Grafos I

Joaquim Madeira

01/06/2021

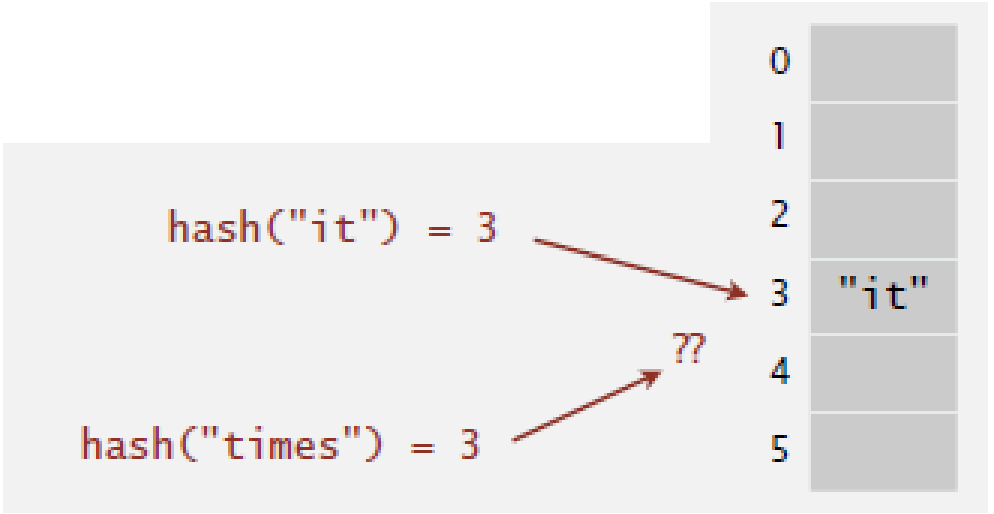
Sumário

- Recap
- Grafos: Terminologia; Exemplos de aplicação; Propriedades
- Os Tipos de Dados Grafo e Grafo Orientado
- Possíveis estruturas de dados
- Operações básicas
- Desempenho computacional
- Sugestão de leitura

Recapitulação

Hash Tables – Tabelas de Dispersão

- Armazenar itens numa tabela/array indexada pela chave
  - Índice é função da chave
- Função de Hashing: para calcular o índice a partir da chave
  - Rapidez !!
- Colisão: 2 chaves diferentes originam o mesmo resultado da função de hashing

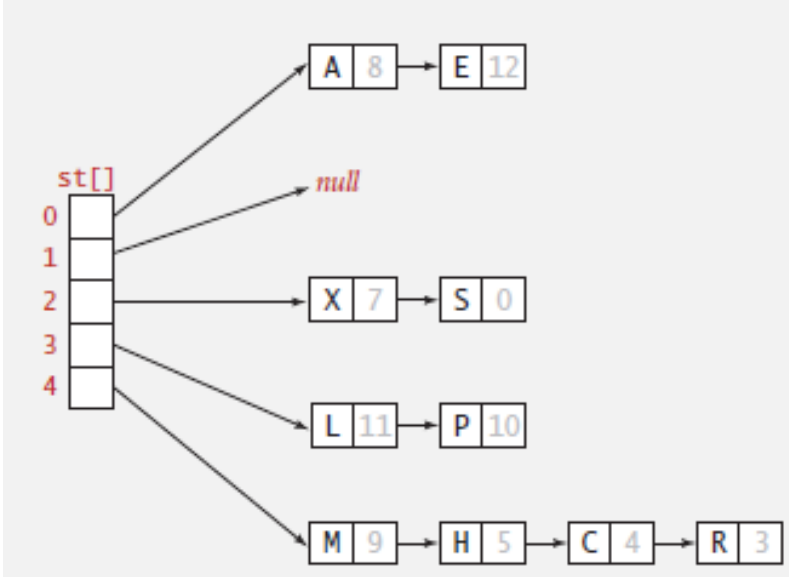


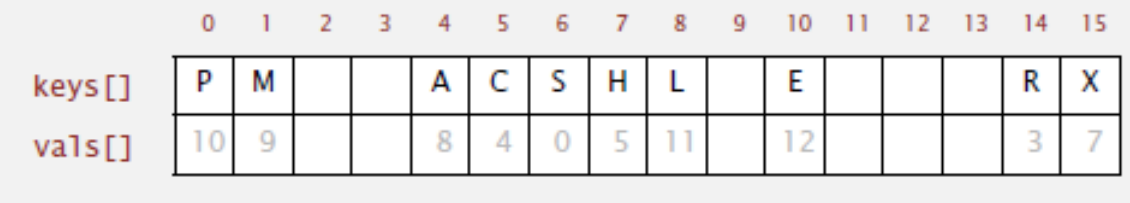
[Sedgewick & Wayne]

Separate Chaining vs Open Addressing

- Separate Chaining
- Desempenho não se degrada abruptamente
- Pouco sensível a funções de hashing menos boas

- Open Addressing
- Menos espaço de memória desperdiçado





[Sedgewick & Wayne]

