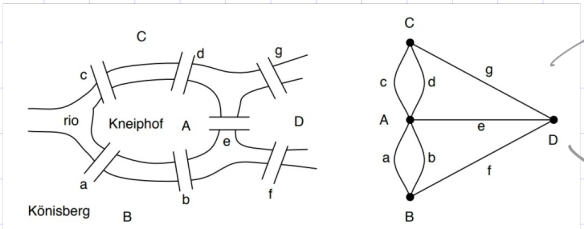
Grafos [Introdução]

Referência

Definição: estrutura matemática que modela relações entre elementos.
↳ ou conceito, ideia, abstração...

Histórico: Sete pontes de Königsberg (EULER, 1736)

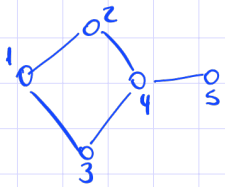


→ Grafo

→ Ciclo euleriano

Definição formal: Grafo $G = (V, E)$ é definido por um conjunto V de vértices e um conjunto E de arestas, onde cada aresta $e \in E$ é definida por um par não ordenado de vértices $\{u, v\}$, onde $u, v \in V$.

Exemplo:

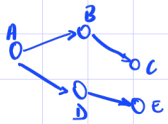


Vértices $V = \{1, 2, 3, 4, 5\}$
Arestas $E \{\{1,2\}, \{1,3\}, \{2,4\}, \{3,4\}, \{4,5\}\}$

Um grafo direcionado (também chamado dígrafo) é aquele no qual as arestas possuem uma direção. Neste caso, as chamamos de arcos.

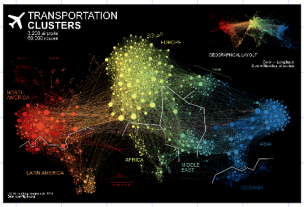
↳ $G = (V, A)$, onde $a = (u, v)$, para $a \in A$ e $u, v \in V$.

↳ Ex: $V = \{A, B, C, D, E\}$, $A = \{(A,B), (A,D), (B,C), (D,E)\}$

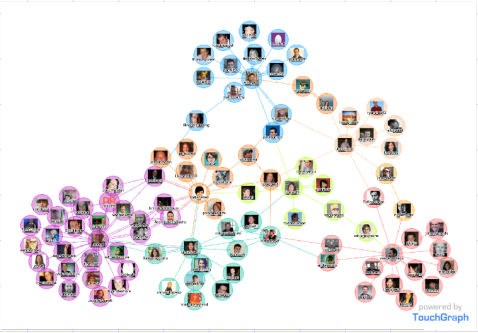


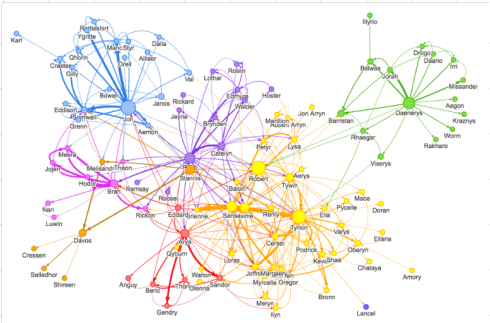
Aplicações

→ Redes de Transporte

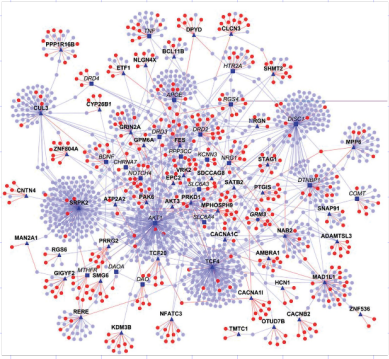


→ Redes Sociais

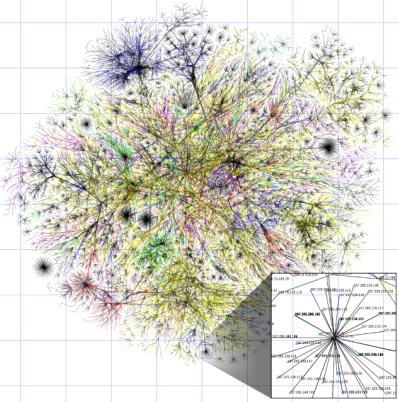




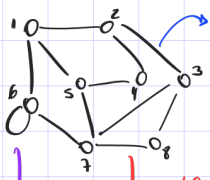
→ Interação de Proteínas



Internet



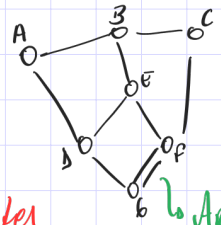
Conceitos:



Aresta é incidente ao vértice 2 (e também ao 3)

