pythonpro

Python para profisionais

www.python.pro.br

Email: renzo@python.pro.br **Twitter**: @renzoprobr

Roteiro - Grafos

Definição

Nomenclatura

*

Email: renzo@python.pro.br **Twitter**: @renzoprobr

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

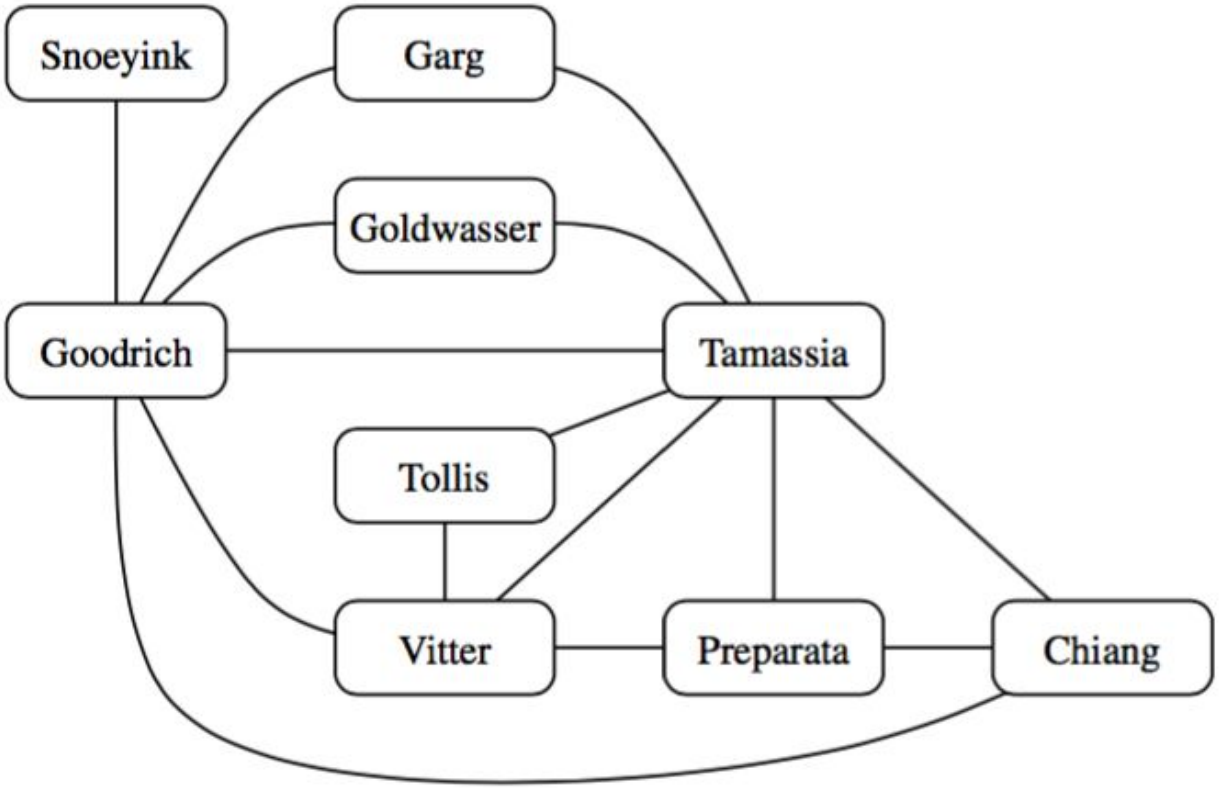


Figure 14.1: Graph of coauthorship among some authors.

