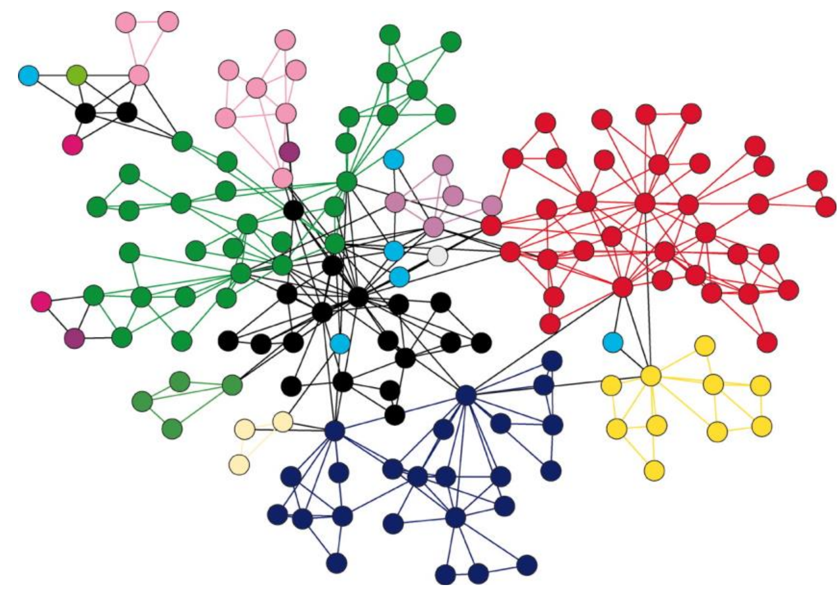
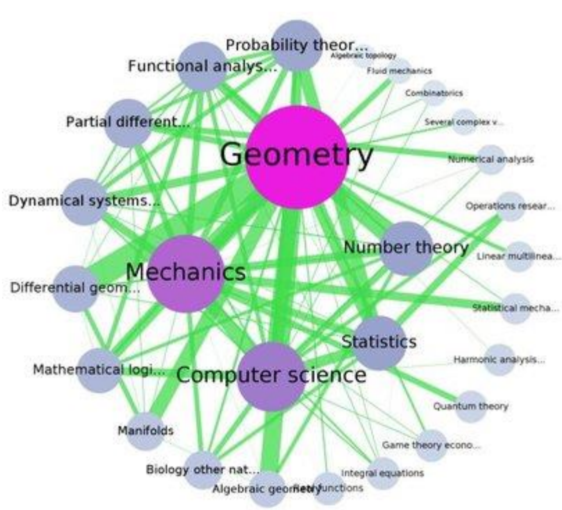
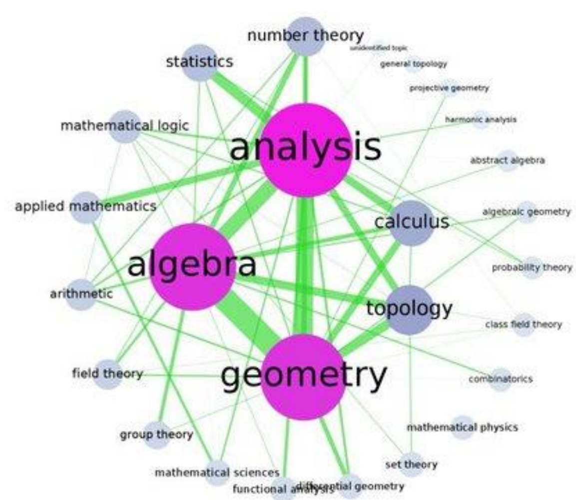
Capítulo

9 Grafos







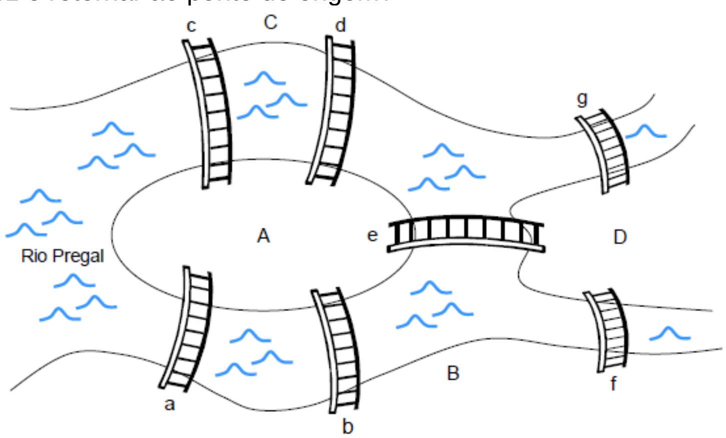
Objetivos

- Aprender o que é um grafo e como ele é usado.
- Implementar o tipo abstrato de dados grafo usando várias representações internas.
- Utilização dos grafos para resolver uma ampla variedade de problemas