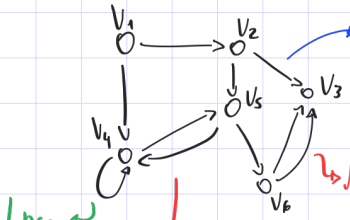
Vértices 7 e 8 são adjacentes (conectados por uma aresta)

Esta aresta é um laço (loop) (incide a somente um vértice)



Arestas paralelas (conectam mesmo par de vértices)

Para Dígrafos:



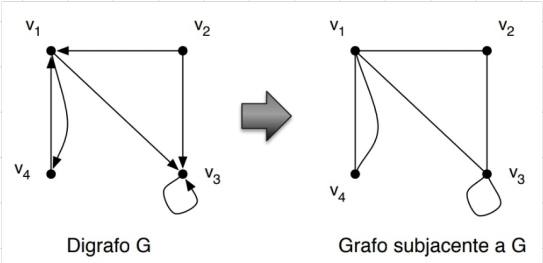
v_3 é adjacente a v_2, mas v_2 NÃO é adjacente a v_3

Arcos paralelos

Laço

NÃO são arcos paralelos

Grafo subjacente: Seja G um grafo dirigido, seu grafo subjacente é aquele onde cada arco (dirigido) é substituído por uma aresta (não dirigida).

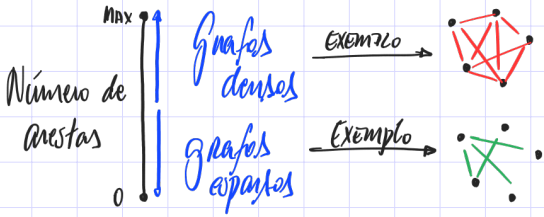


Grafo simples: não possui laços, nem arestas paralelas.

Multigrafo: contém ao menos um laço ou aresta paralela.

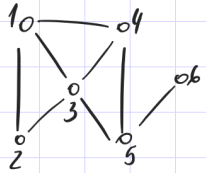
Parâmetros:

- Número de vértices ($|V|, n$) — Ordem do Grafo
- Número de arestas ($|E|, m$) — Tamanho do Grafo
- **Densidade**: Proporção entre número de arestas do grafo e o máximo de arestas que ele pode ter. (grafo simples)
  - $|E|/|V|^2 = m/n^2$ (dígrafo)
  - $2|E|/(|V|^2-|V|) = 2m/(n^2-n)$ (grafo)



- **Grau do vértice**: número de arestas incidentes ao vértice.
  - Para Dígrafos: Grau de entrada → Arcos que chegam no vértice. Grau de saída → Arcos que partem do vértice.

Exemplo:



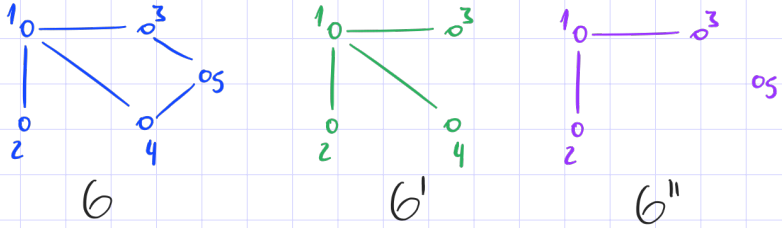
$n = |V| = 6$, $m = |E| = 8$

$Grau(4) = 3$, $Grau(6) = 1$

$Densidade = 2m/(n^2-n) = 28/(6^2-6) = 16/30 \approx 0{,}53$

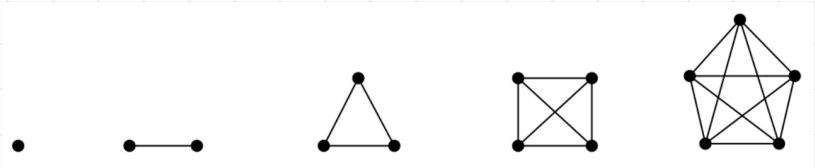
Subgrafo: Parte/recorte de um grafo.

↳ Se $G' = (V', E')$ é um subgrafo de $G = (V, E)$, então $V' \subseteq V$ e $E' \subseteq E$.

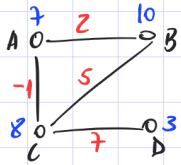


Grafo completo: Grafo simples em que todo par de vértices distintos é adjacente.

Exemplos



Grafo ponderado: Grafo que possui pesos (i.e. valores numéricos) associados a vértices e/ou arestas.



$w_A = 7$, $w_B = 10$

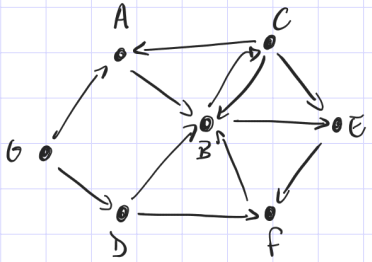
$w_{AC} = -1$, $w_{CD} = 7$

Exemplos: Distâncias (tráfego)
Custos (instalações)
Interações (rede social)

Caminho: Sequência de vértices $v_1, v_2, \dots, v_k$, onde cada par consecutivo v_{i-1}, v_i está ligado por uma aresta $e = \{v_{i-1}, v_i\} \in E$.

