



Programação Avançada

Introdução ao TAD –Graph
Noções de Grafos
Programação Avançada – 2020-21

Bruno Silva, Patrícia Macedo

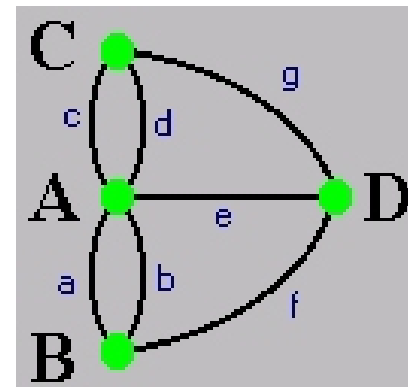
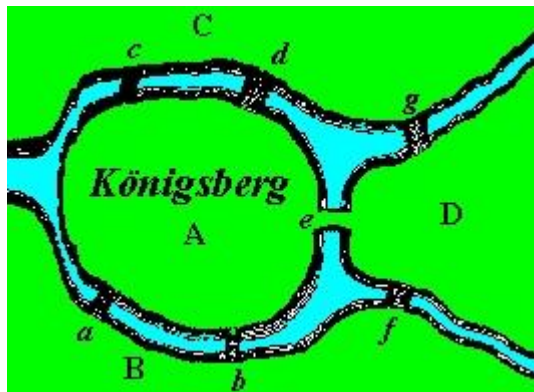
Sumário



- Contexto Histórico
- Aplicações
- Definições
 - Grafo
 - Dígrafo
 - Ordem, adjacência e grau
 - Tipos de grafos
- TAD Graph
 - Estruturas de Dados para implementar os grafos

Contexto Histórico

- Problema das Pontes de Königsberg
 - No século 18, na cidade de Königsberg (antiga Prússia), um conjunto de sete pontes cruzavam o rio Pregel. Elas conectavam duas ilhas entre si (A e D) e as ilhas com as margens (C e B).
 - Os habitantes perguntavam: É possível cruzar as sete pontes numa caminhada contínua sem passar duas vezes por qualquer uma delas?
 - Em 1736, Euler apresenta a solução deste problema na Academia de São Petersburgo, sendo este o primeiro trabalho sobre **Grafos**.



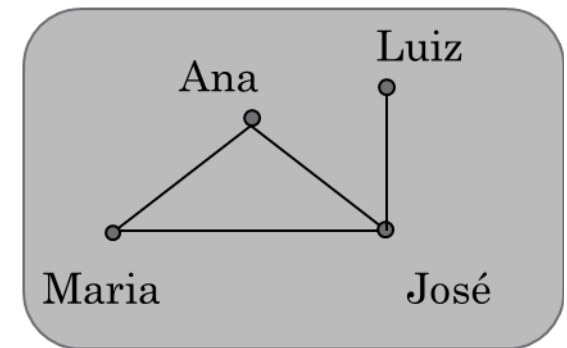
Definição formal de Grafo

- **Grafo (Graph)**

- Um grafo $G(V, A)$ é definido pelos conjuntos V e A , onde:
 - V é um conjunto não vazio: Vértices, Nodos ou Nós do grafo (**vertex**)
 - A é um conjunto de pares ordenados $a=(v,w)$ com v e w pertencente a V : Arestas, Linhas ou Ramos do grafo (**edge**).