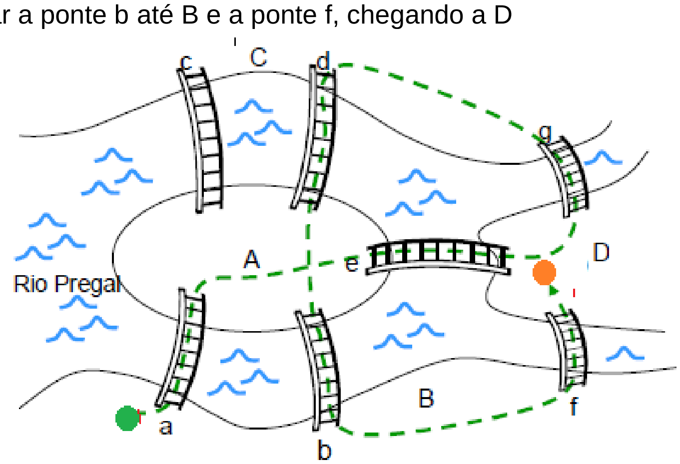
Histórico

A primeira evidência sobre grafos (*graphs*) remonta a 1736, quando Euler fez uso deles para solucionar o problema clássico das pontes de Koenigsberg. Na cidade de Koenigsberg (na Prússia Oriental), o rio Pregal flui em torno da ilha de Kneiphof, dividindo-se em seguida em duas partes. Assim sendo, existem quatro áreas de terra que ladeiam o rio: as áreas de terra (A-D) estão interligadas por sete pontes (a-g).

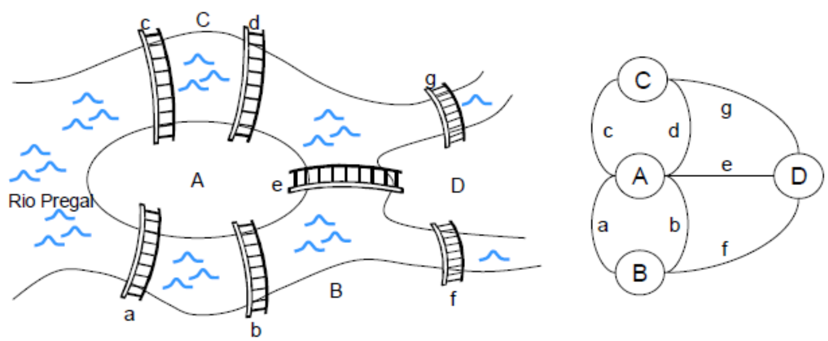
O problema das pontes de Koenigsberg consiste em se determinar se, ao partir de alguma área de terra, é possível atravessar todos os pontos exatamente uma vez, para, em seguida, retornar à área de terra inicial. É possível caminhar sobre cada ponte exatamente uma única vez e retornar ao ponto de origem?



Um caminho possível consistiria em iniciar na área de terra B, atravessar a ponte a para a ilha A; pegar a ponte e para chegar à área D, atravessar a ponte g, chegando a C; cruzar a ponte d até A; cruzar a ponte b até B e a ponte f, chegando a D



Euler provou que não é possível o povo de Koenigsberg atravessar cada ponte exatamente uma vez, retornando ao ponto inicial. Ele resolveu o problema, representando as áreas de terra como vértices e as pontes como arestas de um grafo (na realidade, um multigrafo)



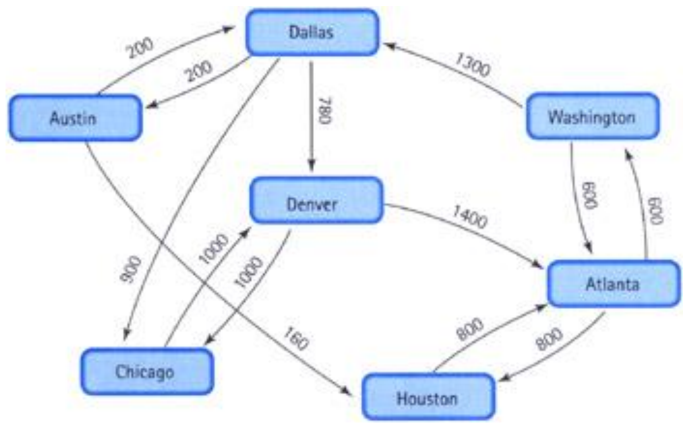
Introdução

As estruturas de dados Vetor, Lista, Fila e Pilha, são estruturas de dados unidimensionais ou lineares, e existem aplicações que necessitamos de outras estruturas mais complexas, como por exemplo, estruturas do tipo hierárquica, isto é, um determinado nó possui, um ou mais nós hierarquicamente inferiores. Podemos destacar vários exemplos de aplicações de estruturas hierárquicas bem como: as pastas, subpastas e documentos de um computador, índice remissivo de um determinado livro, a árvore filogenética da vida etc.

Os grafos são estruturas de dados mais geral as árvores ou podemos pensar que as árvores são estruturas especiais de grafos. Os grafos podem ser usados para representar muitas coisas interessantes sobre o nosso mundo, incluindo sistemas rodoviários, voos aéreos de cidade em cidade, como a Internet está conectada ou mesmo a sequência de aulas que você deve fazer para concluir um curso de ciência da computação. Veremos neste capítulo que, uma vez que tenhamos uma boa representação para um problema, podemos usar alguns algoritmos de gráfico padrão para resolver o que, de outra forma, poderia parecer um problema muito difícil.

