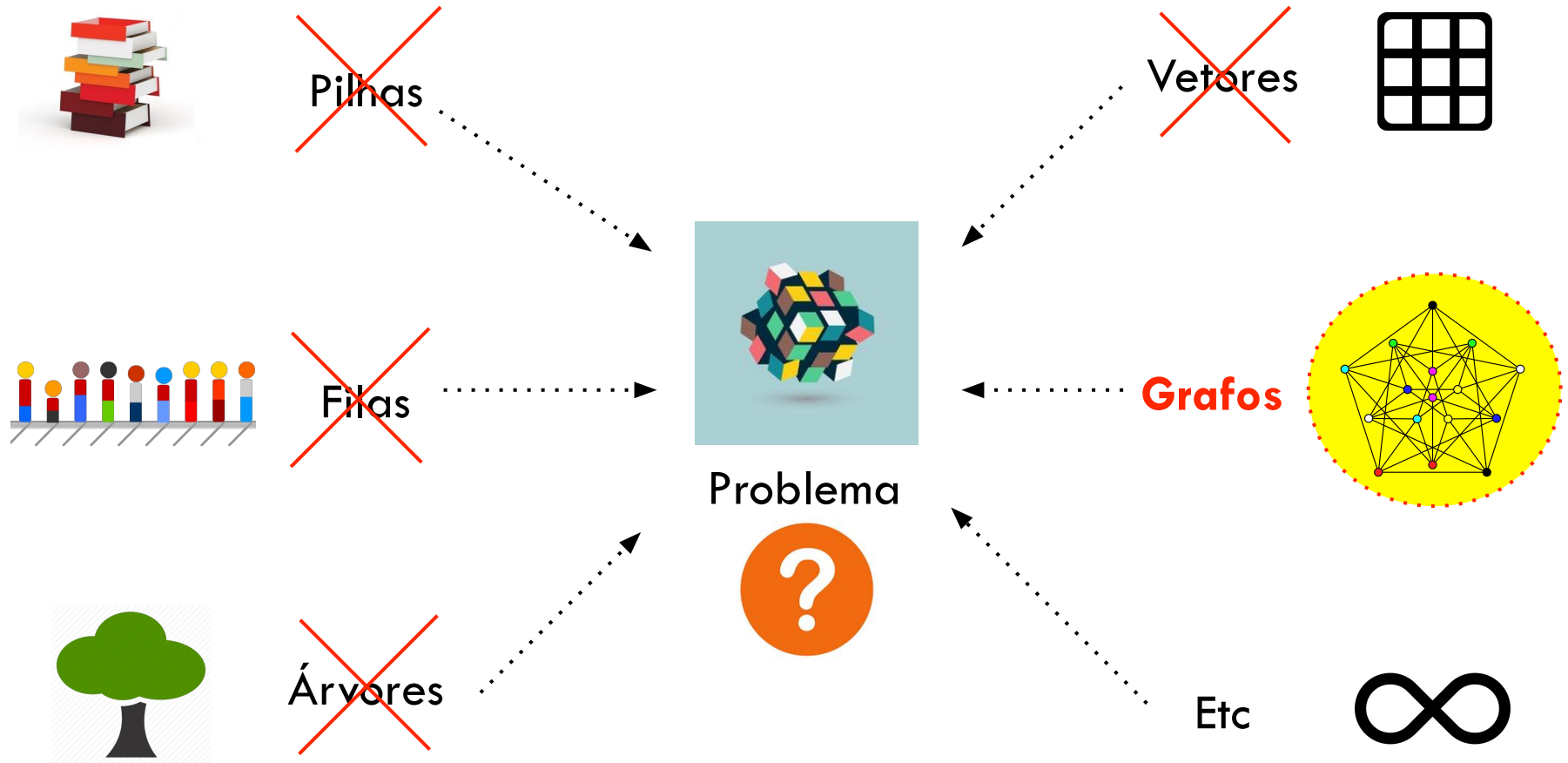
Roteiro

- 1** Introdução
- 2** Grafos
- 3** Definições
- 4** Representações
- 5** Referências