Eficiência

implementation	guarantee			average case			ordered ops?	key interface
	search	insert	delete	search hit	insert	delete		
separate chaining	N	N	N	3-5 *	3-5 *	3-5 *		`equals()` `hashCode()`
linear probing	N	N	N	3-5 *	3-5 *	3-5 *		`equals()` `hashCode()`

* under uniform hashing assumption

[Sedgewick & Wayne]

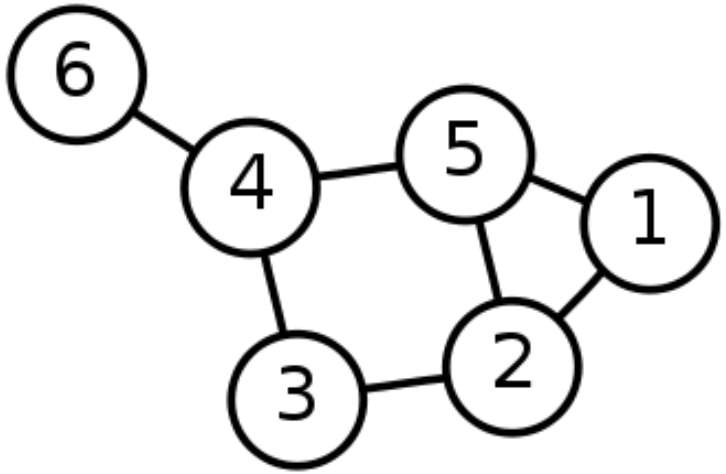
Hash Tables vs Balanced Search Trees

- Tabelas de Dispersão
- Código mais simples
- Melhor alternativa se não pretendermos ordem
- Mais rápidas, para chaves simples

- Árvores Binárias Equilibradas
- Pior caso : O(log N) vs O(N)
- Suportam ordem
- compareTo() vs equals() + hashCode()

Grafos
– Motivação + Exemplos

Grafo



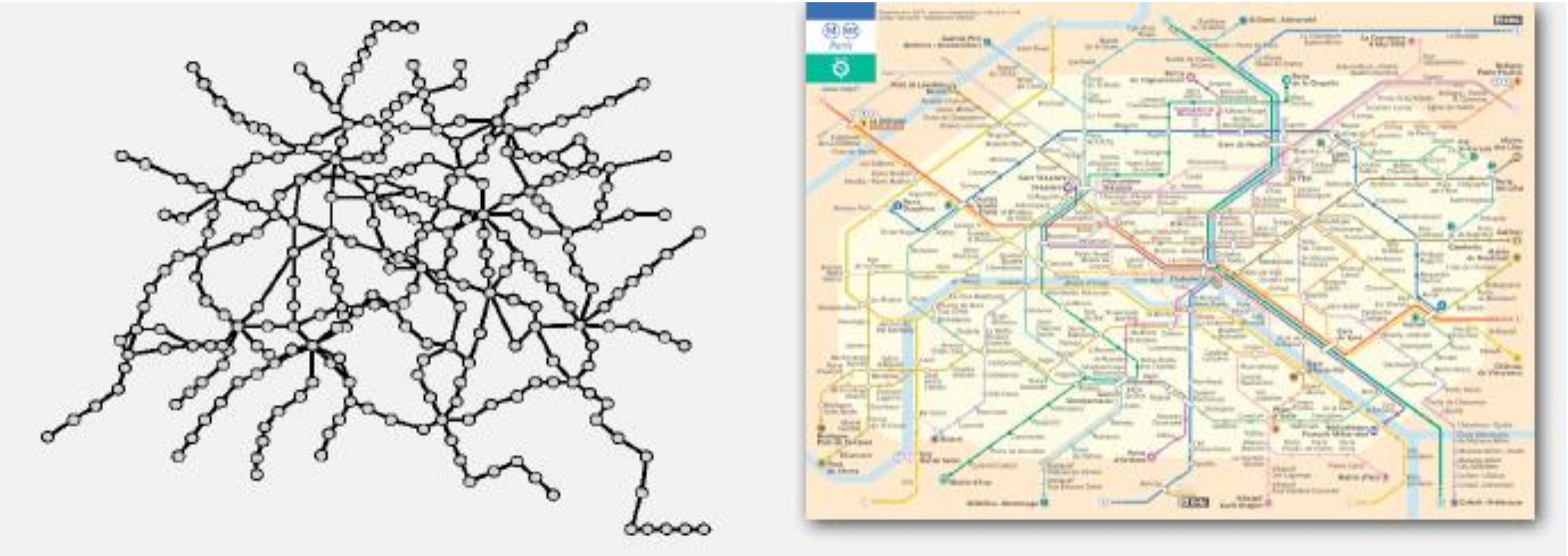
[Wikipedia]

- G(V, E)
- Quando muito uma aresta ligando qualquer par de vértices distintos
- $e_i = (v_j, v_k)$
  - v_j e v_k são vértices adjacentes
  - e_i é incidente em v_j e em v_k

Porquê estudar ?