Email: renzo@python.pro.br **Twitter**: @renzoprobr

pythonpro

Grafo

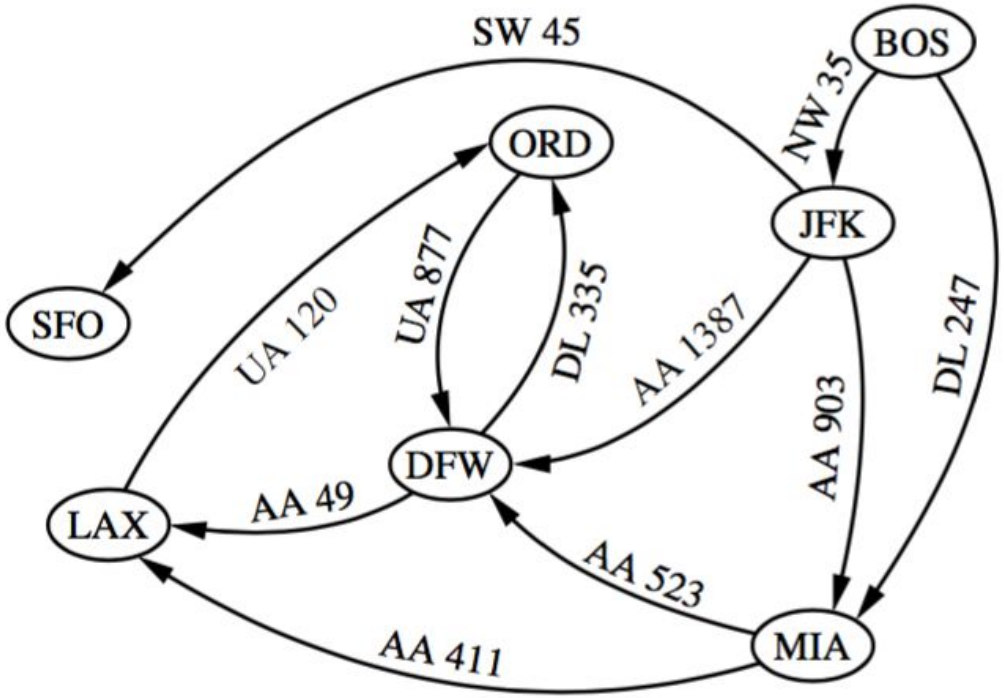


Figure 14.2: Example of a directed graph representing a flight network. The endpoints of edge UA 120 are LAX and ORD; hence, LAX and ORD are adjacent. The in-degree of DFW is 3, and the out-degree of DFW is 2.

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*

Email: renzo@python.pro.br **Twitter**: @renzoprobr

pythonpro

Nomenclatura

Grau de um vértice: número de arcos incidentes

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

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

Grafo Esparso: poucos arcos

Grafo Denso: muitos arcos*

Email: renzo@python.pro.br **Twitter**: @renzoprobr

Nomenclatura

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

Ciclo: caminho que começa e termina no mésmo vértice

Arcos paralelos: possuem msm des e ori

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

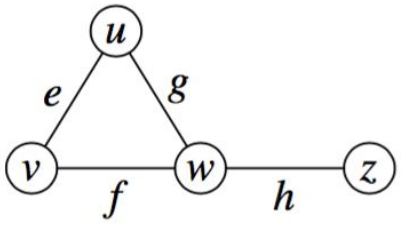
Simples: grafo sem arcos paralelos ou self-loop *

Email: renzo@python.pro.br **Twitter**: @renzoprobr

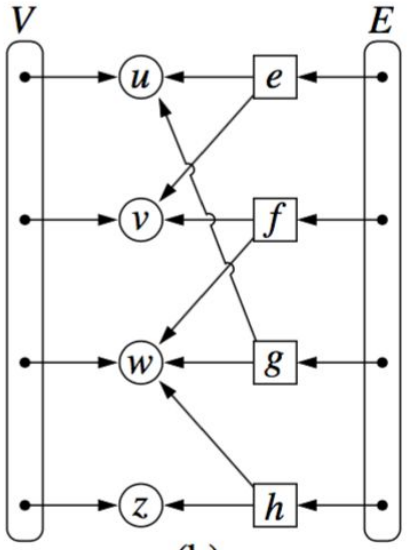
Lista de Arcos

Mais simples, mas talvez não eficiente

Vértices em uma lista e arcos em outra



(a)



(b)