Embora seja relativamente fácil para os humanos olhar para um mapa rodoviário e entender as relações entre os diferentes lugares, um computador não tem esse conhecimento. No entanto, também podemos pensar em um roteiro como um gráfico. Quando fazemos isso, nosso computador pode fazer coisas interessantes por nós. Se você já usou um dos sites de mapas da Internet, sabe que um computador pode encontrar o caminho mais curto, rápido ou fácil de um lugar para outro.

A figura a seguir ilustra um grafo com as distâncias entre algumas cidades americanas. Repare que necessitamos conhecer os nós dos grafos e os elos de ligação entre esses nós.



Primeiras noções

Numa escola algumas turmas resolveram realizar um torneio de vôlei. Participam do torneio as turmas 6*A*, 6*B*, 7*A*, 7*B*, 8*A* e 8*B*. Alguns jogos foram realizados até agora:

6*A* jogou com 7*A*, 7*B*, 8*B*
6*B* jogou com 7*A*, 8*A*, 8*B*
7*A* jogou com 6*A*, 6*B*
7*B* jogou com 6*A*, 8*A*, 8*B*
8*A* jogou com 6*B*, 7*B*, 8*B*
8*B* jogou com 6*A*, 6*B*, 7*B*, 8*A*

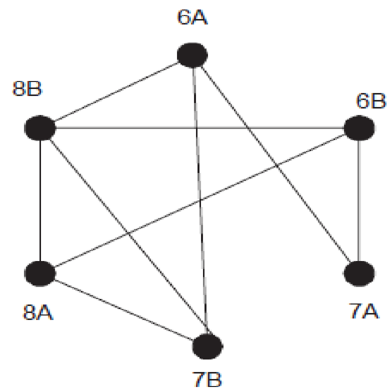
Mas será que isto está correto ? Pode ter havido um erro na listagem?

Uma maneira de representar a situação através de uma figura as turmas serão representadas por pontos e os jogos serão representados por linhas. Não ´e difícil agora constatar a consistência das informações. A estrutura que acabamos de conhecer é um **grafo**.

Apresentamos duas formas de representar esta estrutura

- Por uma lista, dizendo quem se relaciona com quem.
- Por um desenho, isto é, uma representação gráfica do problema.

A figura a seguir ilustra a representação gráfica do problema.



Para que um grafo fique bem definido temos que ter dois conjuntos:

- O conjunto *V* , dos **vértices** - no nosso exemplo é o conjunto das turmas.
- O conjunto *A*, das **arestas** - no nosso exemplo, os jogos realizados.

Em outras palavras, o que nos interessa num grafo é:

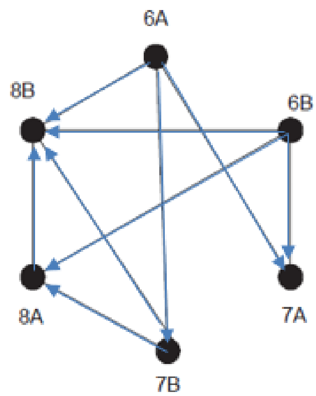
- Quem são os **vértices**.
- Que pares de vértices estão ligados e quais não estão (isto ´e, quem são as **arestas**).

Quando existe uma aresta ligando dois vértices dizemos que os vértices sãó **adjacentes** e que a aresta é **incidente** aos vértices. No nosso exemplo podemos representar o grafo de forma sucinta como:

V = *{* 6*A*; 6*B*; 7*A*; 7*B*; 8*A*; 8*B }*
A = *{* (6*A*; 7*A*); (6*A*; 7*B*); (6*A*; 8*B*); (6*B*; 7*A*); (6*B*; 8*A*); (6*B*; 8*B*); (7*B*; 8*A*);(7*B*; 8*B*); (8*A*; 8*B*)*}*

Observe que não precisamos colocar (8*A*; 7*B*) no conjunto de arestas pois já tínhamos colocado (7*B*; 8*A*).

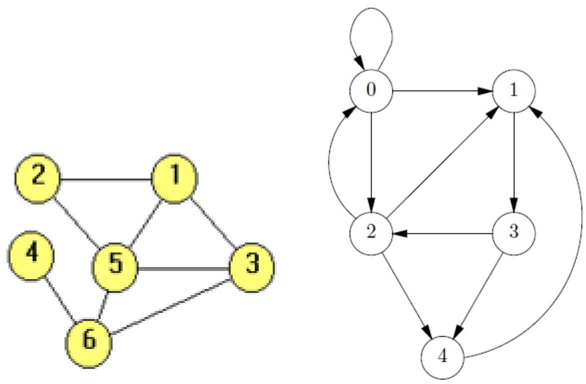
Mas e se o campeonato tivesse turno e returno?! Aí o nosso grafo teria que ser representado graficamente da seguinte forma



Observe que nesse caso, a direção da seta (arco ou arresta) é fundamental para entendermos qual vértice liga a outro vértice. Esse caso é conhecido como grafo dirigido.