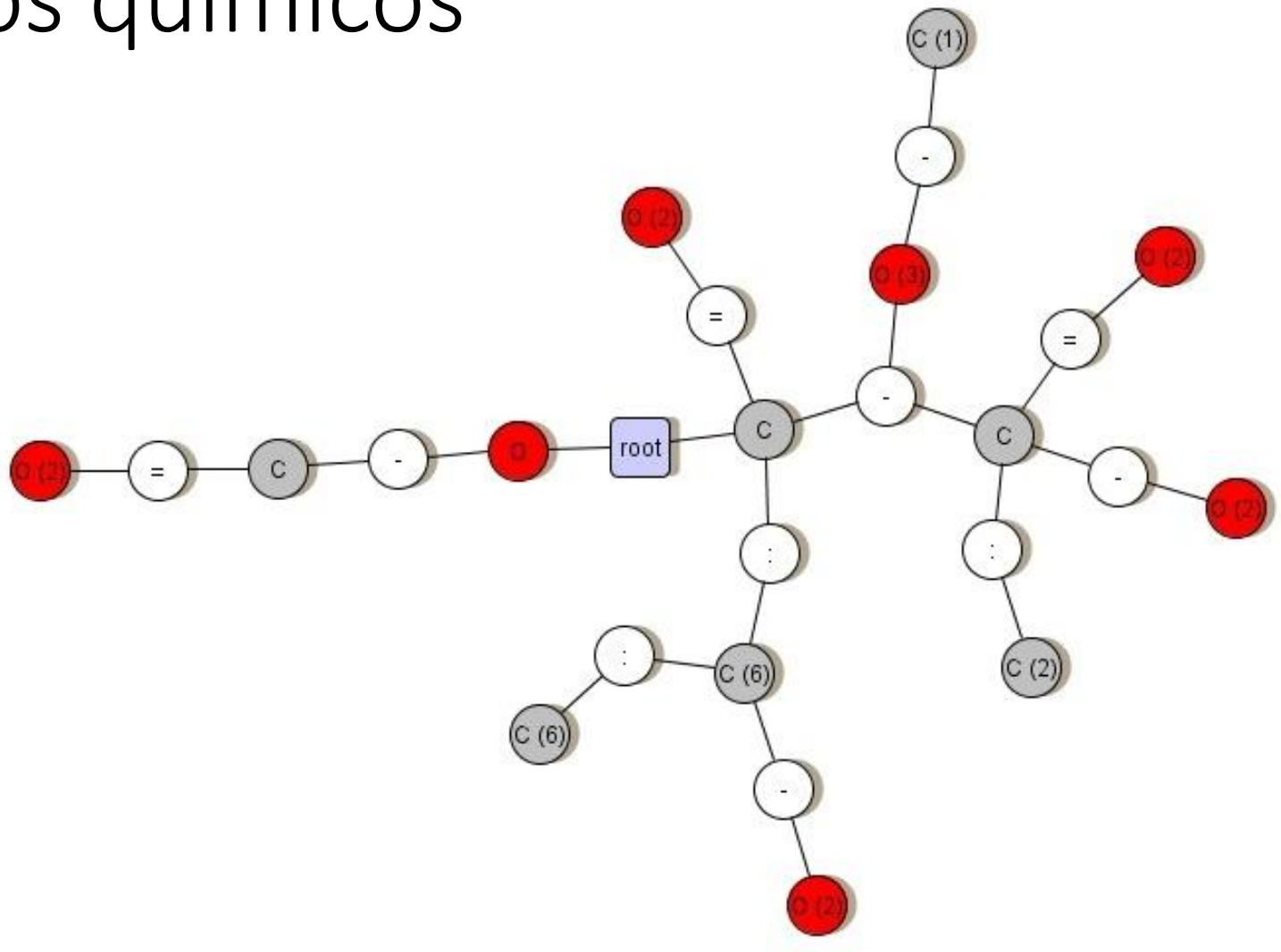
- Abstração útil
- Subárea das Ciências da Computação e da Matemática Discreta
  - Problemas / Algoritmos / Aplicações
- Centenas de algoritmos
- Milhares de aplicações práticas