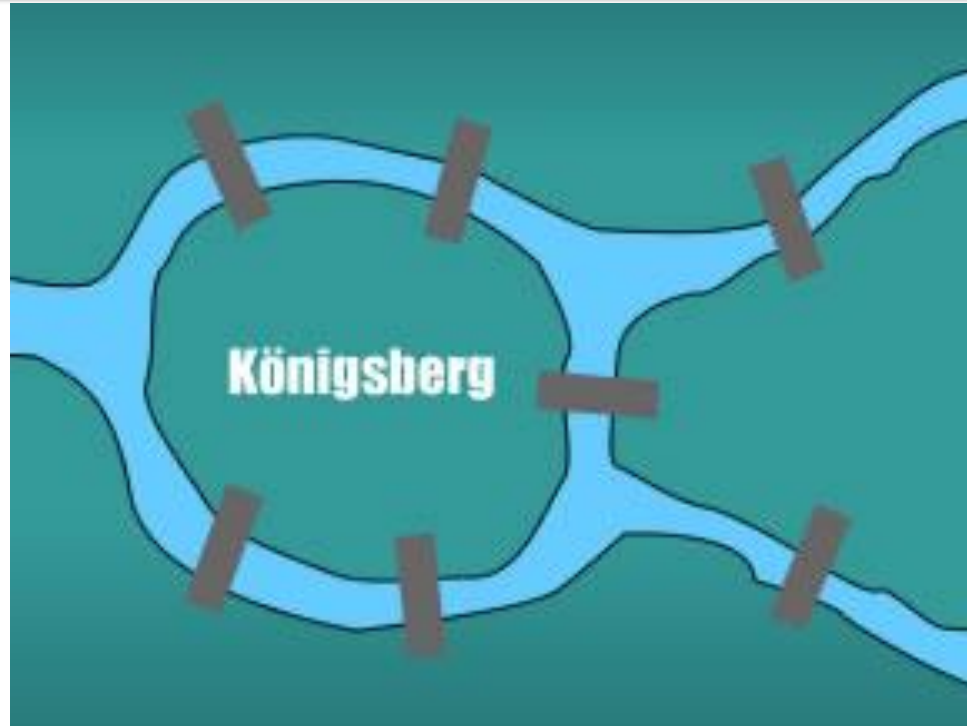
Introdução



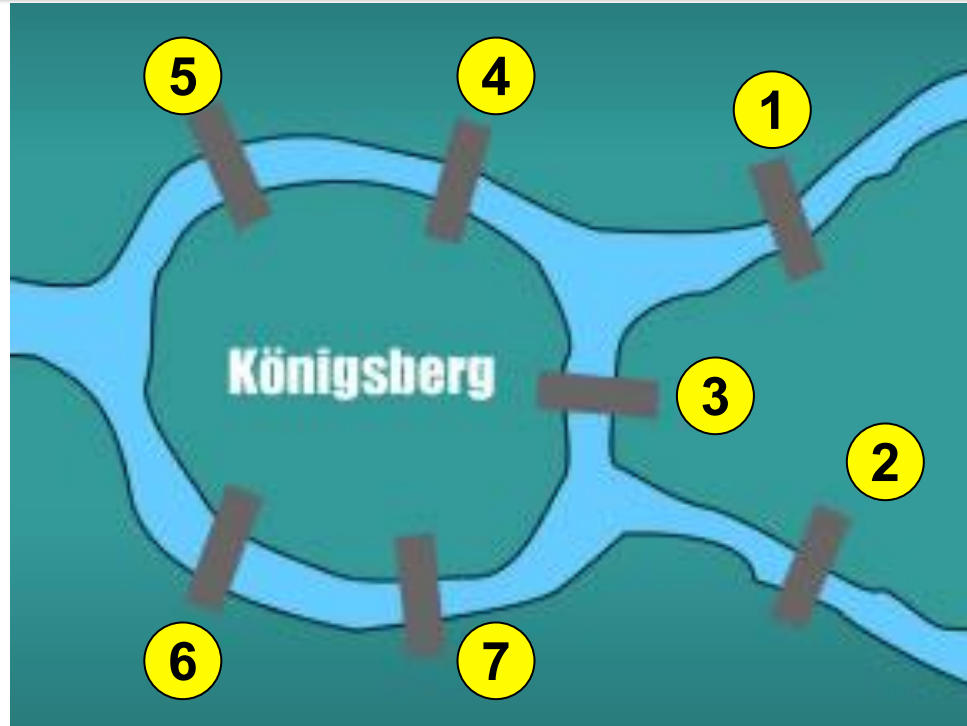
Introdução



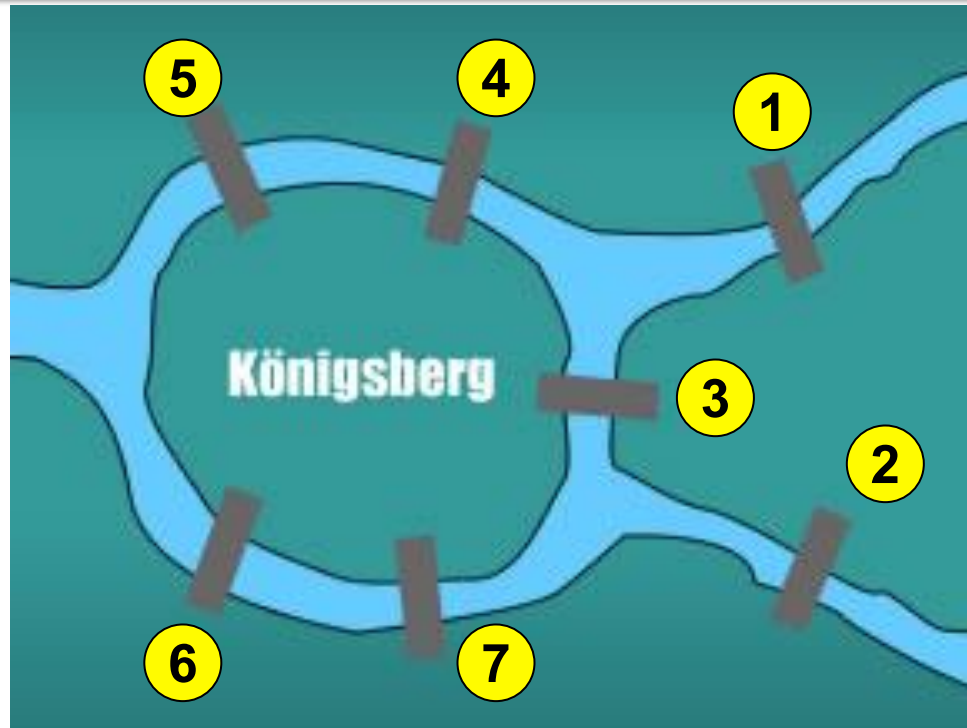
Sete Pontes de Königsberg



Sete Pontes de Königsberg

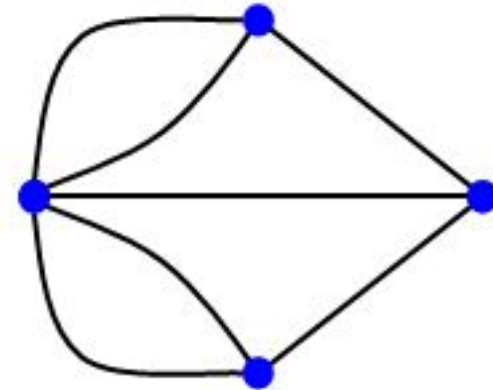
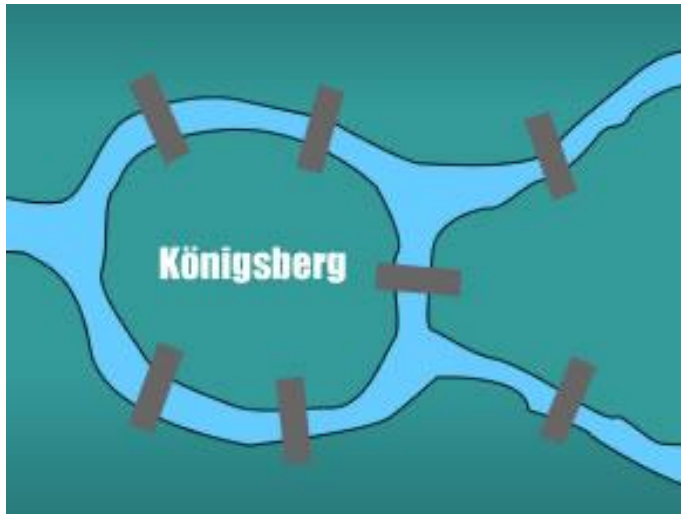


Sete Pontes de Königsberg

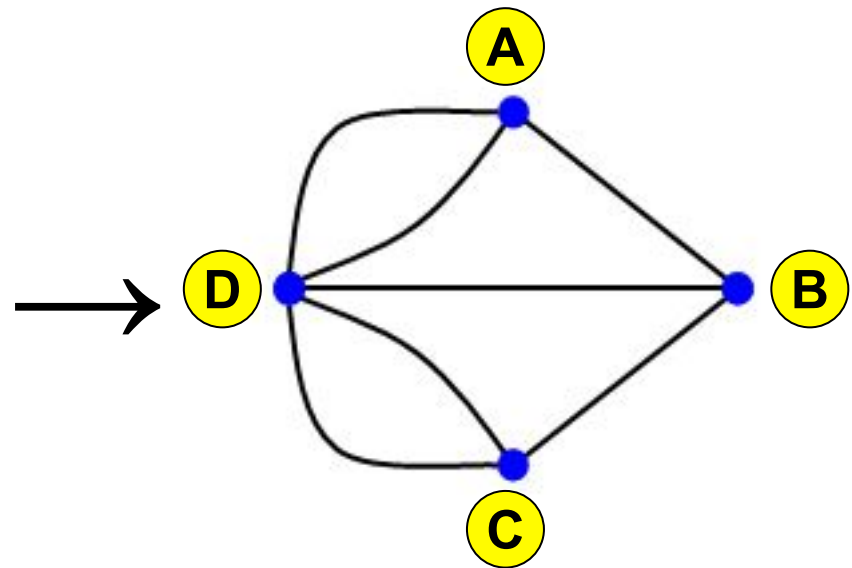
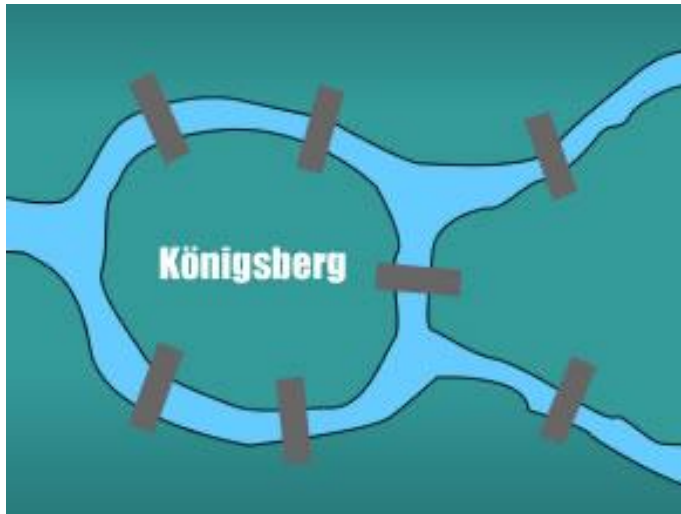


Problema: atravessar as sete pontes durante uma caminhada contínua, sem passar duas vezes por qualquer uma delas