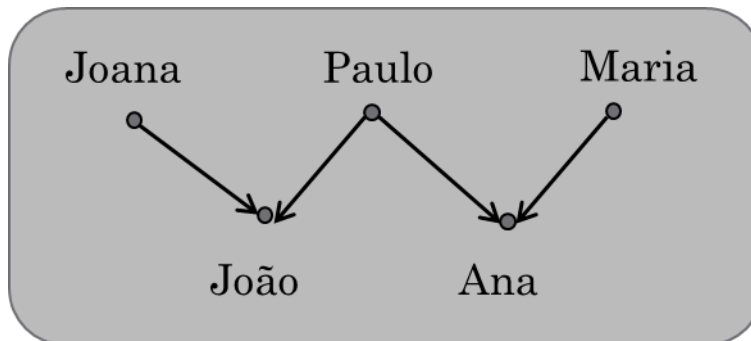
- **Exemplo**

- $V = \{p \mid p \text{ é uma pessoa}\}$
- $A = \{(v,w) \mid v \text{ é amiga de } w\}$
 - $V = \{\text{Maria, José, Ana, Luiz}\}$
 - $A = \{(\text{Maria, José}), (\text{Maria, Ana}), (\text{José, Luiz}), (\text{José, Ana})\}$



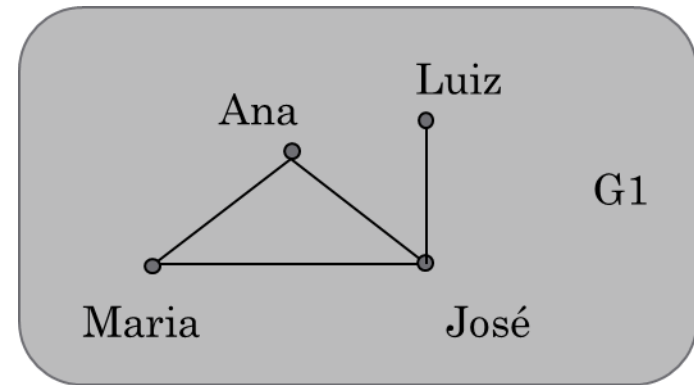
Definição de Dígrafo (grafo orientado)

- Dígrafo
 - Grafo orientado
- Exemplo
 - $V = \{p \mid p \text{ é uma pessoa}\}$
 - $A = \{(v,w) \mid v \text{ é pai ou mãe de } w\}$



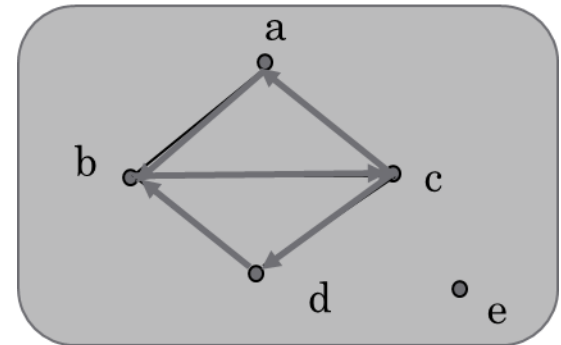
Ordem e Adjacência

- Ordem
 - É o número de vértices do grafo
 - $\text{Ordem}(G1) = 4$
- Adjacência
 - Dois vértices - v e w de um grafo - são adjacentes se há uma aresta $a=(v,w)$ em G
 - Ex: José e Luiz em $G1$
 - Duas arestas são adjacentes se incidem sobre o mesmo vértice
 - Ex: $(\text{Ana}, \text{Maria})$ e $(\text{Ana}, \text{José})$ em $G1$



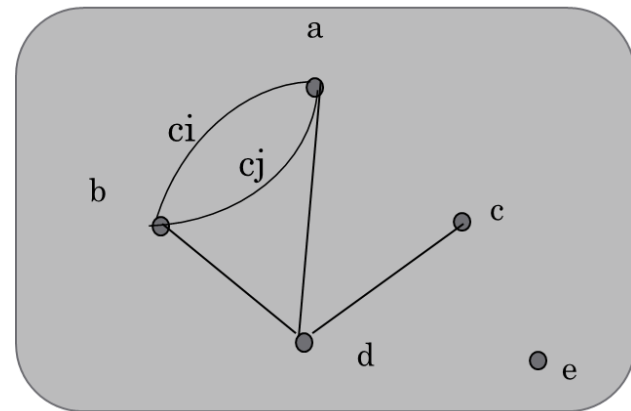
Graus

- Grau de um vértice
 - É o número de arestas incidentes no vértice
 - $\text{Grau}(b)=3$
- Grau de saída (*outdegree*)
 - Número de arestas que têm ponta inicial no vértice
 - $\text{GrauSaída}(b) = 1$
- Grau de entrada (*indegree*)
 - Número de arestas têm ponta final no vértice
 - $\text{GrauEntrada}(b) = 2$