Redes de transportes



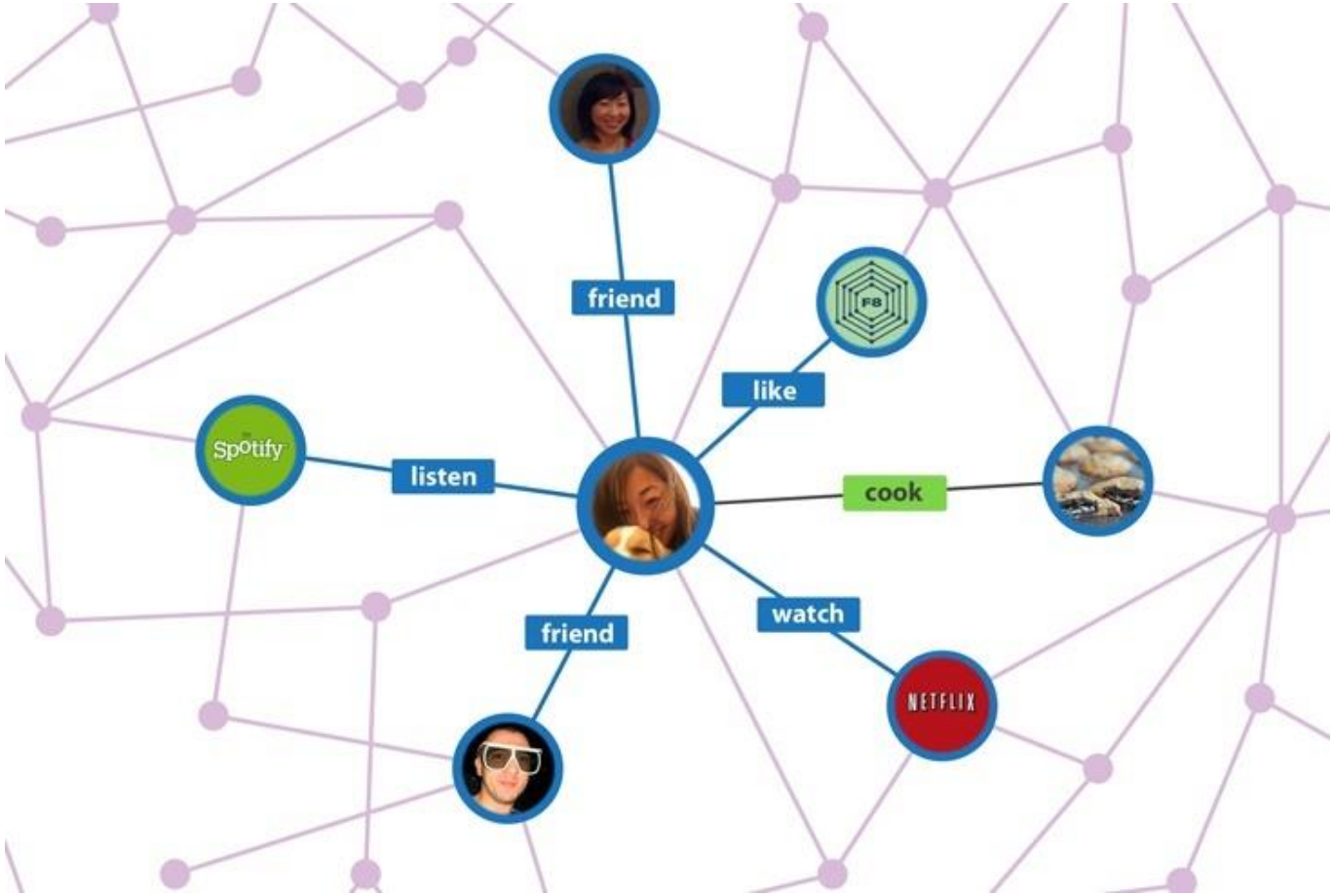
[Sedgewick & Wayne]

Modelos químicos



[Quora]