Grafos

Grafo é uma estrutura matemática que consiste em um conjunto nós denominados de vértices e um conjunto de linhas que ligam estes nós. Estas linhas são denominadas de arestas (edges) no caso de linhas não orientadas ou, de setas, denominadas de arcos (arcs), no caso de linhas orientadas. A Figura 1.1 ilustra a representação gráfica de dois grafos sendo que na letra (a) pode-se observar que as linhas que ligam dois vértices quaisquer, não é orientada, neste caso são arestas. No caso da letra (b) as linhas são orientadas, isto é, são setas, denominadas de arcos.



(a) (b)

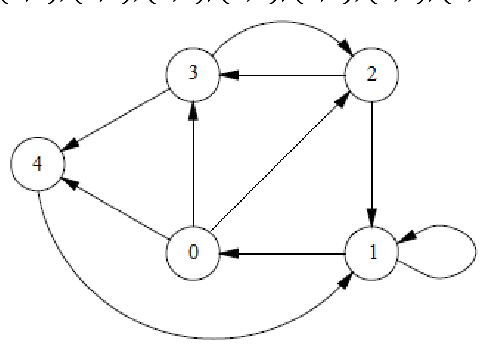
Figura 1.1 – Representação Gráfica de dois Grafos

O grafo da figura (a) é denominado de grafo não dirigido e o grafo da letra (b) é denominado de dirigido.

Grafo Dirigido

Um Grafo Dirigido (Directed Graph) consiste de um conjunto de nós (vértices) V e um conjunto de arcos que são as setas que ligam dois nós quaisquer do conjunto V. É importante frisarmos que o conjunto de arcos { (u,v) } indica que u é o vértice origem e v o vértice extremidade do arco.

$$V = \{\ 0,1,2,3,4\ \}$$
$$A = \{\ (0,4), (0,2), (0,3), (1,0), (1,1), (2,1), (2,3), (3,2), (3,4), (4,1)\ \}$$

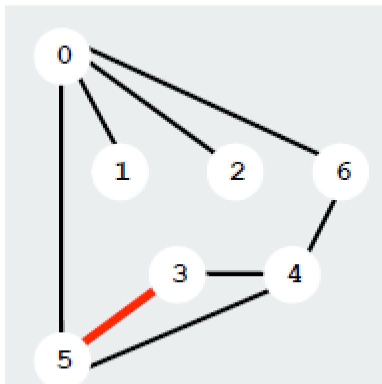


,

Grafo Não Dirigido

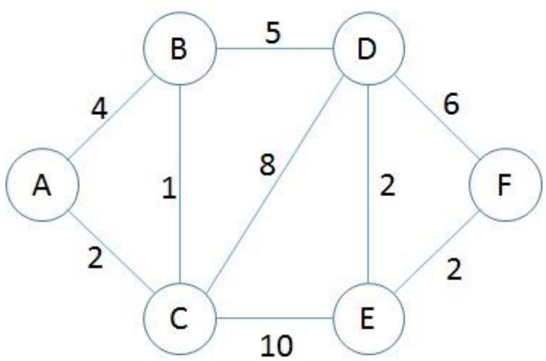
Um Grafo não Dirigido (UnDirected Graph) consiste de um conjunto de nós (vértices) V e um conjunto de arestas que são as linhas que ligam dois nós quaisquer do conjunto V. É importante frisarmos no grafo não dirigido, se existir uma aresta (u,v) obrigatoriamente terenos a aresta (v, u).

$$V = \{ 0,1,2,3,4,5,6 \}$$
$$A = \{(0,1), (0,2), (0,5), (0,6), (1,0), (2,0), (3,4), (3,5), (4,3), (4,5), (4,6), (5,0), (5,3), (5,4), (6,0), (6,4)\}$$



Grafo Valorado

Um **grafo valorado**, é um tipo de grafo cujas arestas possuem um peso (weight). Por exemplo considere o grafo a seguir.



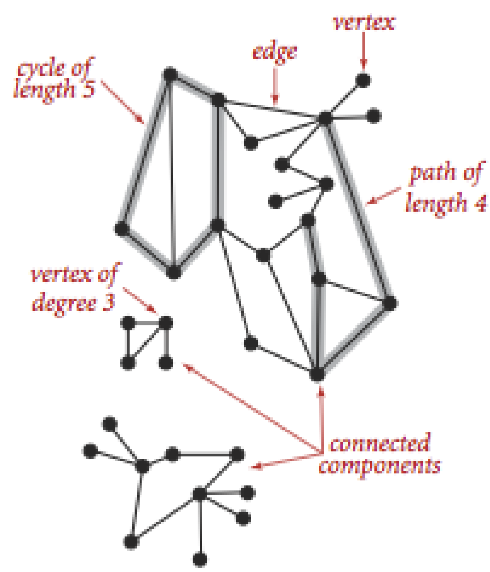
O peso (weight) de uma aresta é um valor alocado à aresta. Por exemplo no grafo anterior, temos

