



[www.python.pro.br](http://www.python.pro.br)

# Roteiro - Grafos

Definição

Nomenclatura

Lista de Arcos

Lista de Adjacências

Mapa de Adjacências

Matriz de Adjacências

Análise geral

Caminho entre 2 pontos

Exercício \*

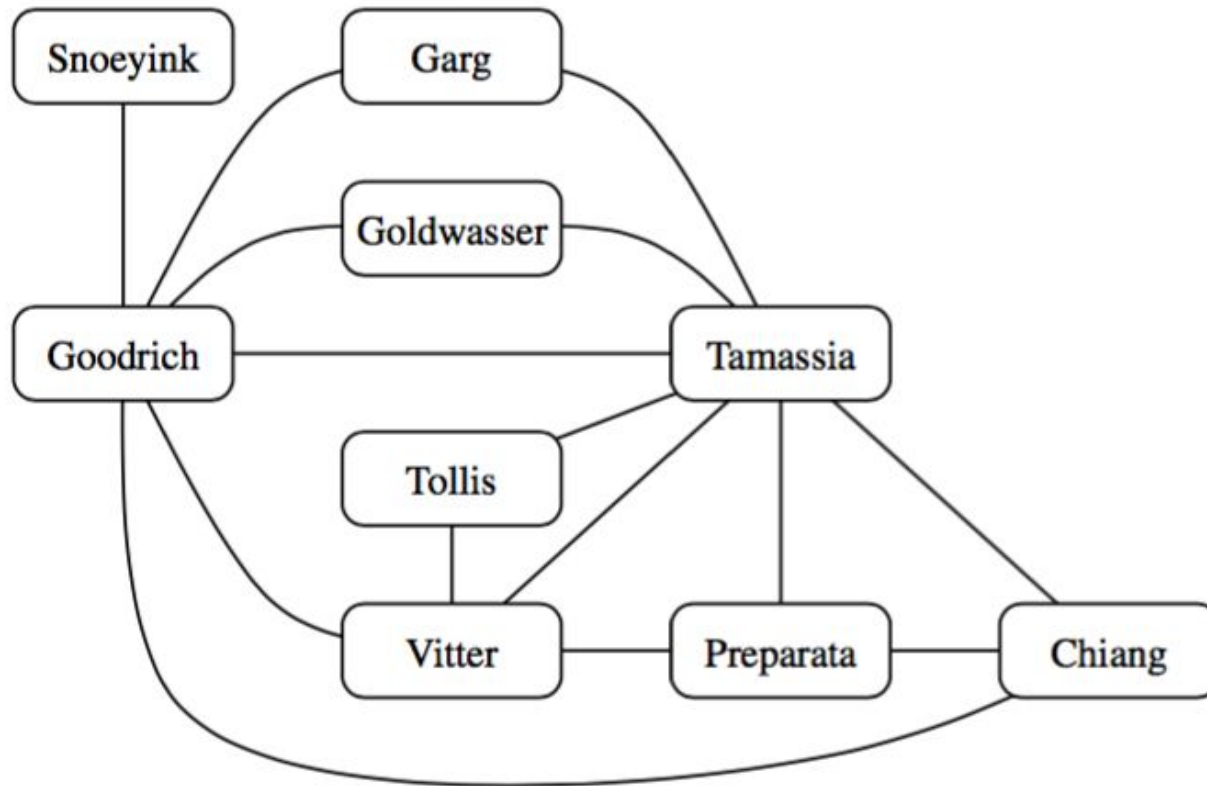
# Grafo

Estrutura de dados que contém Vértices

Cada vértice pode se conectar com outros através de arcos (edges)