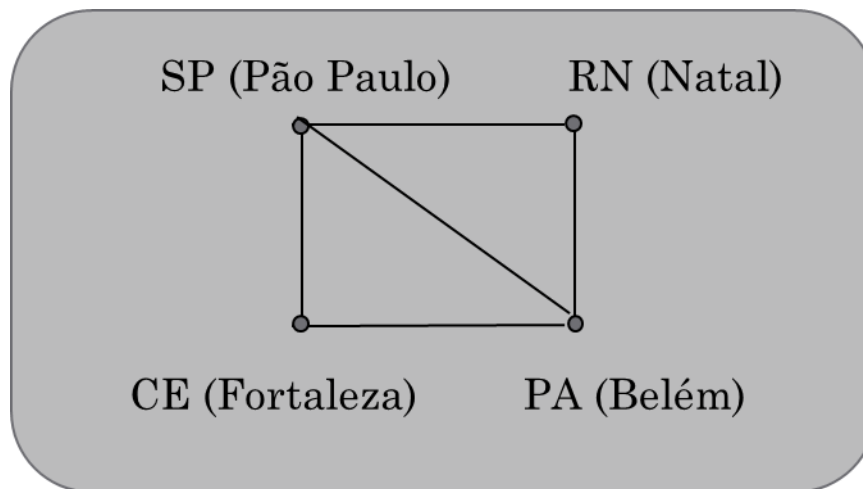
Vértice Isolado, Arestas Paralelas

- Vértice isolado
 - É aquele que possui grau igual a zero
 - Ex: Vértice e
- Arestas paralelas
 - Possuem os mesmos vértices terminais
 - Exemplo $ci=(a,b)$ e $cj=(a,b)$



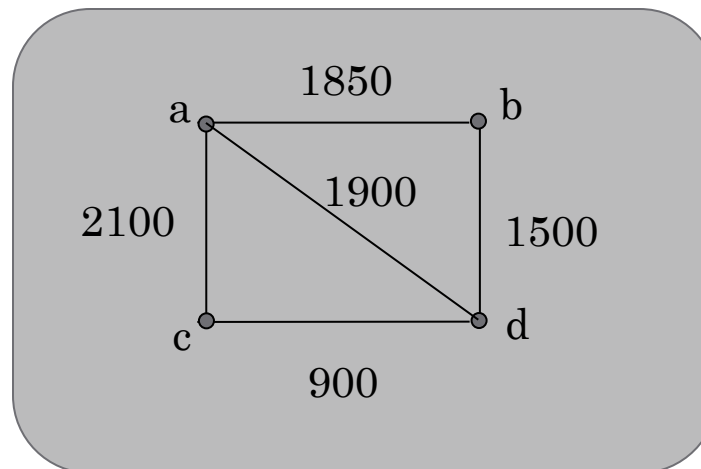
Grafo Rotulado

- Grafo rotulado
 - Grafo em que cada vértice está associado a um rótulo



Grafo Valorado

- Grafo valorado
 - Um grafo $G(V, A)$ é denominado **valorado** quando cada aresta tem associado um valor.



Algumas áreas de aplicação dos Grafos

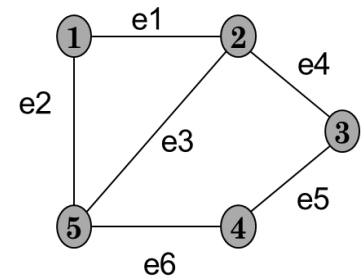
GRAPH	VERTICES	EDGES
circulatory	organ	blood vessel
skeletal	joint	bone
nervous	neuron	synapse
social	person	relationship
epidemiological	person	infection
chemical	molecule	bond
n-body	particle	force
genetic	gene	mutation
biochemical	protein	interaction
transportation	intersection	road
Internet	computer	cable
Web	web page	link
distribution	power station	power line
mechanical	joint	beam
software	module	call
financial	account	transaction