Redes sociais

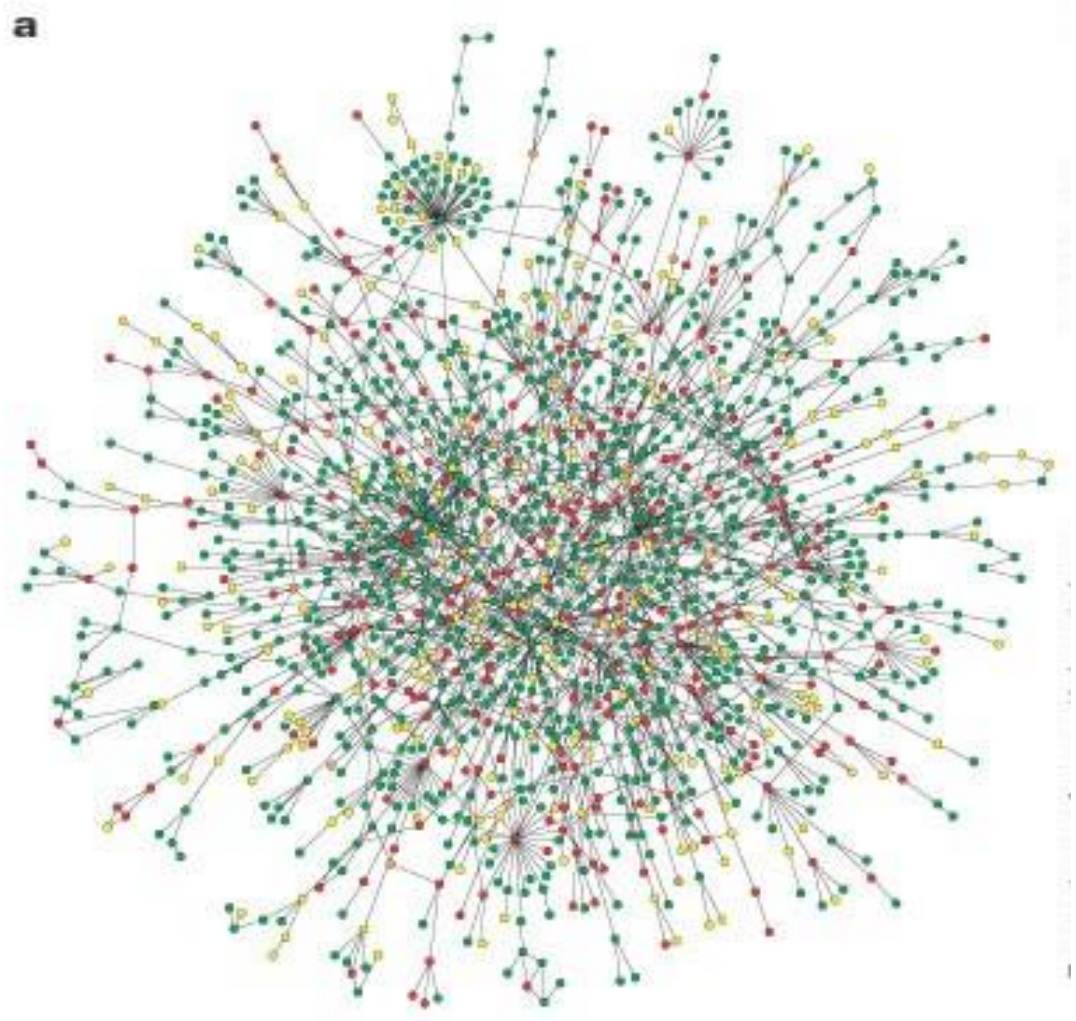


[Quora]

Sistemas de recomendação

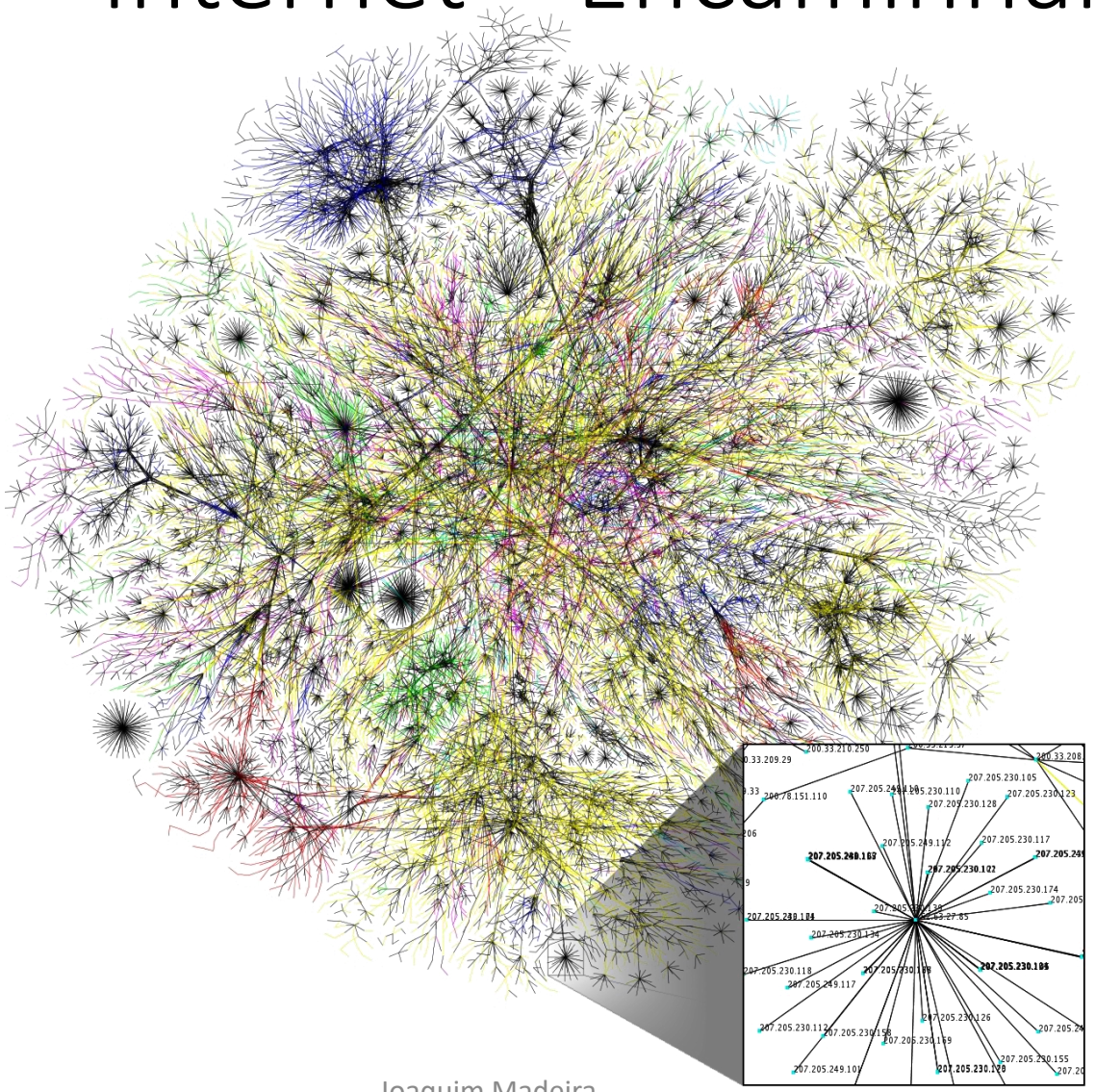
[Quora]