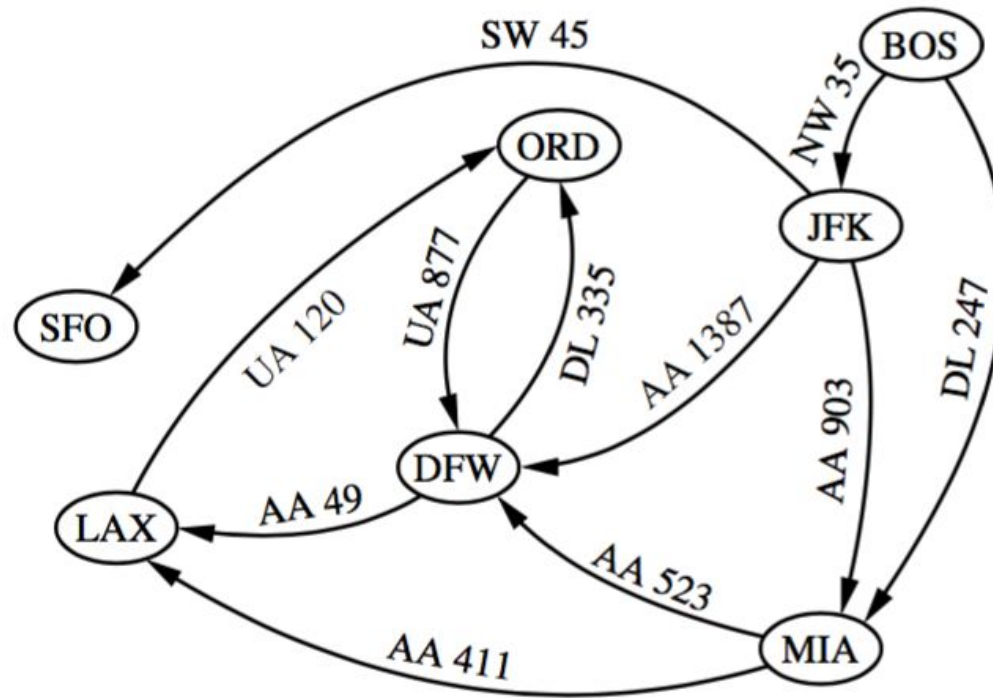
Pode ser orientado (directed) ou não - dígrafo (undirected) \*

# Grafo



**Figure 14.1:** Graph of coauthorship among some authors.

# Grafo



**Figure 14.2:** Example of a directed graph representing a flight network. The endpoints of edge UA 120 are LAX and ORD; hence, LAX and ORD are adjacent. The in-degree of DFW is 3, and the out-degree of DFW is 2.

# Nomenclatura

Adjacentes ou vizinhos: vértices conectados por arcos

Origem de um arco: vértice de onde parte um arco

Destino de um arco: vértice onde chega um arco

Vértice incidente de um arco: é origem ou destino\*

# Nomenclatura

Grau de um vértice: número de arcos incidentes

Grau de entrada: número de arcos em que vértice é destino

Grau de saída: número de arcos em que vértice é origem

Grafo Esparso: poucos arcos

Grafo Denso: muitos arcos\*

# Nomenclatura

Caminho (path): sequência de vértices, arcos, que leva de um vértice a outro

Ciclo: caminho que começa e termina no mesmo vértice

Arcos paralelos: possuem msm des e ori

Self-loop: arco cuja origem e destino são ==

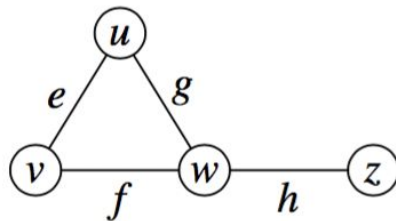
Simples: grafo sem arcos paralelos ou self-loop

\*

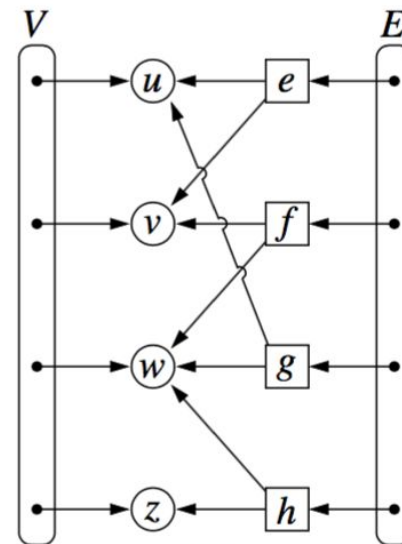


# Lista de Arcos

Mais simples, mas talvez não eficiente  
Vértices em uma lista e arcos em outra



(a)