TAD Graph| Exercícios(1)



Para o grafo da imagem indica quais as frases verdadeiras

- A ordem do grafo é 3
- Os vértice com maior grau são os vértices 5 e 2
- As arestas incidentes ao vertice2 são as arestas e1 e e4
- O vértice oposto ao vértice 2 na aresta e3 é o vértice 5
- O vértice 1 e 5 são adjacentes



Desenha um grafo que obedeça aos seguintes critérios

- É um digrafo
- Tem ordem 4
- Tem um vértice isolado
- Tem duas arestas paralelas.
- O vértice com maior grau de saída tem grau 4
- O vértice com maior grau de entrada tem grau 2

TAD Graph - Especificação

Um grafo $G(V, E)$ é definido pelos conjuntos V e E , onde:

- V é um conjunto não vazio: Vértices.
- E é um conjunto de pares ordenados $e=(v,w)$ com v e w pertencente a V : arestas (edges).

As operações modificadoras do TAD Graph são:

- **insertVertex(v)**: insere v como sendo vértice do grafo.
- **insertEdge(u, v, e)**: insere uma aresta x entre os vértices u e v ; devolve erro se v e x não correspondem a vértices do grafo
- **removeVertex(v)**: remove o vértice v e todas as suas arestas adjacentas, devolve erro se v não existir no grafo.
- **removeEdge(e)**: remove a aresta e , devolve erro se e não existir no grafo.

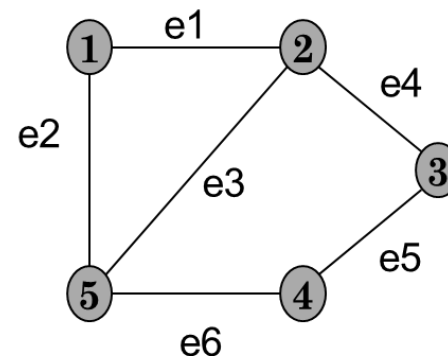
TAD Graph – Especificação (cont)

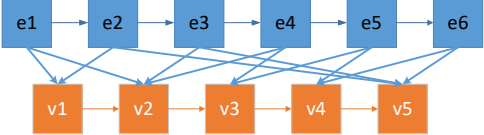
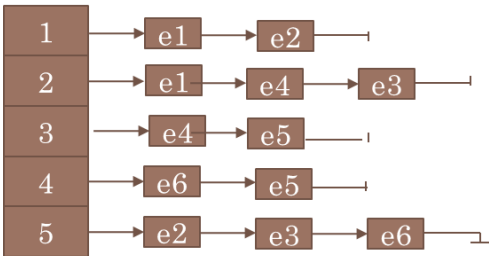
As operações inspetoras (devolvem informação sobre o estado do grafo):

- **numvertices()**: devolve o número de vértices
- **numEdges()**: devolve o número de arestas
- **edges()** : devolve uma coleção iterável das arestas do grafo.
- **vertices()** : devolve uma coleção iterável dos vértices do grafo.
- **opposite(v, e)**: devolve o vértice da aresta e oposto ao vértice v, devolve erro se v ou e não existem no grafo, ou se v não é vértice da aresta e.
- **degree(v)**: devolve o grau do vértice v, devolve erro se v não existe no grafo.
- **incidentEdges(v)**: devolve uma coleção iterável das arestas incidentes ao vértice v, devolve erro se v não existe no grafo.
- **areAdjacent(v,w)**: devolve um valor lógico (true/false) que indica se os vértices v e w são adjacentes , devolve erro se v ou w não existirem como vértices do grafo.