$$\text{weight}(A, B) = 4$$
$$\text{weight}(C, D) = 8$$

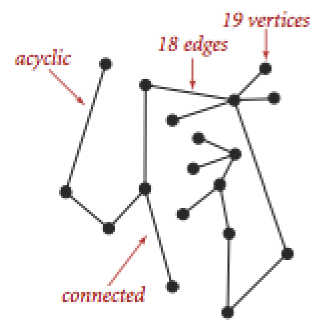
Genericamente

$$\text{weight}(vérice\ u, vérice\ v) = value$$

Anatomia de um Grafo



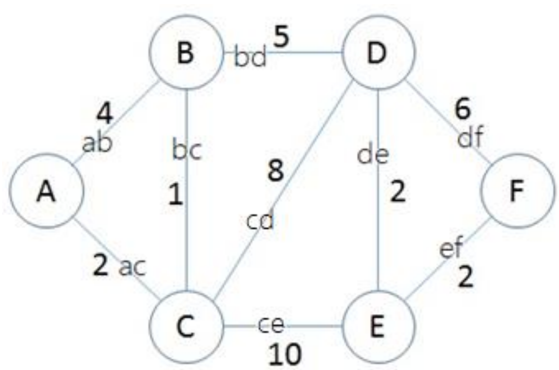
Anatomy of a graph



A tree

Conceitos e características de um Grafo

Considere o grafo não dirigido a seguir



O **rótulo** (label) de uma aresta é um identificador da aresta. Por exemplo no grafo anterior podemos determinar alguns rótulos de arestas:

$$label\ (A, B) = ab$$
$$label\ (A, C) = ac$$

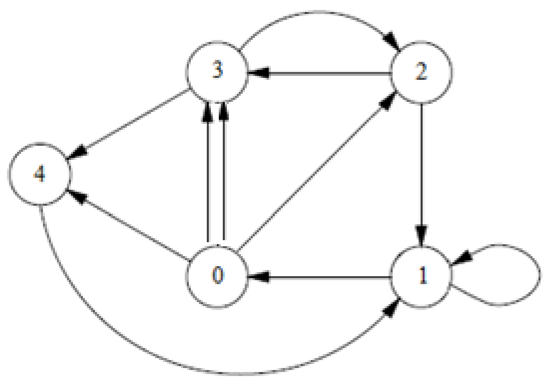
O vértice **oposto** (opposite) à um vértice u, considerando que o rótulo (label) de uma determinada aresta seja l, é um vértice v, tal $label\ (u, v) = l$. Na figura anterior podemos obter alguns rótulos de determinados arcos

$$opposite(A, ab) = B$$
$$opposite(C, cd) = D$$

Genericamente

$$opposite(source\ vertex, edge\ label) = \ sink\ vertex$$
$$oposto(\ vertex\ origem, label\ da\ aresta) = \ vertex\ chegada$$

Considere agora o grafo dirigido a seguir



Dizemos que um determinado grafo possui um ou mais laços, quando existir pelo menos um vértice que tem ligação com ele mesmo. Na figura anterior temos um laço que ocorre no vértice 1.

Dizemos que um determinado grafo possui um duas ou mais arestas paralelas, quando existir pelo menos um par de vértices (u,v) que possui mais de uma aresta ligando esses vértices. Na figura anterior temos as arestas paralelas entre os vértices 0 e 3.

Dizemos que um determinado grafo possui um duas ou mais arestas antiparalelas, quando existir pelo menos um par de vértices (u,v) que possui pelo menos uma pelo menos uma aresta ligando u a v, e pelo menos uma aresta ligando v e u. Nesse caso o grafo tem que ser dirigido. Na figura anterior temos as arestas antiparalelas entre os vértices 2 e 3.

Dois vértices u e v, são ditos adjacentes (, quando existir, pelo menos, uma aresta que liga esses vértices. Na figura anterior podemos verificar se dois vértices são adjacentes/

$adjacent(0,4) \quad True$
$adjacent(4,0) \quad False$ (repare que p grafo é dirigido)

O conjunto de vértices adjacentes à um determinado vértice é denominado de sucessores (successors). Na figura anterior podemos obter sucessores de alguns vértices:

$$successors(0) = \{2,3,4\}$$
$$successors(1) = \{0,1\}$$

Genericamente,

$$successores(source\ vertex) = \ \{vertices\ arrival\ \}$$

conjunto de vértices que chegam, de um determinado vértice é denominado de predecessores (predecessor) ou incidentes (incidentes). Na figura anterior podemos obter sucessores de alguns vértices:

$$incidents(0) = \{1\}$$
$$incidents(1) = \{1,4\}$$

Genericamente,

$$incidentes(vétice\ chegada) = \{\ vértices\ origem\}$$
incidents$incidents(sink\ vertex) = \{\ source\ vertices\}$

Um caminho (path) em um grafo é uma determina sequência de vértices adjacentes que começam em um vértice origem e terminam em um vértice destino. Na figura anterior podemos obter caminhos entre alguns vértices:

$path(0,1) = \{0,4,1\}$
$path(0,3) = \{0,4,1,0,3\}$
$path(1,3) = \{1,1,0,3\}$ Observe que nesse caminho temos um laço

Genericamente,

$$path(source\ vertex) = \{\ vertex, \ldots\ldots vertex\}$$
$$caminho(vétice\ origem) = \{\ vertice, \ldots, vertice\}$$