(b)

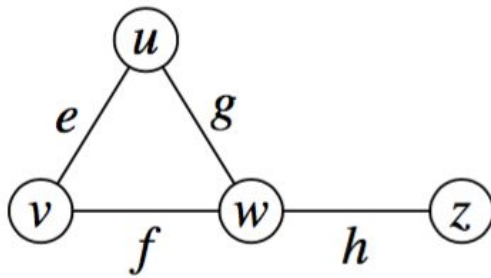
# Lista de Arcos - Análise

Operation	Running Time
<code>vertex_count()</code> , <code>edge_count()</code>	$O(1)$
<code>vertices()</code>	$O(n)$
<code>edges()</code>	$O(m)$
<code>get_edge(u,v)</code> , <code>degree(v)</code> , <code>incident_edges(v)</code>	$O(m)$
<code>insert_vertex(x)</code> , <code>insert_edge(u,v,x)</code> , <code>remove_edge(e)</code>	$O(1)$
<code>remove_vertex(v)</code>	$O(m)$

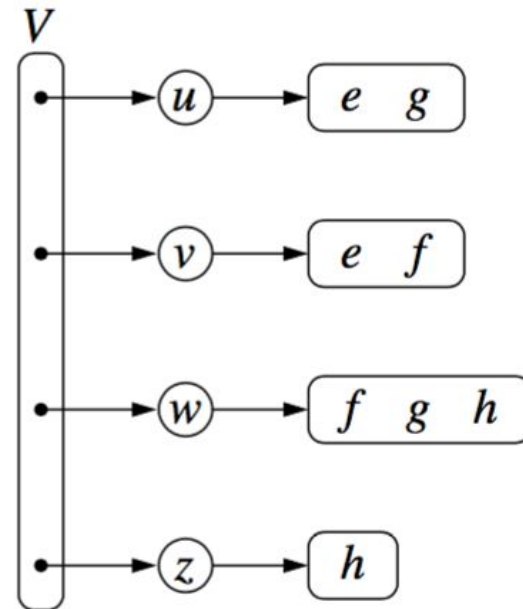
**Table 14.2:** Running times of the methods of a graph implemented with the edge list structure. The space used is  $O(n + m)$ , where  $n$  is the number of vertices and  $m$  is the number of edges.

# Lista de Adjacências

Arcos criados em listas relacionadas a vértices



(a)



(b)

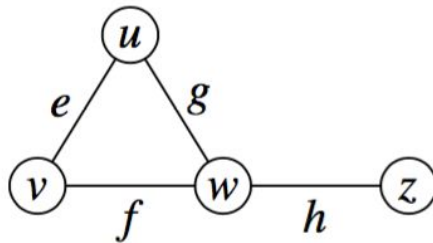
# Lista de Adjacências

Operation	Running Time
<code>vertex_count()</code> , <code>edge_count()</code>	$O(1)$
<code>vertices()</code>	$O(n)$
<code>edges()</code>	$O(m)$
<code>get_edge(u,v)</code>	$O(\min(\deg(u), \deg(v)))$
<code>degree(v)</code>	$O(1)$
<code>incident_edges(v)</code>	$O(\deg(v))$
<code>insert_vertex(x)</code> , <code>insert_edge(u,v,x)</code>	$O(1)$
<code>remove_edge(e)</code>	$O(1)$
<code>remove_vertex(v)</code>	$O(\deg(v))$