Sete Pontes de Königsberg

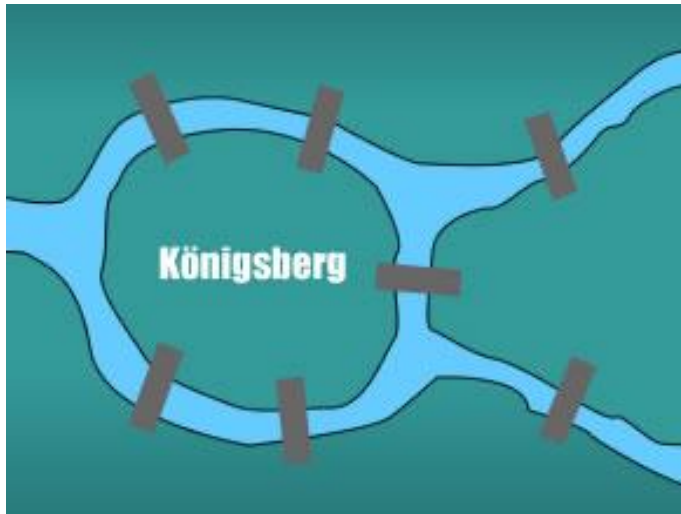


Sete Pontes de Königsberg

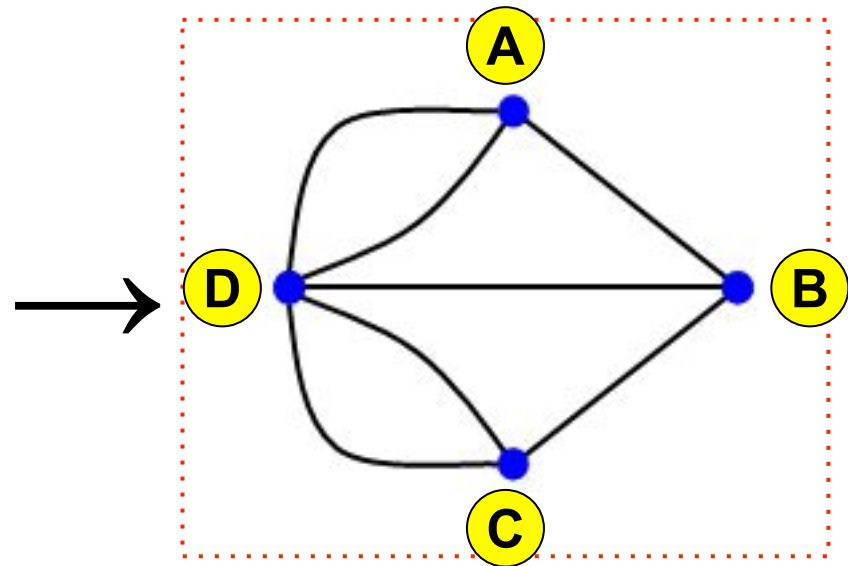


* Solução ?

Sete Pontes de Königsberg



Grafo = (V, E)



- Caminhos: linhas (arestas - E)
- Lugares: pontos (vértices - V)

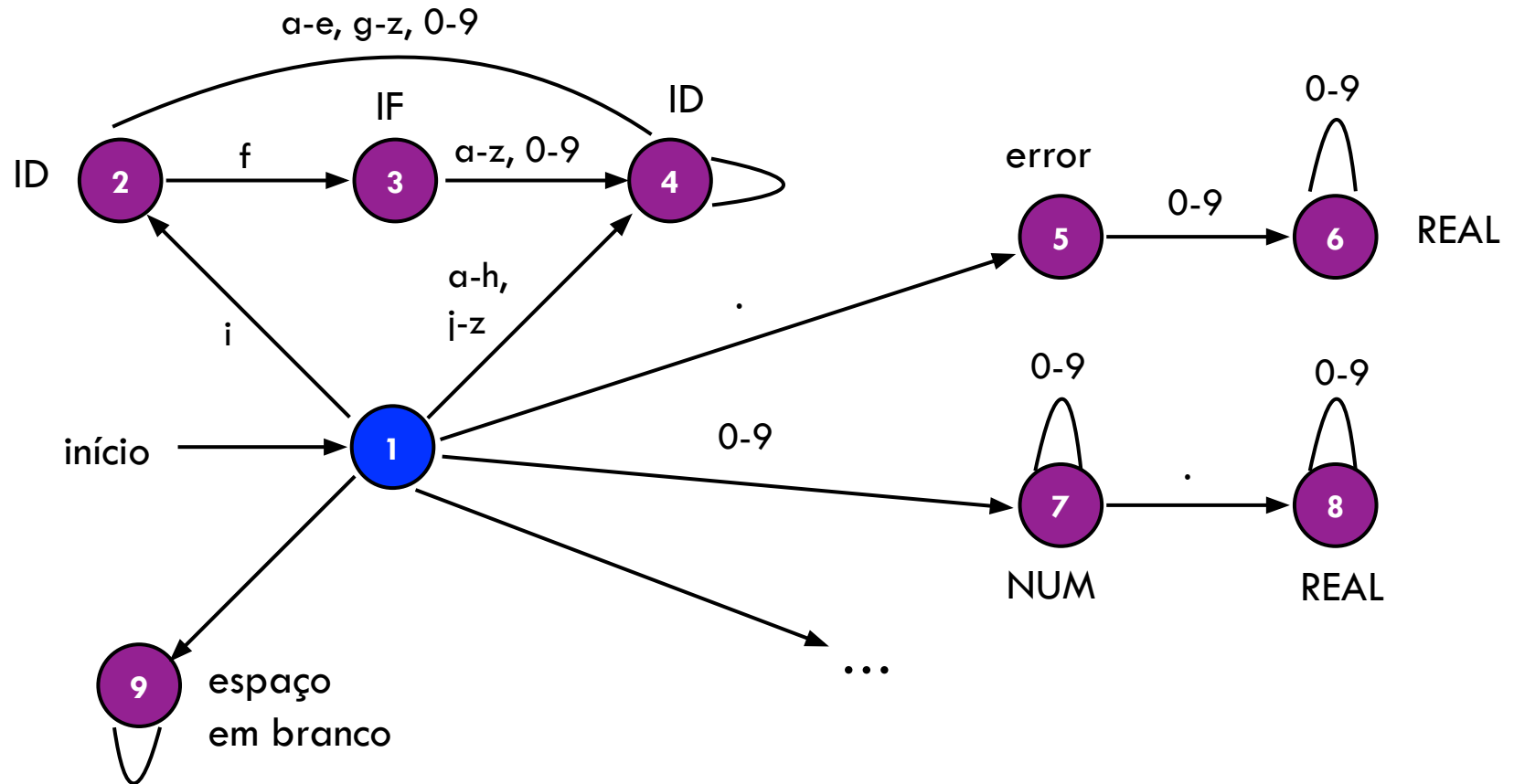
Sete Pontes de Königsberg

- **Euler**: mostrou que **não** existe solução que satisfaça tais restrições.
 - Teoria dos grafos
- Existe solução **apenas** se:
 - houvessem dois vértices com número ímpar de arestas;
 - demais vértices com número par de arestas;

Contextualização

- Diversos problemas podem usar **Grafos** para encontrar uma solução:
 - Análise de circuitos elétricos;
 - Identificação de caminhos mais curtos, rotas, etc;
 - Modelagem de redes pluviais, esgoto, etc;
 - Identificação de compostos químicos;
 - Análise léxica;
 - etc.

Análise léxica



Contextualização

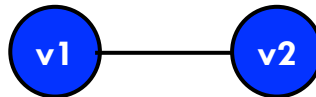
- Diversos problemas podem usar **Grafos** para encontrar uma solução:
 - Análise de circuitos elétricos;
 - Identificação de caminhos mais curtos, rotas, etc;
 - Modelagem de redes pluviais, esgoto, etc;
 - Identificação de compostos químicos;
 - Análise léxica;
 - Pode-se dizer que, de todas as estruturas matemáticas, os grafos são as que se encontram em mais amplo uso.