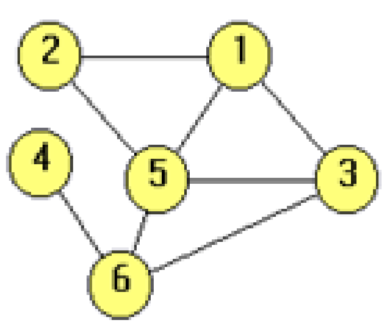
Um ciclo (cycle) em um grafo é uma determino caminho que começa e termina em um mesmo vértice. Na figura anterior podemos obter ciclos entre alguns vértices:

$cycle = \{0,4,1,0\}$
$cycle = \{1,0,3,2,1\}$
$cycle = \{1\}$ Observe que nesse ciclo temos um laço

Genericamente,

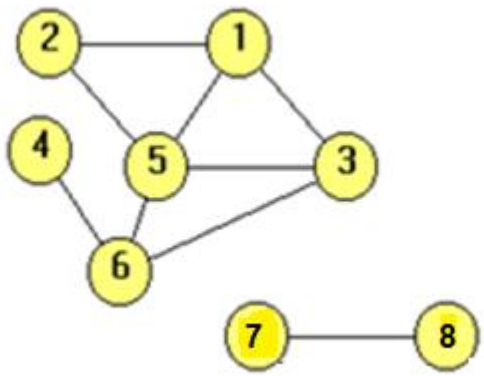
$$cycle = \{\ source\ vertex, \ldots\ldots source\ vertex\}$$
$$ciclo = \{\ vértice\ origem, \ldots\ldots vértice\ origem\}$$

Considere agora o grafo não dirigido a seguir

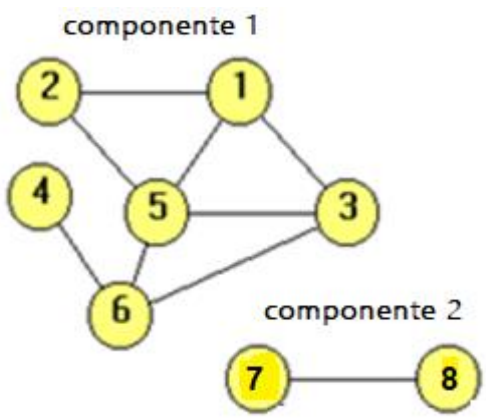


Um grafo é dito conexo (connected) quando existir pelo menos um caminho que liga um determinado vértice à todos os outro vértices. Na figura anterior podemos observar que sempre existirá um caminho que passa por todos os vértices:

Observe que no grafo da figura a seguir, não existe caminho que percorre todos os vértices, nesse caso, denominamos que o grafo e desconexo (disconnected).



Observe que no grafo desconexo da figura anterior, temos duas partes, que são grafos conexos, denominado de componentes (components).

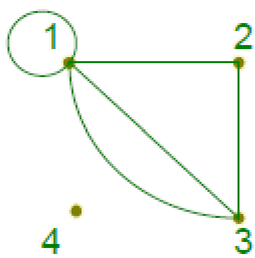


Representação interna dos grafos

Os grafos podem ser estruturados de várias formas. As mais conhecidas são na forma de uma matriz, denominada de matriz de adjacência e, na forma de listas, denominado de lista de adjacência.

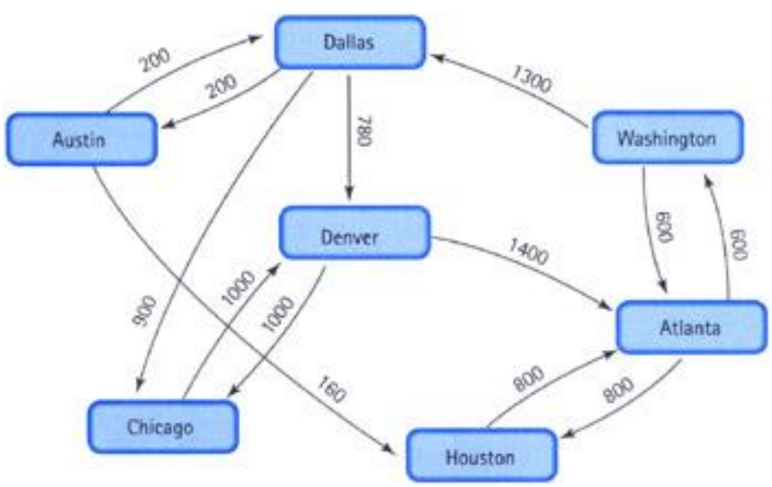
Matriz de adjacências:

Seja Γ um grafo com o conjunto de vértices $\{v_1, v_2, ..., v_n\}$. A matriz de adjacências de Γ é uma matriz n×n $A = A(\Gamma)$ tal que a_{ij} é o número de arestas distintas que ligam v_i a v_j.



$$A = \begin{bmatrix} 1 & 1 & 2 & 0 \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Considerando o exemplo das ligações entre algumas cidades americanas



graph

.num Vertices 7

.vertices

[0]	"Atlanta "
[1]	"Austin "
[2]	"Chicago "
[3]	"Dallas "
[4]	"Denver "
[5]	"Houston "
[6]	"Washington"
[7]	
[8]	
[9]	