Letalidade e centralidade – Proteínas



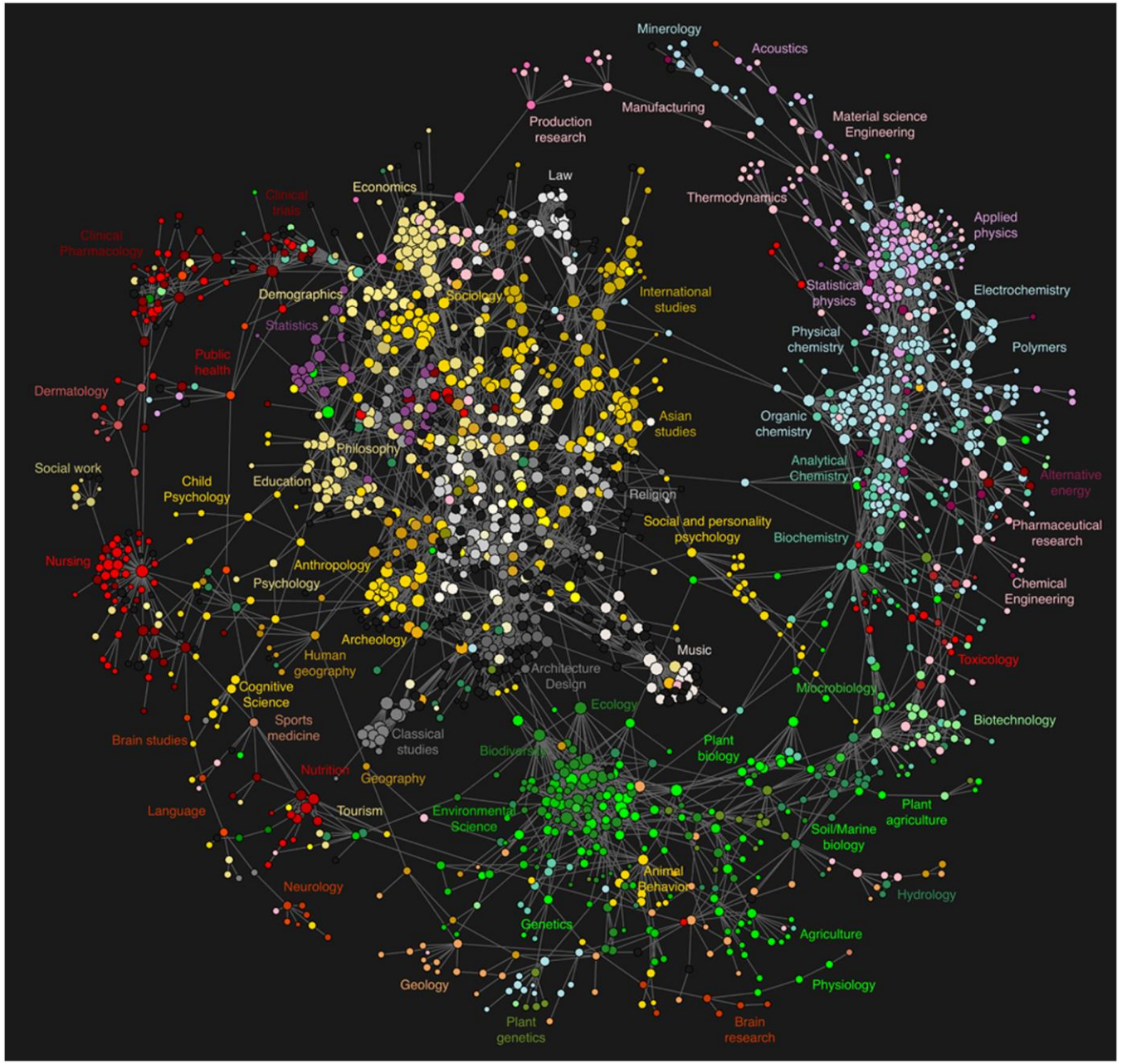
[Jeong et al 2001]

Visualização – Internet – Encaminhamento



[Wikipedia]

Atividade científica



[Bollen et al 2009]