↳ Caminho simples: não repete vértice.

↳ Mesmo conceito para digrafos → Caminho direcionado



A B E f é um caminho de A a f

A D f é um caminho alternativo

Não existe caminho de B a G

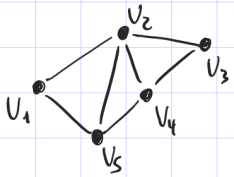
A D f E não é um caminho, pois a aresta (f,E) não existe no grafo

Comprimento do caminho: Número de arestas.

↳ Caso as arestas sejam ponderadas, o comprimento do caminho pode ser o somatório dos pesos das arestas.

↳ Chamado Custo do caminho.

Ciclo: Caminho fechado com mais de um vértice.
Ou seja, $v_1, v_2, \dots, v_k$, em que $v_1 = v_k$ e $k > 2$.



Ciclos: $\langle v_1, v_5, v_4, v_2, v_1 \rangle$
$\langle v_5, v_2, v_3, v_4, v_5 \rangle$
$\langle v_2, v_3, v_4, v_3 \rangle$

Representação (implementação) de Grafos

↳ Melhor escolha depende diretamente da aplicação.
↳ Cada representação fornece mais eficiência em algumas operações, e menos em outras.

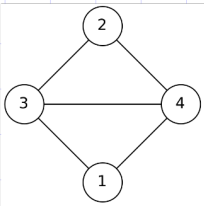
Abordagens comuns: Matriz de Adjacências
Lista de Adjacências

Matriz de Adjacências

Matriz $n \times n$ em que

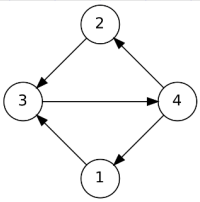
$$A_{uv} = \begin{cases} 1, & \text{se } \{u,v\} \in E, \\ 0, & \text{caso contrário.} \end{cases}$$

Em vez de {1,0}, a matriz pode armazenar o peso das arestas, ou qualquer outra informação (e.g. um objeto).



	1	2	3	4
1	0	0	1	1
2	0	0	1	1
3	1	1	0	1
4	1	1	1	0

Não dirigido



	1	2	3	4
1	0	0	1	0
2	0	0	1	0
3	0	0	0	1
4	1	1	0	0

Dígrafo

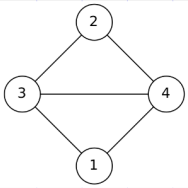
Análise:

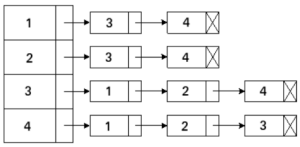
- Complexidade de espaço: $O(n^2)$
- Acessar uma aresta $\{u,v\}$: $O(1)$
- Percorrer vizinhos de um vértice v: $O(n)$
- Recuperar todas as arestas: $O(n^2)$

Lista de Adjacências:

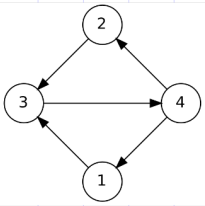
Vetor indexado pelos nodos com listas de vizinhos.

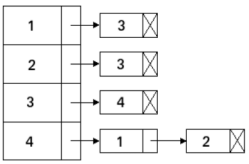
Podemos usar listas encadeadas ou arrays para esta implementação





Não dirigido





Dígrafo

Análise:

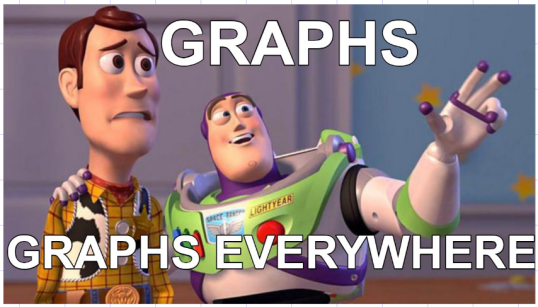
- Complexidade de espaço: $O(m+n)$
- Acessar uma aresta $\{u,v\}$: $O(grau(u))$
- Percorrer vizinhos de um vértice v: $O(grau(v))$
- Recuperar todas as arestas: $O(m+n)$

Arrays? → Mais eficiência em várias operações (acesso aleatório; busca binária). → Grafos estáticos

ou

Listas encadeadas? → Mais eficiência para inserir/remover vértices e arestas. → Grafos dinâmicos

Outras estruturas básicas podem ajudar

↳ Ex: Usar um mapa para armazenar a lista de adjacências de um vértice

1	→ {<2,->,<4,->{
2	→ {<1,->{
3	→ {<4,->{
4	→ {<2,->,<3,->{