Roteiro

- 1 Introdução
- 2 Grafos
- 3 Definições
- 4 Representações
- 5 Referências

Grafo

- Um grafo $G = (V, E)$ é definido por:
 - um conjunto de vértices (nós), V
 - um conjunto de arestas (arcos), E
 - cada aresta é especificada por um par de vértices. Ex: $(v1, v2)$



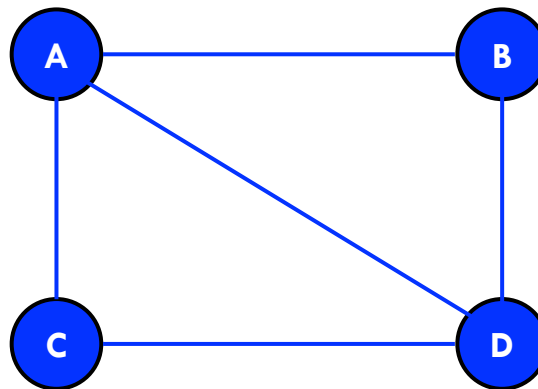
- A análise de complexidade dos algoritmos de grafos é feita com base:
 - tamanho dos conjuntos de vértices $|V|$ e arestas $|E|$
 - $O(V+E) = O(|V| + |E|)$

Grafos não dirigidos

- Um grafo não dirigido (ou não orientado) é um grafo cujas arestas não são dirigidas:
 - $(A, B) = (B, A)$
 - Ex: $G = (V, E)$
 - $V = \{A, B, C, D\}$
 - $E = \{(A,B), (A,C), (B,D), (C,D), (A,D)\}$

Grafos não dirigidos

- Um grafo não dirigido (ou não orientado) é um grafo cujas arestas não são dirigidas:
 - $(A, B) = (B, A)$
 - Ex: $G = (V, E)$
 - $V = \{A, B, C, D\}$
 - $E = \{(A,B), (A,C), (B,D), (C,D), (A,D)\}$

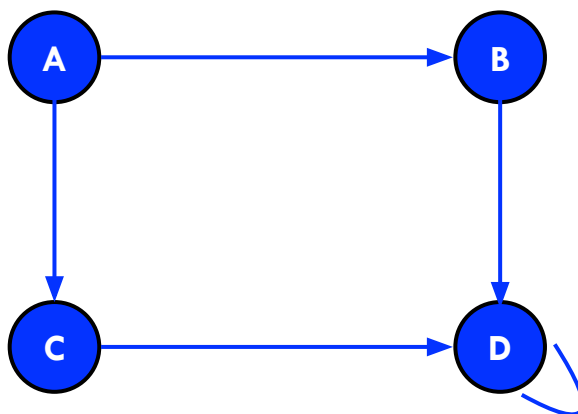


Grafos dirigidos

- Se as arestas do grafo são dirigidas, as arestas são representadas por pares ordenados de vértices, e o grafo é dito dirigido (dígrafo)
 - Aresta: $\langle A, B \rangle$, A é o vértice de origem, B é o vértice de destino
 - $\langle A, B \rangle \neq \langle B, A \rangle$
 - Ex: $G=(V,E)$, $V = \{A, B, C, D\}$, $E = \{\langle A,B \rangle, \langle A,C \rangle, \langle B,D \rangle, \langle C,D \rangle, \langle D,D \rangle\}$

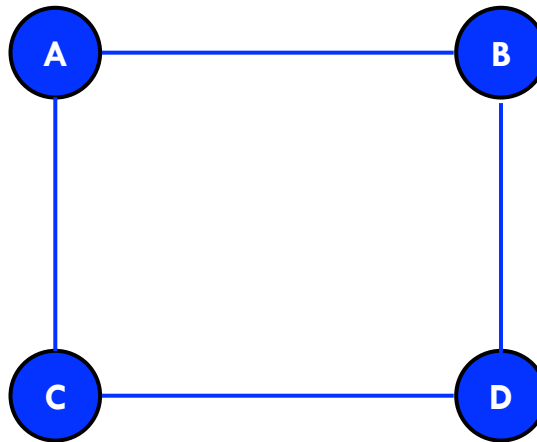
Grafos dirigidos

- Se as arestas do grafo são dirigidas, as arestas são representadas por pares ordenados de vértices, e o grafo é dito dirigido (dígrafo)
 - Aresta: $\langle A, B \rangle$, A é o vértice de origem, B é o vértice de destino
 - $\langle A, B \rangle \neq \langle B, A \rangle$
 - Ex: $G=(V,E)$, $V = \{A, B, C, D\}$, $E = \{\langle A,B \rangle, \langle A,C \rangle, \langle B,D \rangle, \langle C,D \rangle, \langle D,D \rangle\}$



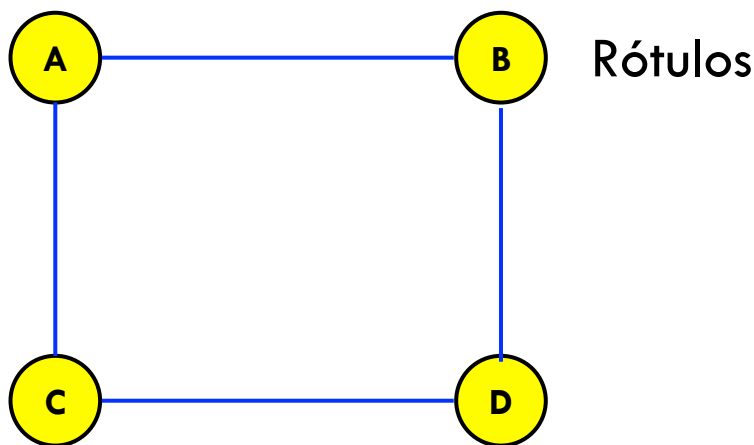
Grafos rotulados e ponderados

- Um grafo é dito ser rotulado em vértices (ou arestas) quando cada vértice (aresta) estiver associado a um rótulo
- Em um grafo ponderado, cada aresta possui um peso associado



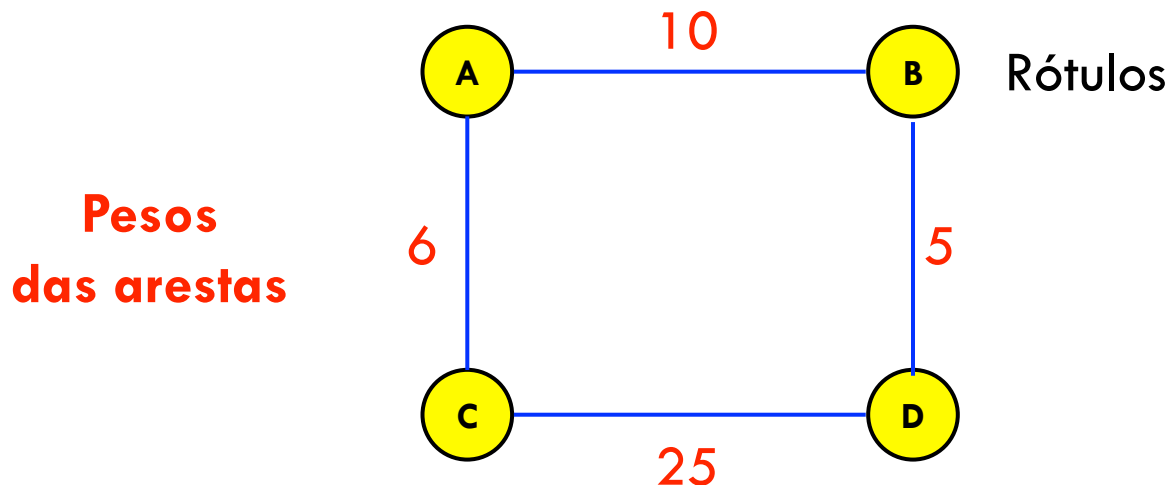
Grafos rotulados e ponderados

- Um grafo é dito ser rotulado em vértices (ou arestas) quando cada vértice (aresta) estiver associado a um rótulo
- Em um grafo ponderado, cada aresta possui um peso associado



Grafos rotulados e ponderados

- Um grafo é dito ser rotulado em vértices (ou arestas) quando cada vértice (aresta) estiver associado a um rótulo
- Em um grafo ponderado, cada aresta possui um peso associado