Estruturas de Dados para implementar o TAD Graph

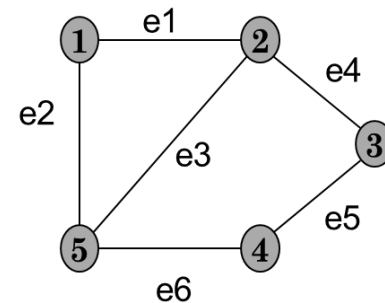
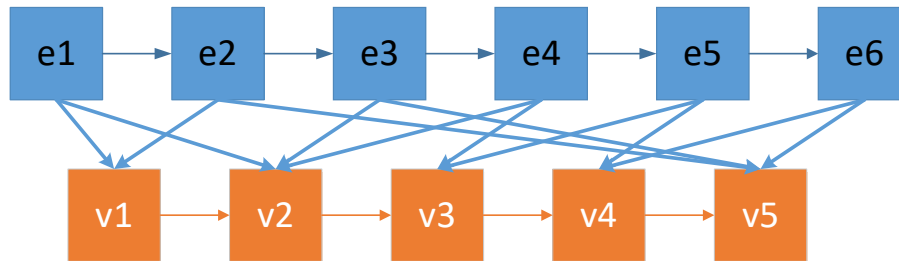
- Existem várias estruturas de dados possíveis para implementar o grafo TAD.
- A seleção da estrutura de dados influencia diretamente a complexidade algorítmica das operações.



Lista de Arestas	Lista de Adjacências	Matriz de Adjacências																																				
		<table><tr><th></th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr><tr><th>1</th><td></td><td>e1</td><td></td><td></td><td>e2</td></tr><tr><th>2</th><td>e1</td><td></td><td>e4</td><td></td><td>e3</td></tr><tr><th>3</th><td></td><td>e4</td><td></td><td>e5</td><td></td></tr><tr><th>4</th><td></td><td></td><td>e5</td><td></td><td>e6</td></tr><tr><th>5</th><td>e2</td><td>e3</td><td></td><td>e6</td><td></td></tr></table>		1	2	3	4	5	1		e1			e2	2	e1		e4		e3	3		e4		e5		4			e5		e6	5	e2	e3		e6	
	1	2	3	4	5																																	
1		e1			e2																																	
2	e1		e4		e3																																	
3		e4		e5																																		
4			e5		e6																																	
5	e2	e3		e6																																		

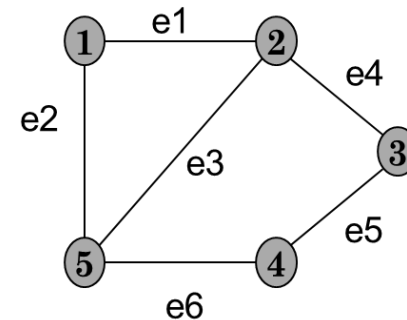
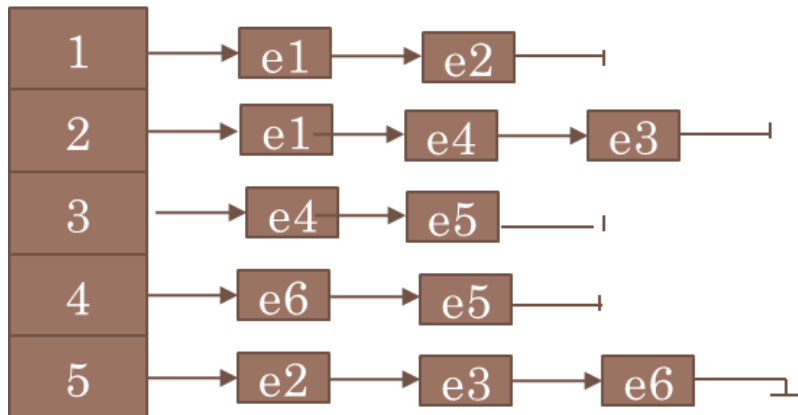
(1) Estrutura de dados - Listas de arestas

- Lista de arestas (ou dicionário-map).
- Lista de vértices (ou dicionário-map).
- Cada Aresta guarda as referências para os vértices que liga.