Email: renzo@python.pro.br **Twitter**: @renzoprobr

pythonpro

Lista de Arcos - Análise

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

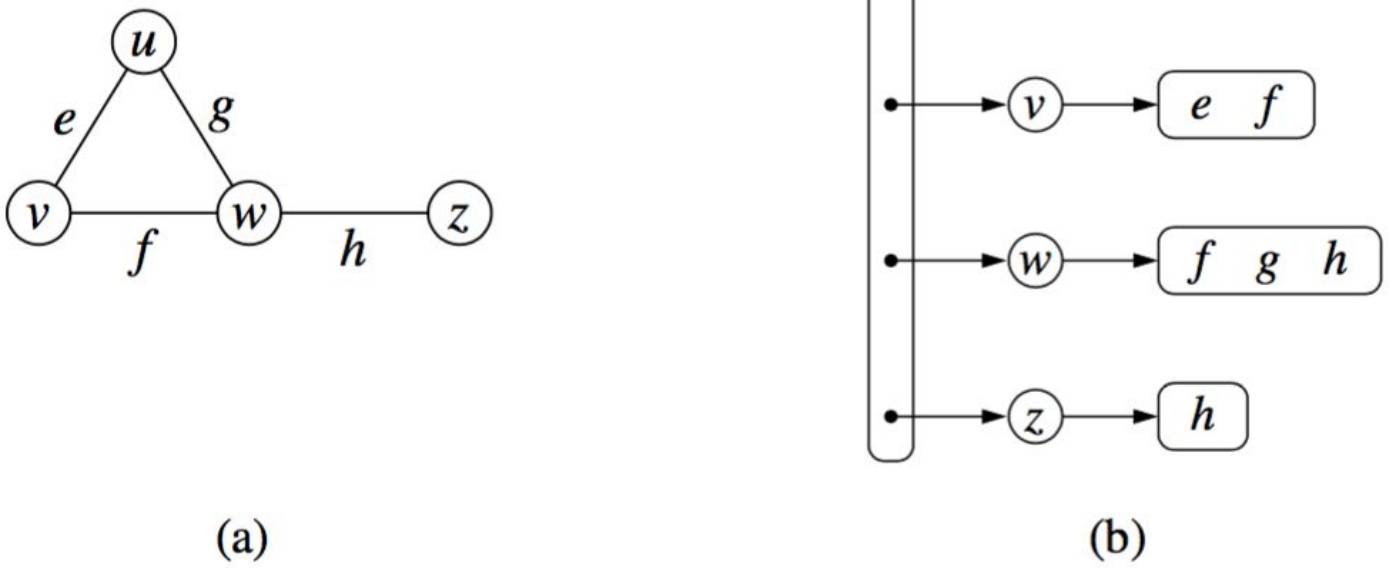
Table 14.2: Running times of the methods of a graph implemented with the edge list structure. The space used is $O(n+m)$, where n is the number of vertices and m is the number of edges.

Email: renzo@python.pro.br **Twitter**: @renzoprobr

pythonpro

Lista de Adjacencias

Arcos criados em listas relacionadas a vértices



Email: renzo@python.pro.br **Twitter**: @renzoprobr

pythonpro

Lista de Adjacências

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

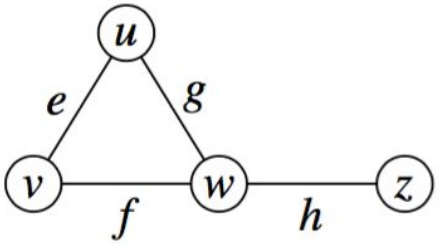
Table 14.3: Running times of the methods of a graph implemented with the adjacency list structure. The space used is $O(n+m)$, where n is the number of vertices and m is the number of edges.

Email: renzo@python.pro.br **Twitter**: @renzoprobr

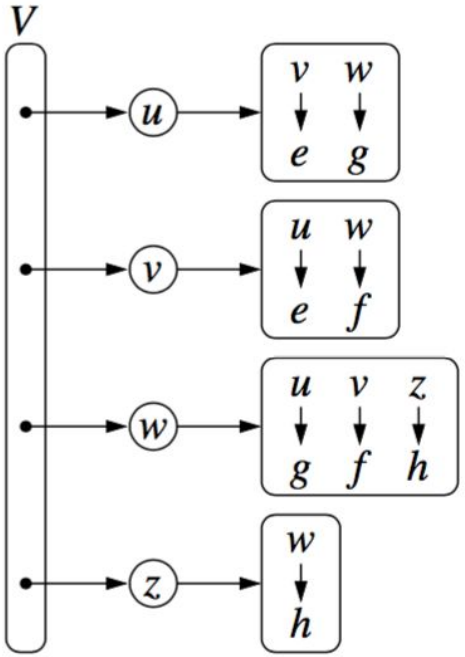
pythonpro

Mapa de Adjacencias

Arcos em mapas relacionados com vértice oposto



(a)



(b)

Email: renzo@python.pro.br **Twitter**: @renzoprobr

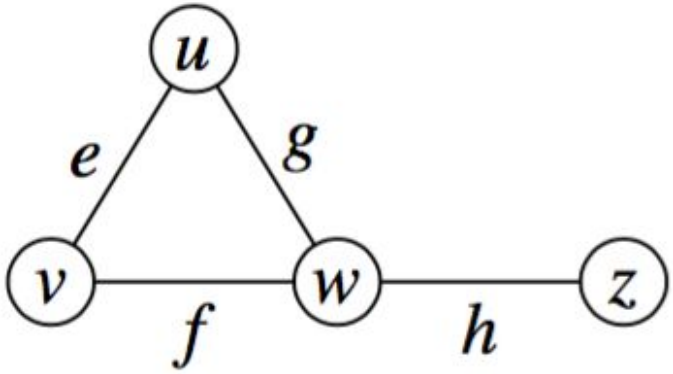
pythonpro

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	0		e	g	
v	1	e		f	
w	2	g	f		h
z	3			h	