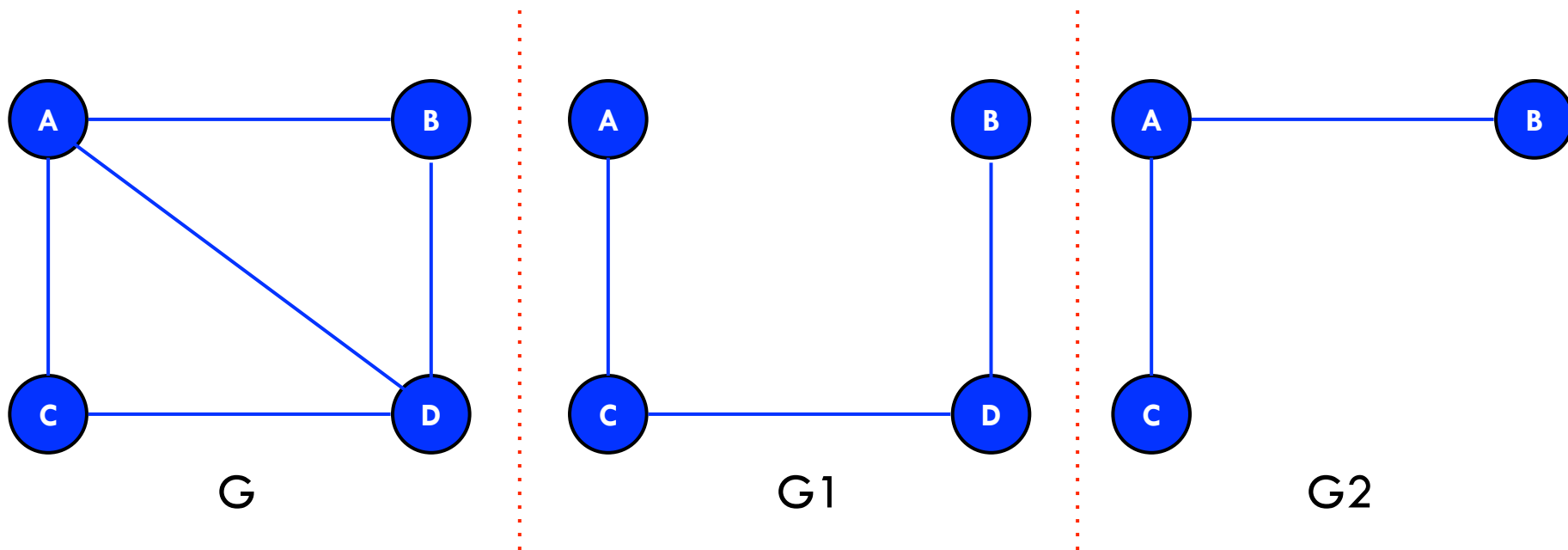
Roteiro



- 1 Introdução
- 2 Grafos
- 3 Definições
- 4 Representações
- 5 Referências

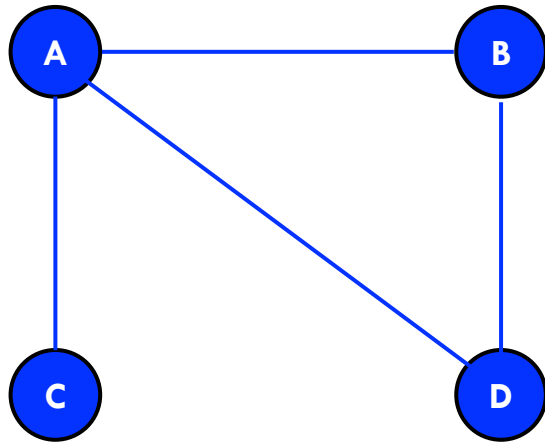
Subgrafo

- Um subgrafo $G'=(V', E')$ de um grafo $G=(V, E)$, se:
 - $V' \subseteq V$ e $E' \subseteq E$
 - Ex: $G1$ e $G2$ são subgrupos de G

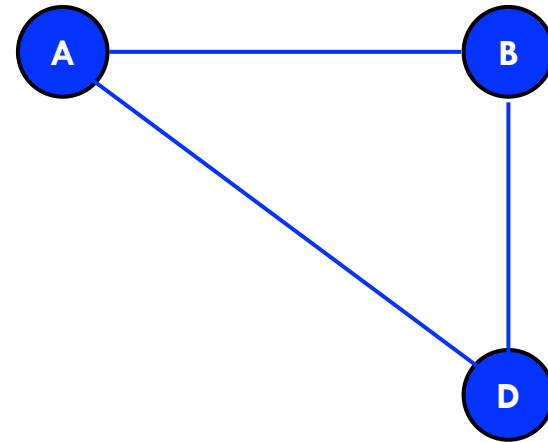


Ordem

- **Ordem:** a ordem de um grafo $G=(V, E)$ é dada pelo número de vértices, isto é, $|V|$



G1
ordem = 4

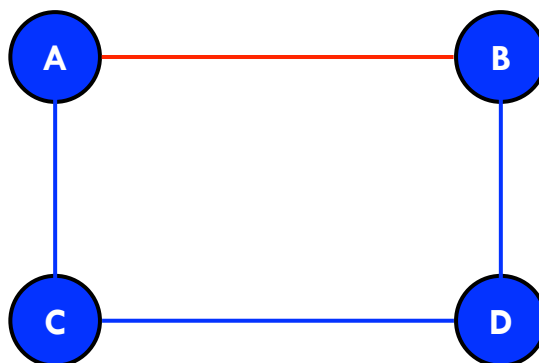


G2
ordem = 3

Adjacência

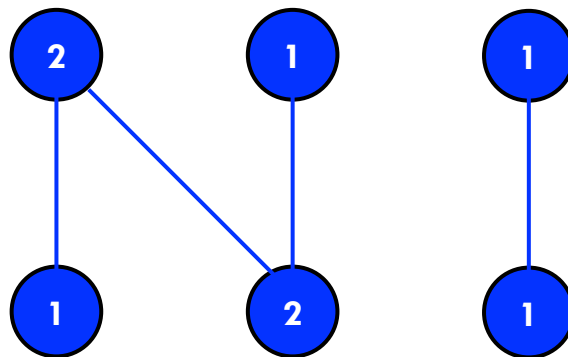
□ Adjacência:

- dois vértices A e B são adjacentes se existe uma aresta (A,B) no conjunto E
- A é “antecessor” de B, se há uma aresta (A,B), que sai de A e chega em B
 - B é “sucessor” de A



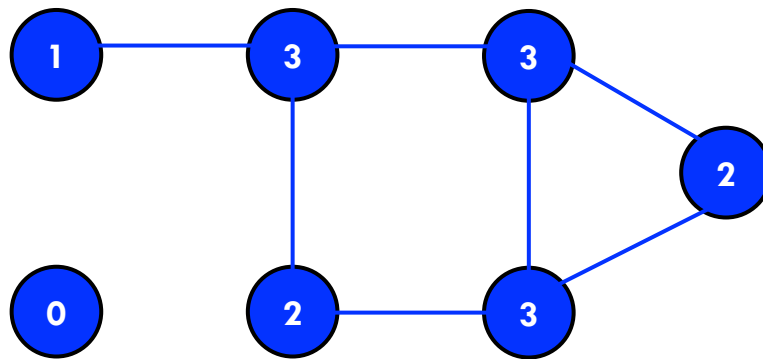
Grau de vértice

- **Grau(vértice):** é o número de arestas que incidem nele
 - grau de entrada: número de arestas que saem de V
 - grau de saída: número de arestas que chegam de C



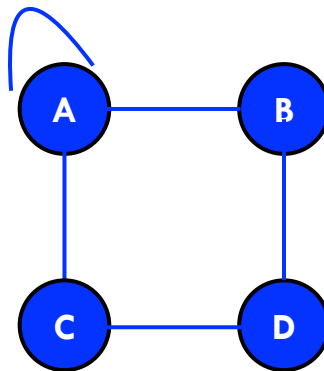
Grau de vértice

- **Grau(vértice):** é o número de arestas que incidem nele
 - grau de entrada: número de arestas que saem de V
 - grau de saída: número de arestas que chegam de C