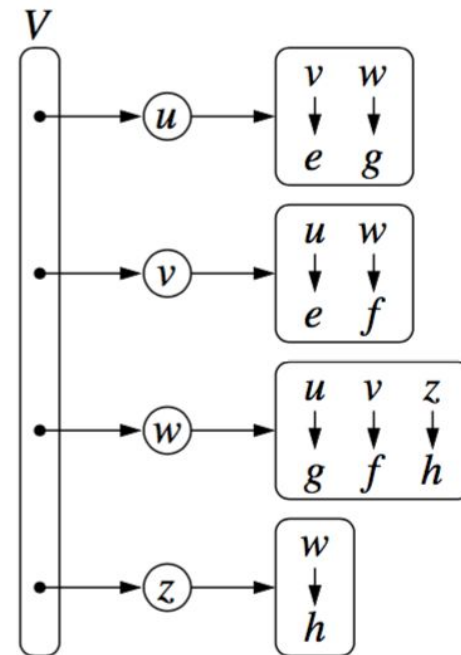
**Table 14.3:** Running times of the methods of a graph implemented with the adjacency list structure. The space used is  $O(n + m)$ , where  $n$  is the number of vertices and  $m$  is the number of edges.

# Mapa de Adjacências

Arcos em mapas relacionados com vértice oposto



(a)



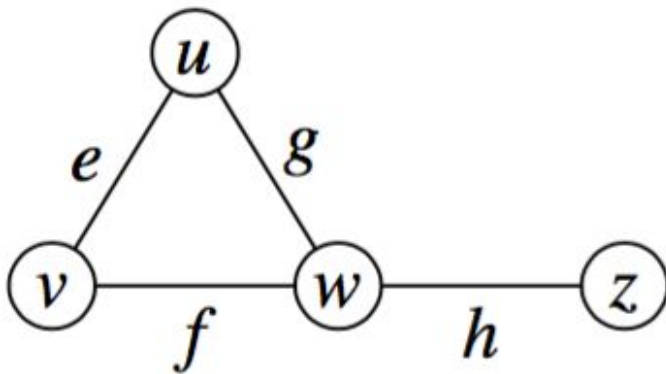
(b)

# Mapa de Adjacencias - Análise

Consegue encontrar arco( $u, v$ ) em  $O(1)$  se comparado com anterior \*

# Matriz de Adjacencias

Arcos numerados em Matriz Quadrada



(a)

		0	1	2	3	
$u$	$\longrightarrow$	0		$e$	$g$	
$v$	$\longrightarrow$	1	$e$		$f$	
$w$	$\longrightarrow$	2	$g$	$f$		$h$
$z$	$\longrightarrow$	3			$h$	

(b)