(2) Estrutura de dados - Listas de adjacências

- Lista de vértices (ou dicionário-map).
- Cada vértice, guarda a lista de arestas adjacentes.
- A utilização de lista ligada facilita a “ampliação” da estrutura.

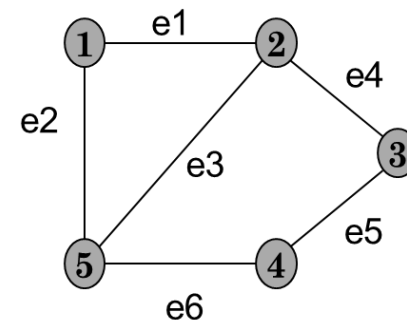


(3) Estrutura de dados - Matriz de adjacências

- Lista de vértices.
- As adjacências são representadas por uma matriz $N \times N$ onde N é o número de vértices. Espaço: $O(n^2)$.



	1	2	3	4	5
1		e1			e2
2	e1		e4		e3
3		e4		e5	
4			e5		e6
5	e2	e3		e6	

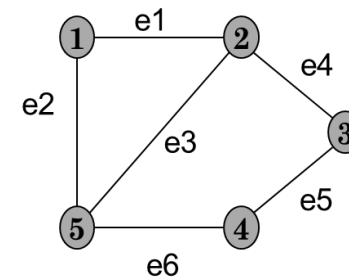


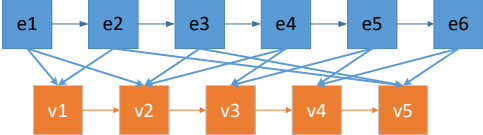
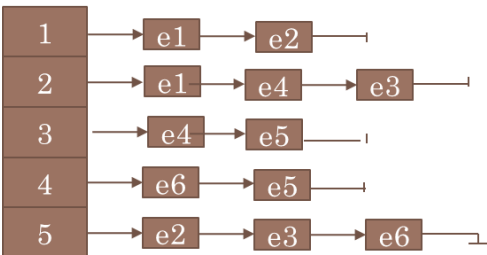
TAD Graph| Exercícios(2)



Para cada uma das estruturas de dados apresentadas complete os esquemas abaixo de forma a ilustrar como ficaria cada uma delas após efetuar as seguintes operações no grafo dado.

- adicionar o vértice 6
- remover a aresta e3
- adicionar a aresta e7 que liga o vértice 6 com o vértice 4.

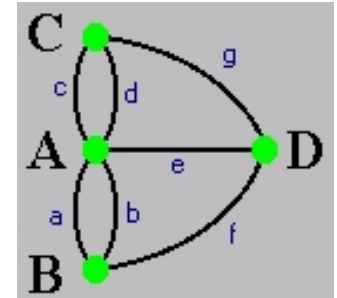


Lista de Arestas	Lista de Adjacências	Matriz de Adjacências																																				
		<table><tr><th></th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr><tr><th>1</th><td></td><td>e1</td><td></td><td></td><td>e2</td></tr><tr><th>2</th><td>e1</td><td></td><td>e4</td><td></td><td>e3</td></tr><tr><th>3</th><td></td><td>e4</td><td></td><td>e5</td><td></td></tr><tr><th>4</th><td></td><td></td><td>e5</td><td></td><td>e6</td></tr><tr><th>5</th><td>e2</td><td>e3</td><td></td><td>e6</td><td></td></tr></table>		1	2	3	4	5	1		e1			e2	2	e1		e4		e3	3		e4		e5		4			e5		e6	5	e2	e3		e6	
	1	2	3	4	5																																	
1		e1			e2																																	
2	e1		e4		e3																																	
3		e4		e5																																		
4			e5		e6																																	
5	e2	e3		e6																																		

TAD Graph| Exercícios(3)



- Para cada uma das estruturas de dados apresentadas:
 - (1) lista de arestas
 - (2) lista de adjacências
 - (3) matriz de adjacências



instancie o grafo que modela o problema das pontes de Königsberg.

- Conseguiu criar o grafo em todas as estruturas de dados ?