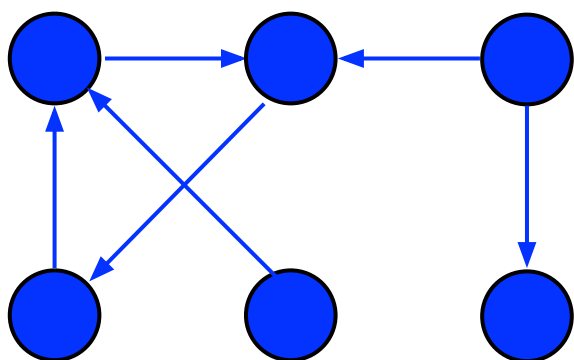
Laço

- **Laço:** é uma aresta ligando um vértice a ele próprio
 - (A, A)

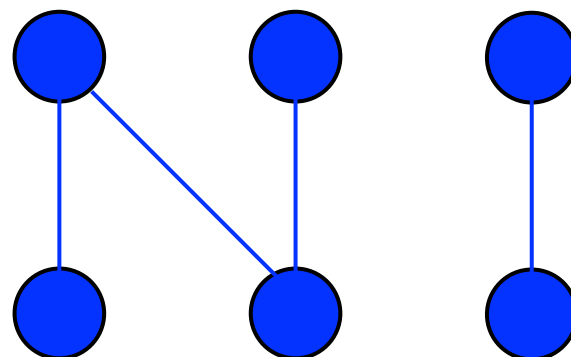


Conectividade

- Um grafo é **conexo** se existe uma sequência de arestas adjacentes que ligam todos os pares de vértices do grafo.
 - Caso contrário, o grafo é dito **desconexo**



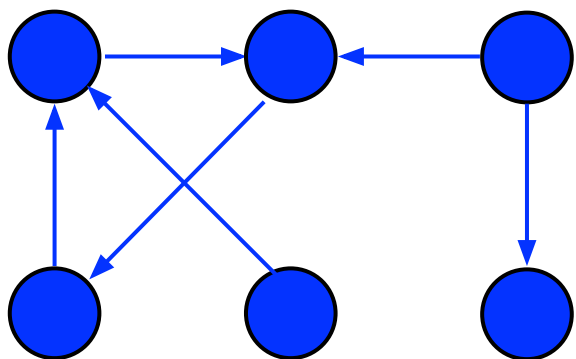
Grafo conexo



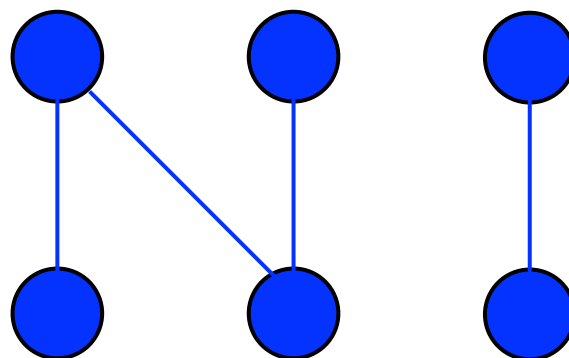
Grafo desconexo

Conectividade

- Uma **componente conexa** de um grafo desconexo é um subgrafo conexo do grafo
- Uma **ponte** é uma aresta que, se retirada, torna desconexo um grafo conexo



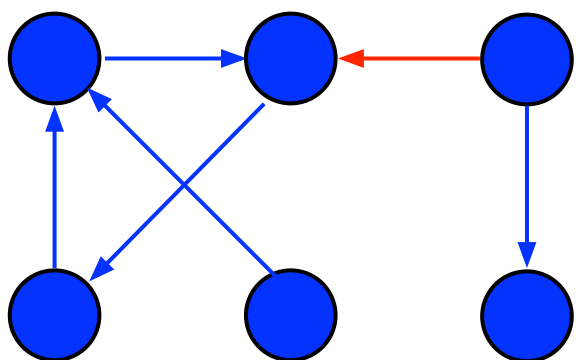
Grafo conexo



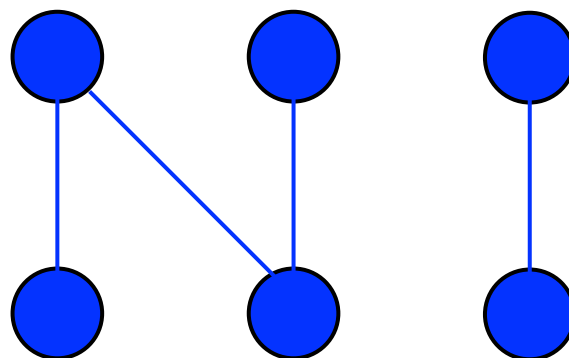
Grafo desconexo

Conectividade

- Uma **componente conexa** de um grafo desconexo é um subgrafo conexo do grafo
- Uma **ponte** é uma aresta que, se retirada, torna desconexo um grafo conexo



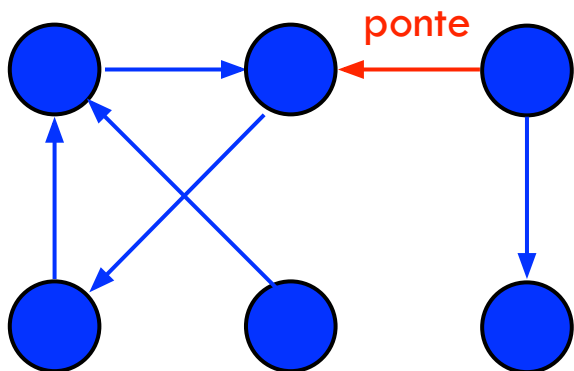
Grafo conexo



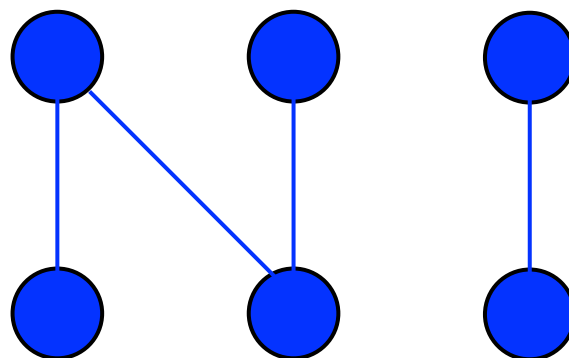
Grafo desconexo

Conectividade

- Uma **componente conexa** de um grafo desconexo é um subgrafo conexo do grafo
- Uma **ponte** é uma aresta que, se retirada, torna desconexo um grafo conexo



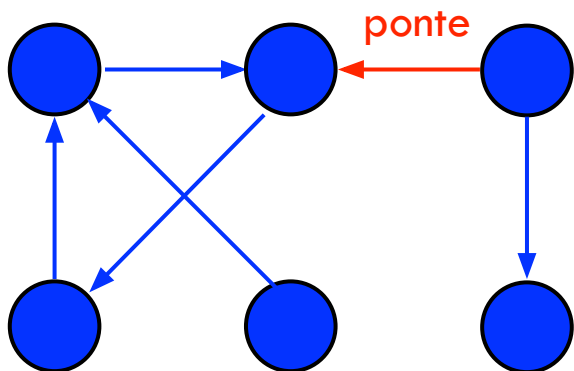
Grafo conexo



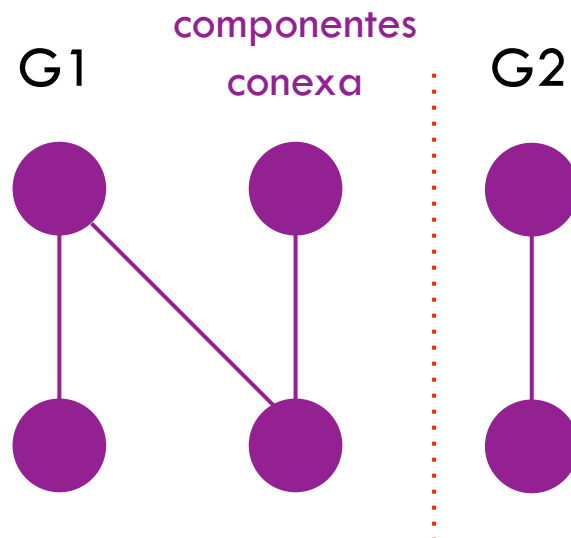
Grafo desconexo

Conectividade

- Uma **componente conexa** de um grafo desconexo é um subgrafo conexo do grafo
- Uma **ponte** é uma aresta que, se retirada, torna desconexo um grafo conexo