(b)

Email: renzo@python.pro.br **Twitter**: @renzoprobr

pythonpro

Matriz de Adjacencias

Ocupa muito espaço para grafos esparsos

Ruim para Inserção de elementos porque tem que construir nova matriz

Boa para acesso*

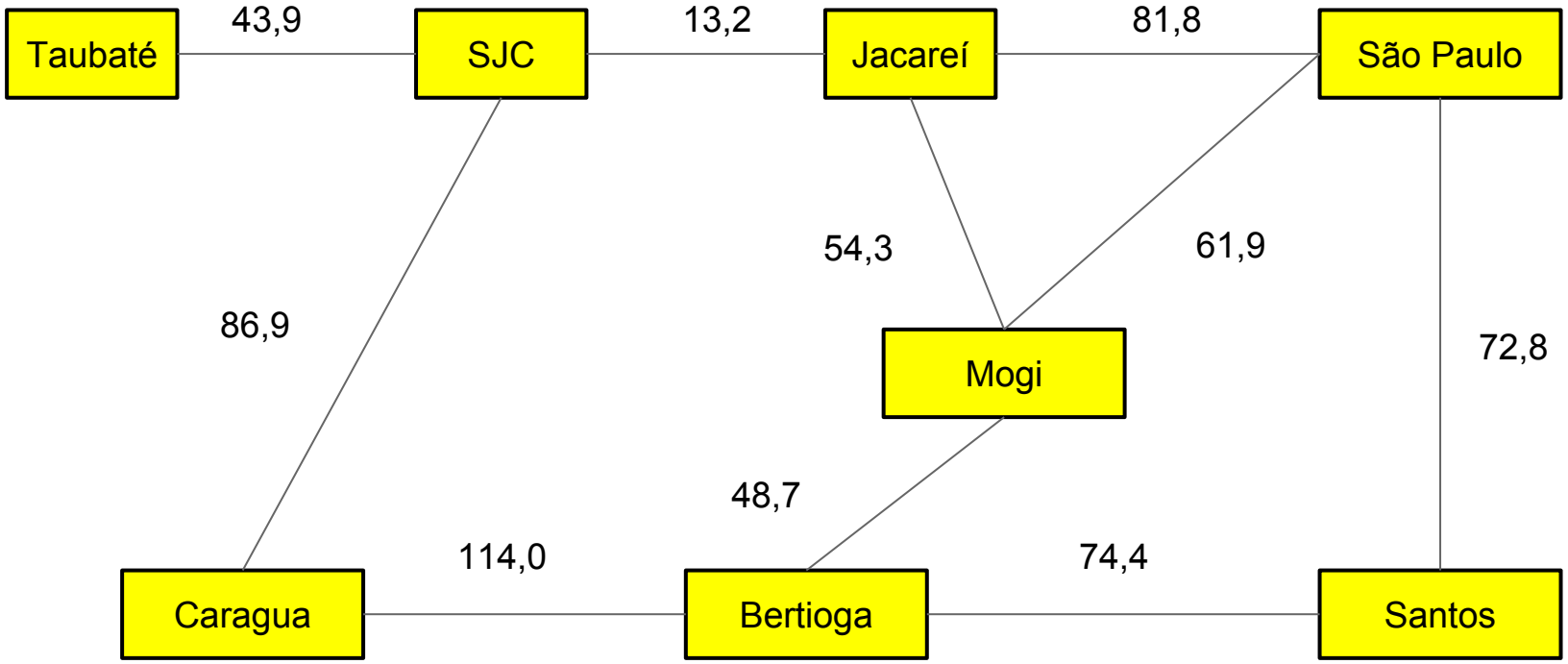
Análise Geral

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.	$O(1)$

Email: renzo@python.pro.br **Twitter**: @renzoprobr

pythonpro

Caminho entre dois pontos



Email: renzo@python.pro.br **Twitter**: @renzoprobr

pythonpro

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 *

Email: renzo@python.pro.br **Twitter**: @renzoprobr

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 *

Email: renzo@python.pro.br **Twitter**: @renzoprobr

Exercício

Implementar grafo não orientado

Utilizar Mapa de Adjacências

Será utilizado no próximo exercício

Slides de testes: https://github.com/renzon/estrutura-de-dados/tree/master/aula11 *

Email: renzo@python.pro.br **Twitter**: @renzoprobr

Obrigado

renzo@python.pro.br
@renzoprobr