# Matriz de Adjacencias

Ocupa muito espaço para grafos esparsos

Ruim para Inserção de elementos porque tem que construir nova matriz

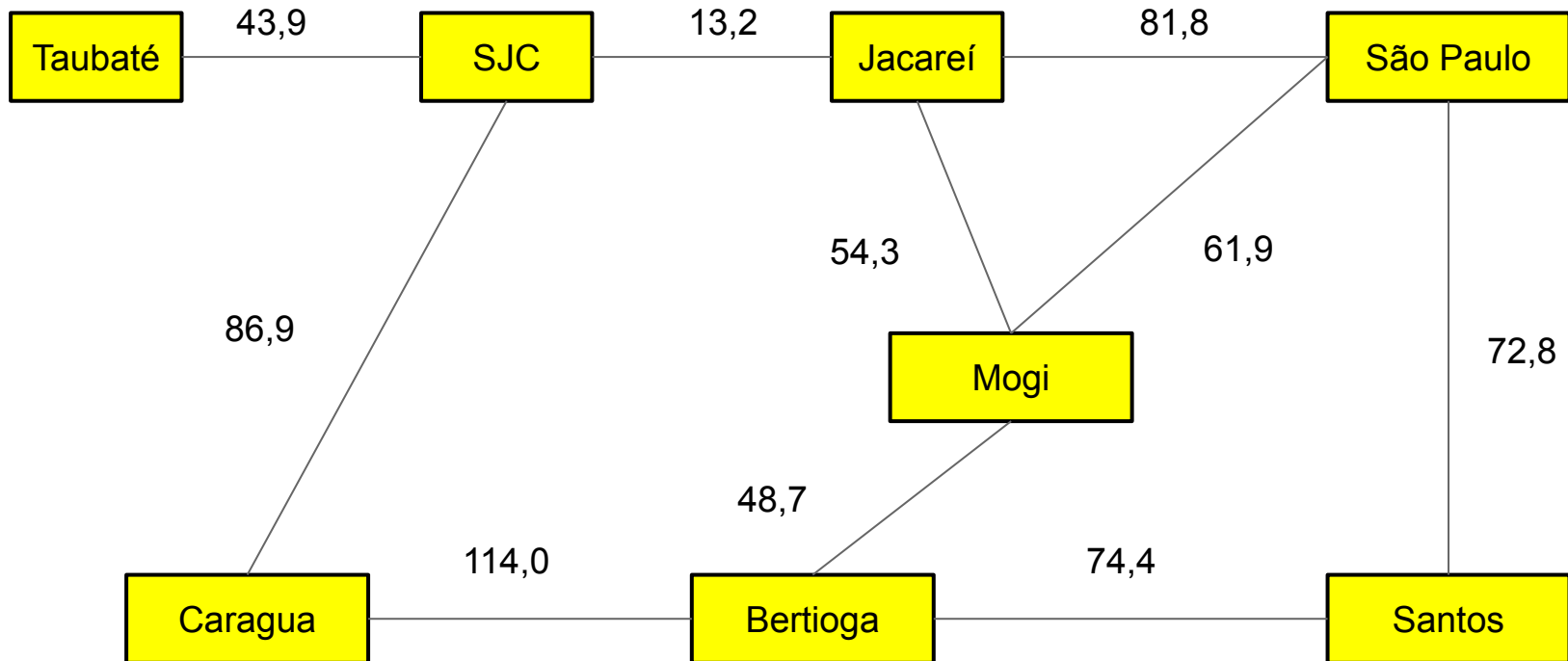
Boa para acesso\*



# Análise Geral

Operation	Edge List	Adj. List	Adj. Map	Adj. Matrix
vertex_count()	$O(1)$	$O(1)$	$O(1)$	$O(1)$
edge_count()	$O(1)$	$O(1)$	$O(1)$	$O(1)$
vertices()	$O(n)$	$O(n)$	$O(n)$	$O(n)$
edges()	$O(m)$	$O(m)$	$O(m)$	$O(m)$
get_edge(u,v)	$O(m)$	$O(\min(d_u, d_v))$	$O(1)$ exp.	$O(1)$
degree(v)	$O(m)$	$O(1)$	$O(1)$	$O(n)$
incident_edges(v)	$O(m)$	$O(d_v)$	$O(d_v)$	$O(n)$
insert_vertex(x)	$O(1)$	$O(1)$	$O(1)$	$O(n^2)$
remove_vertex(v)	$O(m)$	$O(d_v)$	$O(d_v)$	$O(n^2)$
insert_edge(u,v,x)	$O(1)$	$O(1)$	$O(1)$ exp.	$O(1)$
remove_edge(e)	$O(1)$	$O(1)$	$O(1)$ exp.	$O(1)$

# Caminho entre dois pontos



# Caminho entre dois pontos

Criar conjunto de vértices visitados

Criar pilha caminho

Incluir vértice de origem a visitados e em caminho

Adicionar arcos de origem em pilha se já não houver sido visitado \*

# Caminho entre dois pontos

Desempilhar primeiro arco

Verificar seu oposto e se ele é igual a destino

Se for, retornar caminho, se não, repetir procedimento com esse vértice \*

# Exercício

Implementar grafo não orientado

Utilizar Mapa de Adjacências

Será utilizado no próximo exercício

Slides de testes:

[https://github.com/pythonprobr/estrutura-de-dados/blob/main/aula\\_09/testes\\_grafo.py](https://github.com/pythonprobr/estrutura-de-dados/blob/main/aula_09/testes_grafo.py) \*

# Obrigado

renzo@python.pro.br  
@renzoprobr