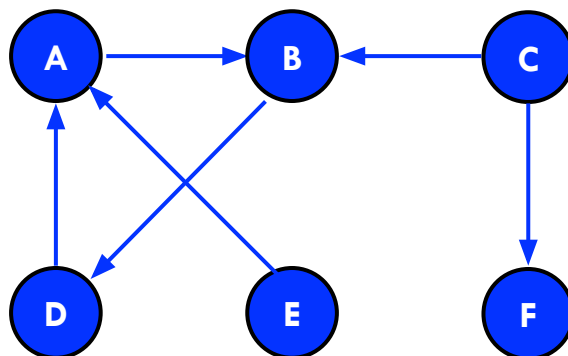
Grafo conexo



Grafo desconexo

Caminhos

- Um **caminho** é uma sequência de vértices ligados por arestas do grafo



C1: {A, B, D}

C2: {C, B}

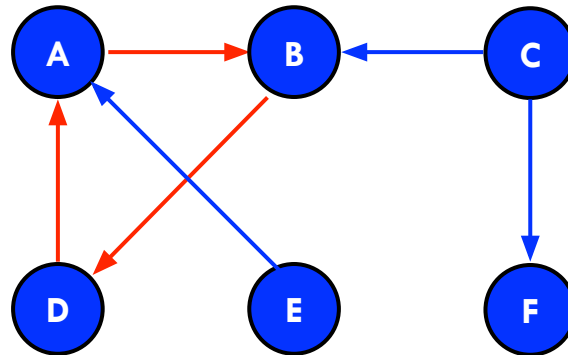
C3: {C, F}

C4: {E, A, B, D}

...

Ciclo

- Um **ciclo** é uma sequência de vértices ligados por arestas do grafo

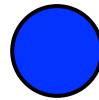


Ciclo: {A, B, D, A}

Grafos completos

- Um grafo é **completo** se possui uma aresta para cada par de vértices

K0



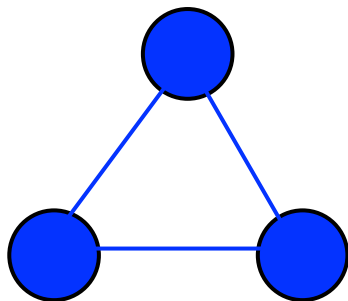
K1



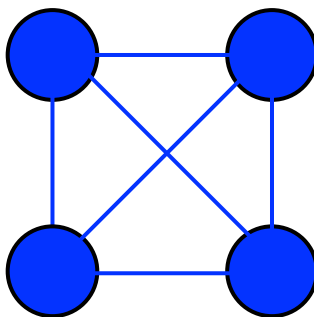
K2

Grafos completos

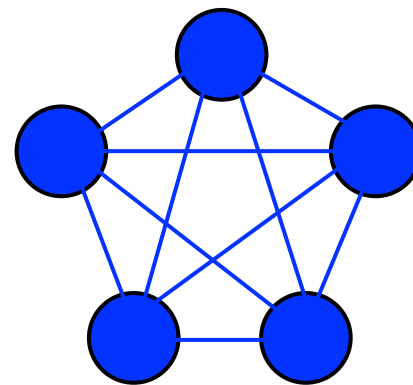
- Um grafo é **completo** se possui uma aresta para cada par de vértices



K3



K4



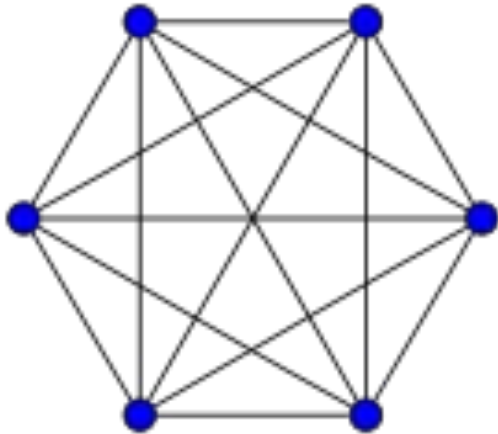
K5

Grafos completos

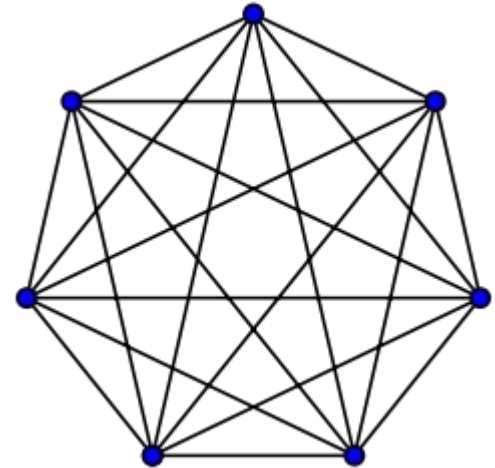
- $K_6, K_7?$

Grafos completos

- $K_6, K_7?$



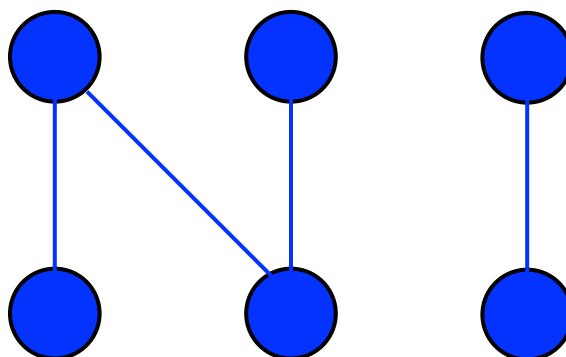
K_6



K_7

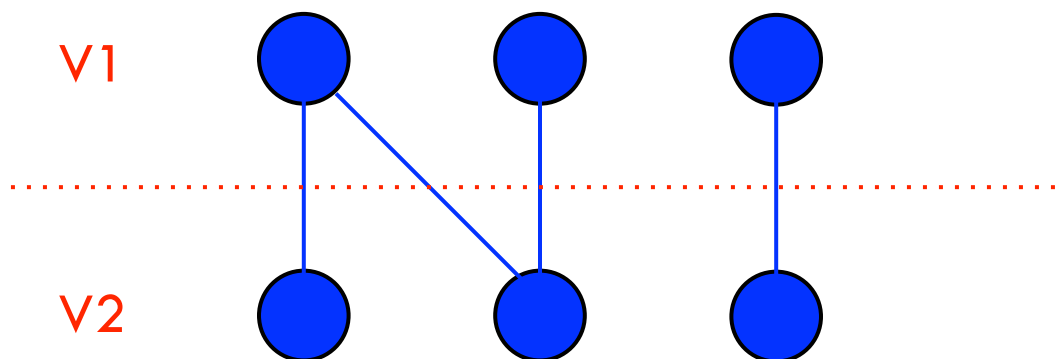
Grafos bipartido

- Um grafo é dito ser **bipartido** quando o conjunto de vértices pode ser particionado em dois conjuntos $V1$ e $V2$, tais que toda aresta do grafo liga um vértice de $V1$ a um vértice de $V2$



Grafos bipartido

- Um grafo é dito ser **bipartido** quando o conjunto de vértices pode ser particionado em dois conjuntos $V1$ e $V2$, tais que toda aresta do grafo liga um vértice de $V1$ a um vértice de $V2$



Roteiro



- 1 Introdução
- 2 Grafos
- 3 Definições
- 4 Representações
- 5 Referências

Representações de grafos

- Lista de adjacência
- Matriz de adjacência

Representações de grafos

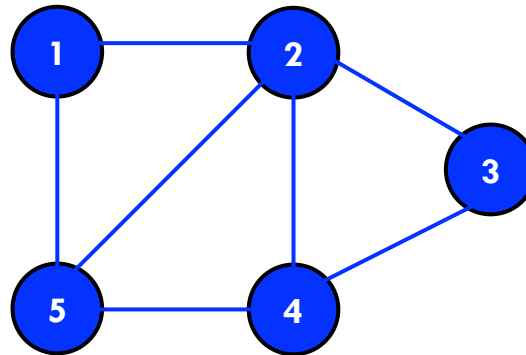
- Matriz de Adjacência
 - forma mais simples
- Propriedades:
 - representa grafo sem ambiguidade
 - é simétrica para grafo não direcionado
 - Armazenamento: $O(n^2)$