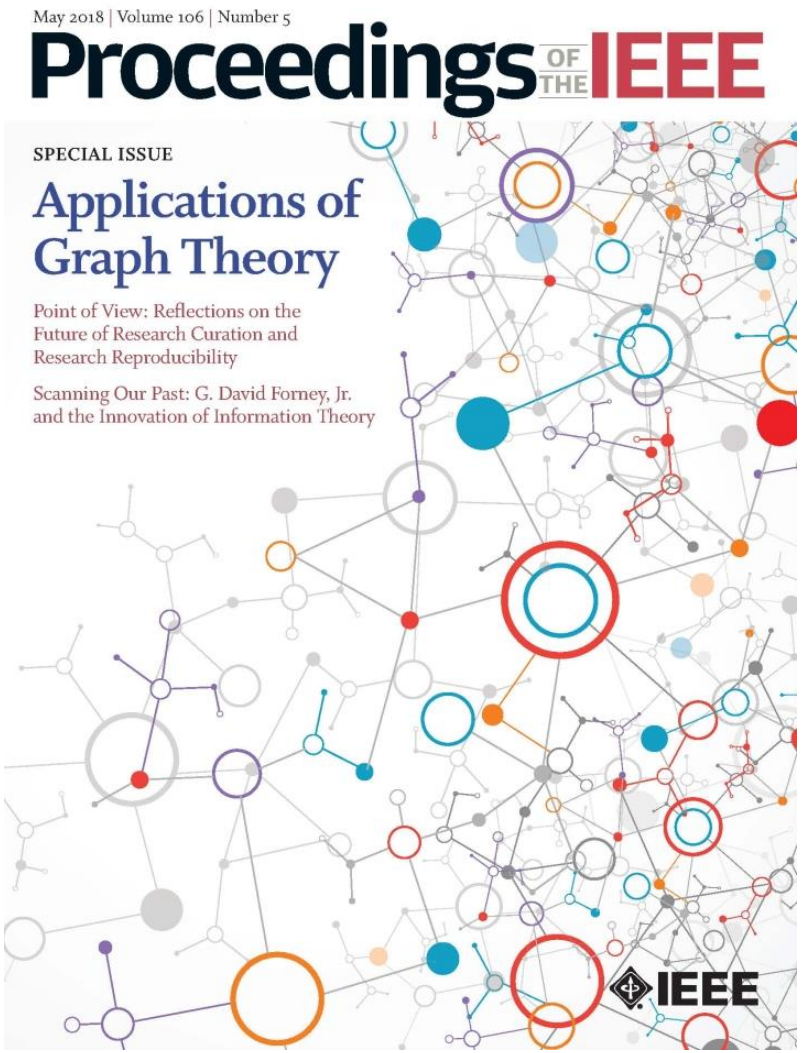
Aplicações

Graph	Vertex	Edge
communication	telephone, computer	cable
circuit	gate, register, processor	wire
mechanical	joint	rod, beam, spring
financial	stock, currency	transaction
transportation	street intersection, airport	highway, airway route
Internet	class C network	connection
game	board position	legal move
relationship	person	friendship
neural network	neuron	synapse
protein network	protein	protein-protein interaction
chemical compound	molecule	bond

[Sedgewick & Wayne]

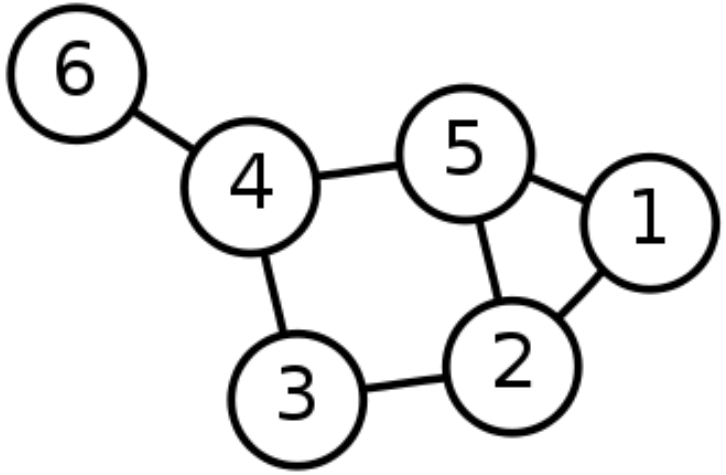
Mais aplicações

- Processamento de sinal
- Processamento de imagem
- Circuitos elétricos
- Neurologia
- …



Grafos
– Terminologia + Propriedades

Grafo



[Wikipedia]

- G(V, E)
- Quando muito uma aresta ligando qualquer par de vértices distintos
- $e_i = (v_j, v_k)$
  - v_j e v_k são vértices adjacentes
  - e_i é incidente em v_j e em v_k

Grafos

- Número máximo de arestas = V x (V – 1) / 2
  - Grafo completo : K_V
- Grau de um vértice
  - Número de arestas incidentes nesse vértice
  - Grau máximo ?
- Grafo regular
  - Todos os vértices têm o mesmo grau k : grafo *k-regular*