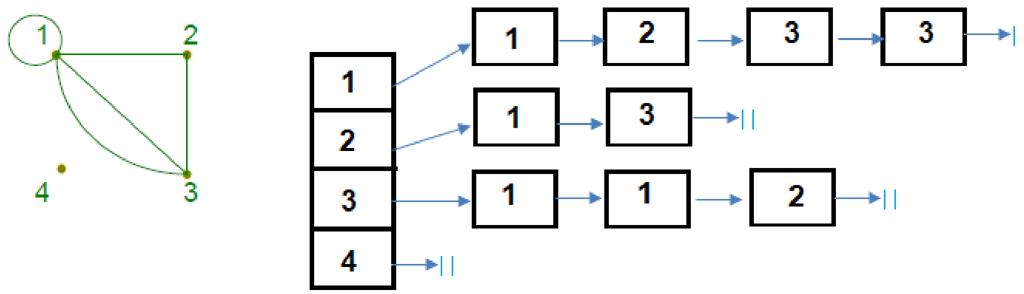
.edges

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
[0]	0	0	0	0	0	800	600	•	•	•
[1]	0	0	0	200	0	160	0	•	•	•
[2]	0	0	0	0	1000	0	0	•	•	•
[3]	0	200	900	0	780	0	0	•	•	•
[4]	1400	0	1000	0	0	0	0	•	•	•
[5]	800	0	0	0	0	0	0	•	•	•
[6]	600	0	0	1300	0	0	0	•	•	•
[7]	•	•	•	•	•	•	•	•	•	•
[8]	•	•	•	•	•	•	•	•	•	•
[9]	•	•	•	•	•	•	•	•	•	•

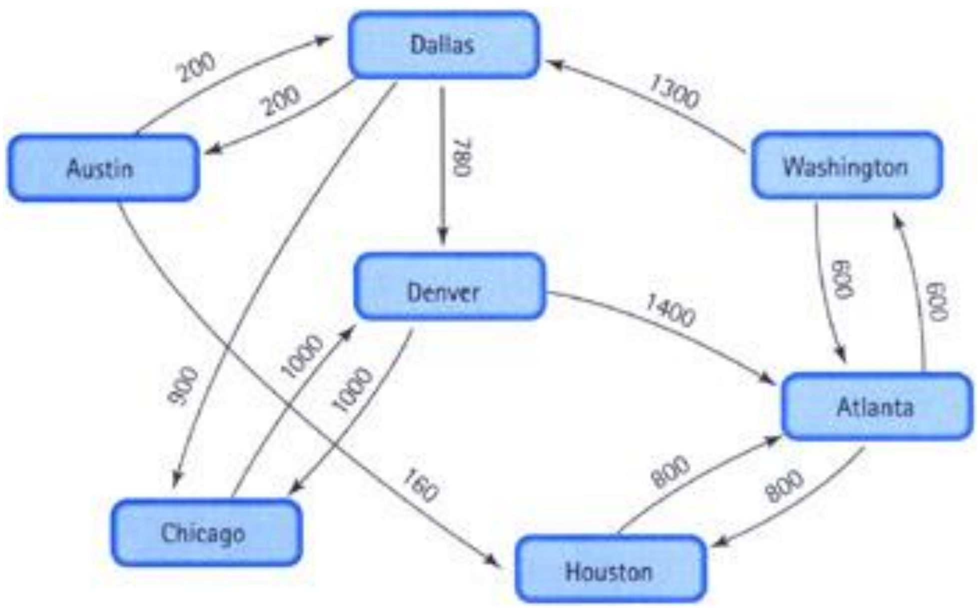
(Array positions marked '•' are undefined)

Lista de adjacências:

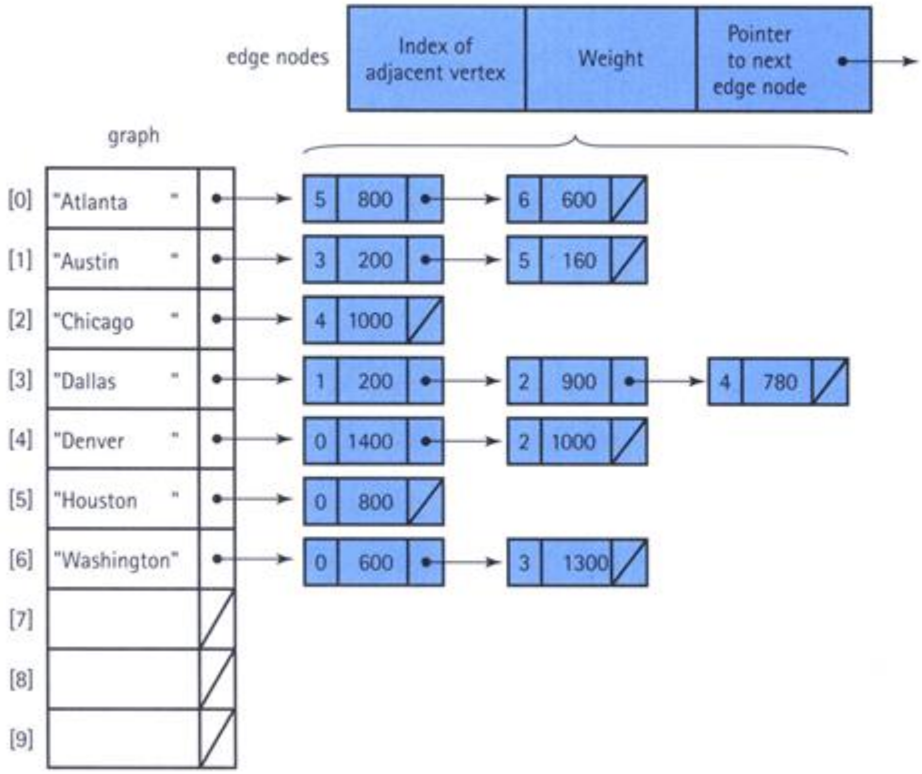
Seja Γ um grafo com o conjunto de vértices {v_1, v_2, ..., v_n}. A lista de adjacências de Γ é um conjunto de n células que contêm os vértices e um ponteiro para uma lista de vértices adjacentes.