Representações de grafos

- Lista de Adjacência
 - forma encadeada
- Propriedades:
 - conjunto de listas para cada vértice
 - vetor de listas lineares

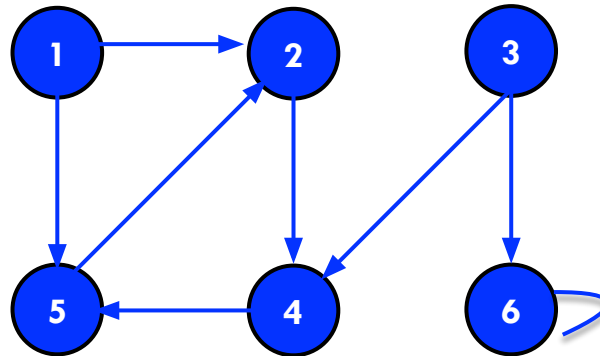
Exemplo 01

- Qual a lista e matriz de adjacência do grafo abaixo?



Exemplo 02

- Qual a lista e matriz de adjacência do grafo abaixo?



Algoritmos de grafos

- **Percursos**

- Busca em largura (Breath-First Search - BFS)
- Busca em profundidade (Depth-First Search - DFS)

- **Ordenação Topológica** (Topological sort)

- **Caminhos mínimos** (Shortest paths)

- Dijkstra

- **Árvore geradora mínima** (Minimum Spanning Trees - MST)

- Prim
- Kruskal

- **Fluxo máximo** (Maximum flow)

Algoritmos de grafos

- **Percursos**

- Busca em largura (Breath-First Search - BFS)
- Busca em profundidade (Depth-First Search - DFS)

- **Ordenação Topológica** (Topological sort)

- **Caminhos mínimos** (Shortest paths)

- Dijkstra

- **Árvore geradora mínima** (Minimum Spanning Trees - MST)

- Prim
- Kruskal

- **Fluxo máximo** (Maximum flow)

Algoritmos de grafos

- **Percursos**

- Busca em largura (Breath-First Search - BFS)
- Busca em profundidade (Depth-First Search - DFS)

- **Ordenação Topológica** (Topological sort)

- **Caminhos mínimos** (Shortest paths)

- Dijkstra

- **Árvore geradora mínima** (Minimum Spanning Trees - MST)

- Prim
- Kruskal

- **Fluxo máximo** (Maximum flow)

Algoritmos de grafos

- **Percursos**

- Busca em largura (Breath-First Search - BFS)
- Busca em profundidade (Depth-First Search - DFS)

- **Ordenação Topológica** (Topological sort)

- **Caminhos mínimos** (Shortest paths)

- Dijkstra

- **Árvore geradora mínima** (Minimum Spanning Trees - MST)

- Prim
- Kruskal

- **Fluxo máximo** (Maximum flow)

Algoritmos de grafos

- **Percursos**

- Busca em largura (Breath-First Search - BFS)
- Busca em profundidade (Depth-First Search - DFS)

- **Ordenação Topológica** (Topological sort)

- **Caminhos mínimos** (Shortest paths)

- Dijkstra

- **Árvore geradora mínima** (Minimum Spanning Trees - MST)

- Prim
- Kruskal

- **Fluxo máximo** (Maximum flow)

Algoritmos de grafos

- **Percursos**

- Busca em largura (Breath-First Search - BFS)
- Busca em profundidade (Depth-First Search - DFS)

- **Ordenação Topológica** (Topological sort)

- **Caminhos mínimos** (Shortest paths)

- Dijkstra

- **Árvore geradora mínima** (Minimum Spanning Trees - MST)

- Prim
- Kruskal

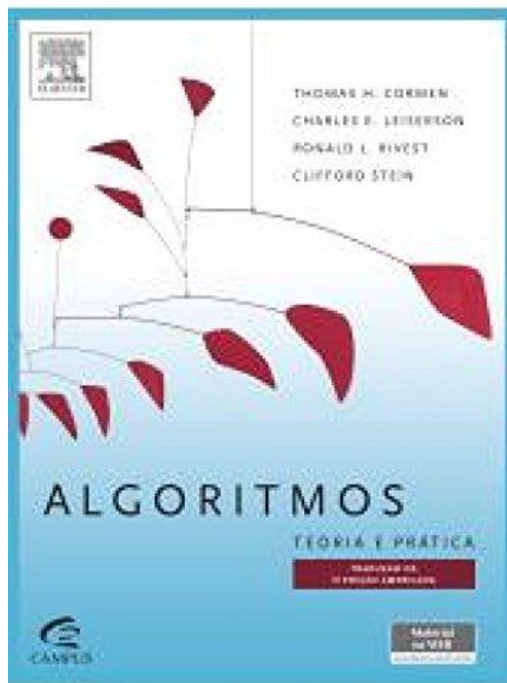
- **Fluxo máximo** (Maximum flow)

Roteiro

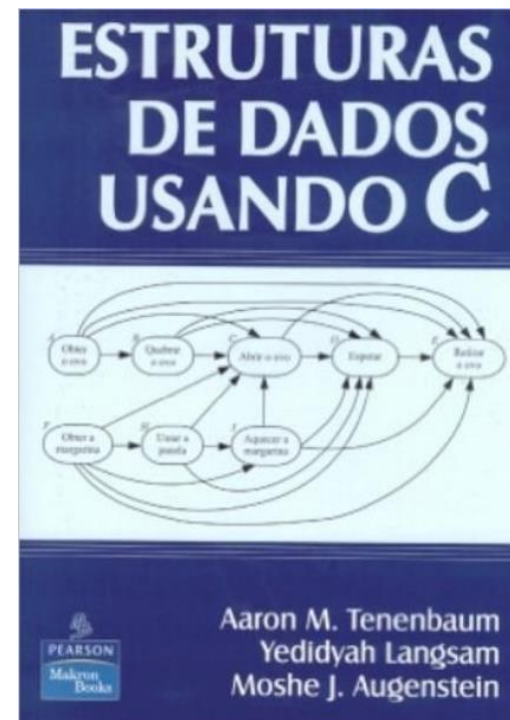


- 1** Introdução
- 2** Grafos
- 3** Definições
- 4** Representações
- 5** Referências

Referências sugeridas

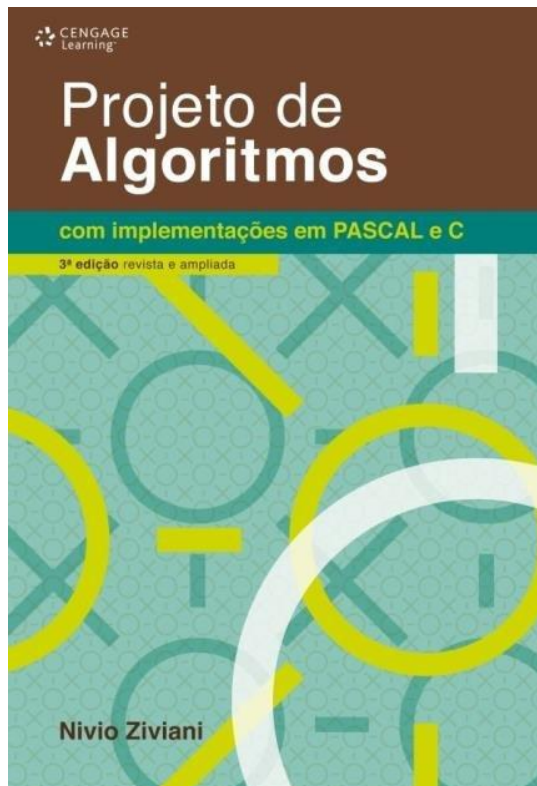


[Cormen et al, 2018]



[Tenenbaum et al, 1995]

Referências sugeridas



[Ziviani, 2010]



[Drozdek, 2017]



Perguntas?