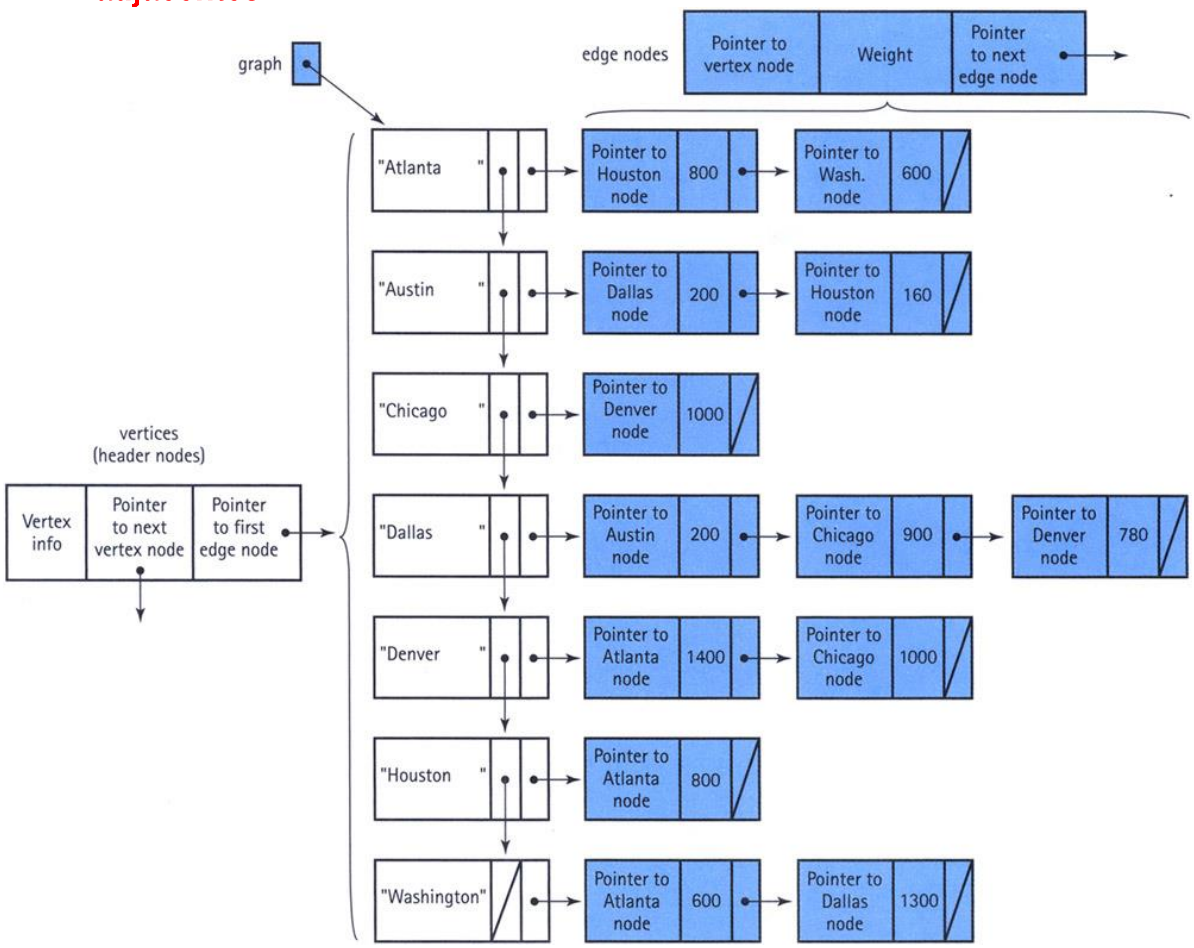
Considerando o exemplo das ligações entre algumas cidades americanas



a) Vetor com Vértices e ponteiros para a lista de adjacência



b) **Lista encadeada de Vértices com ponteiros para a lista de vértices adjacentes**



Implementação de um Grafo em Python

A forma mais simples de se implementar um grafo em Python é utilizarmos a estrutura dicionário.