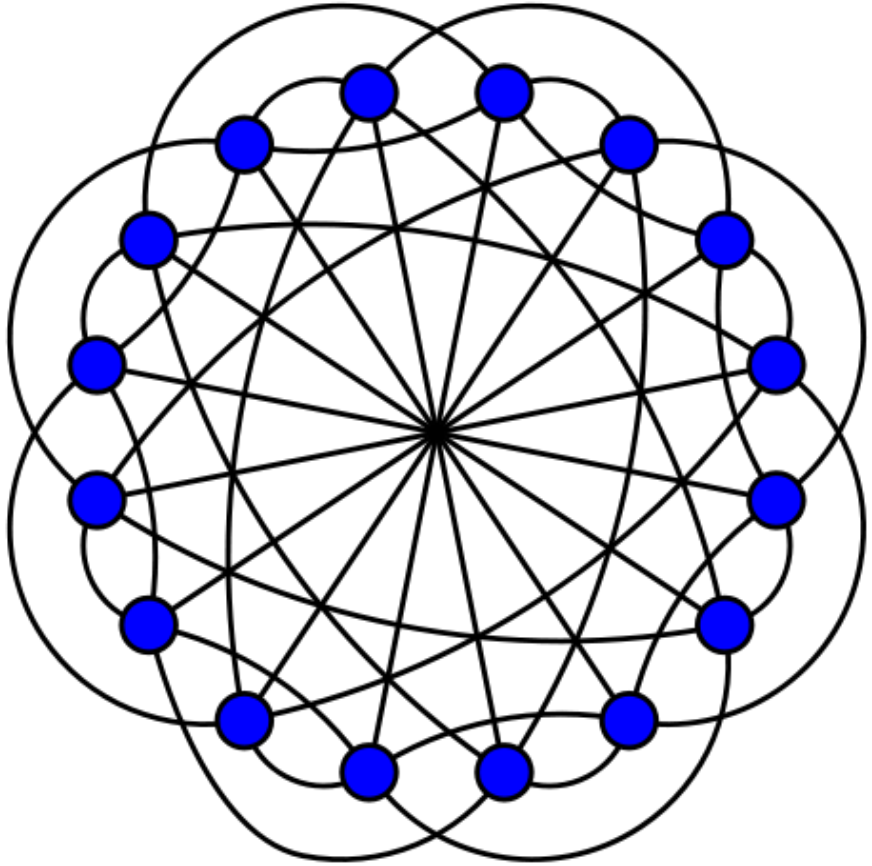
Grafo de Clebsch

16 vértices

40 arestas

5-regular

número cromático 4



[Wikipedia]

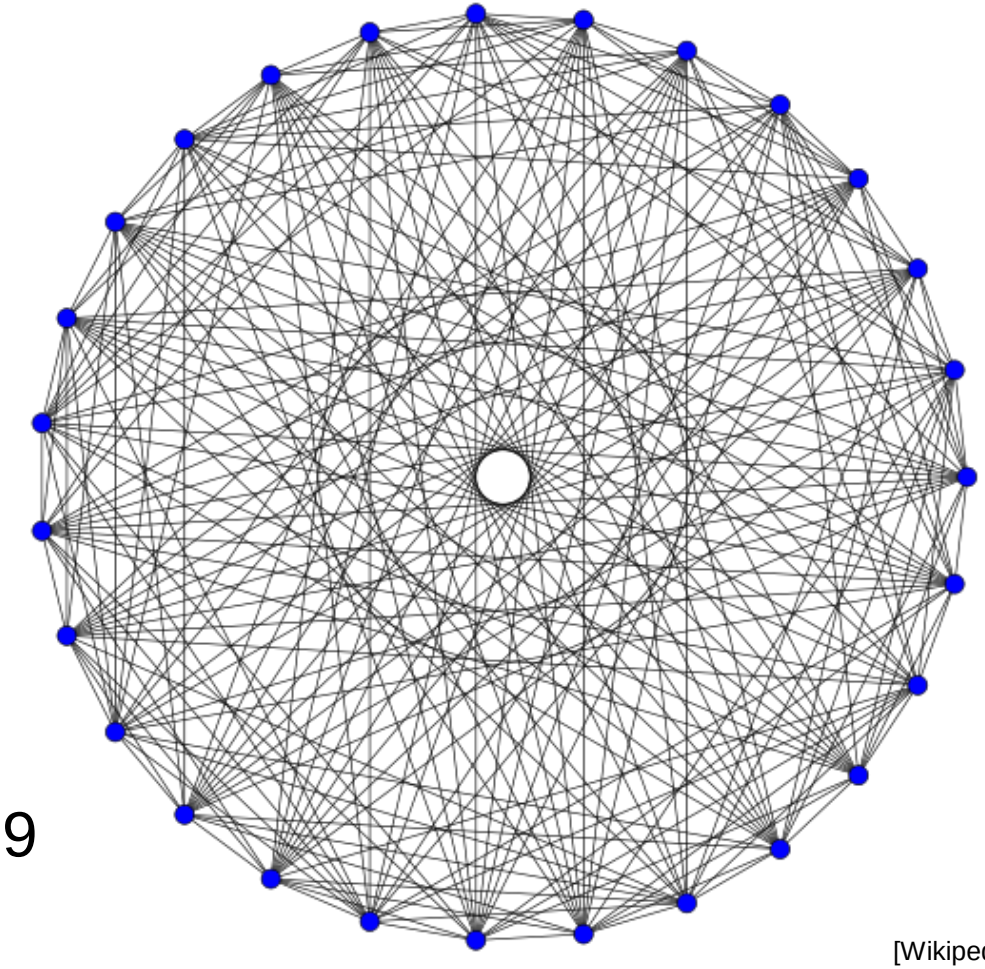
Grafo de Schläfli

27 vértices

216 arestas

16-regular

número cromático 9



[Wikipedia]

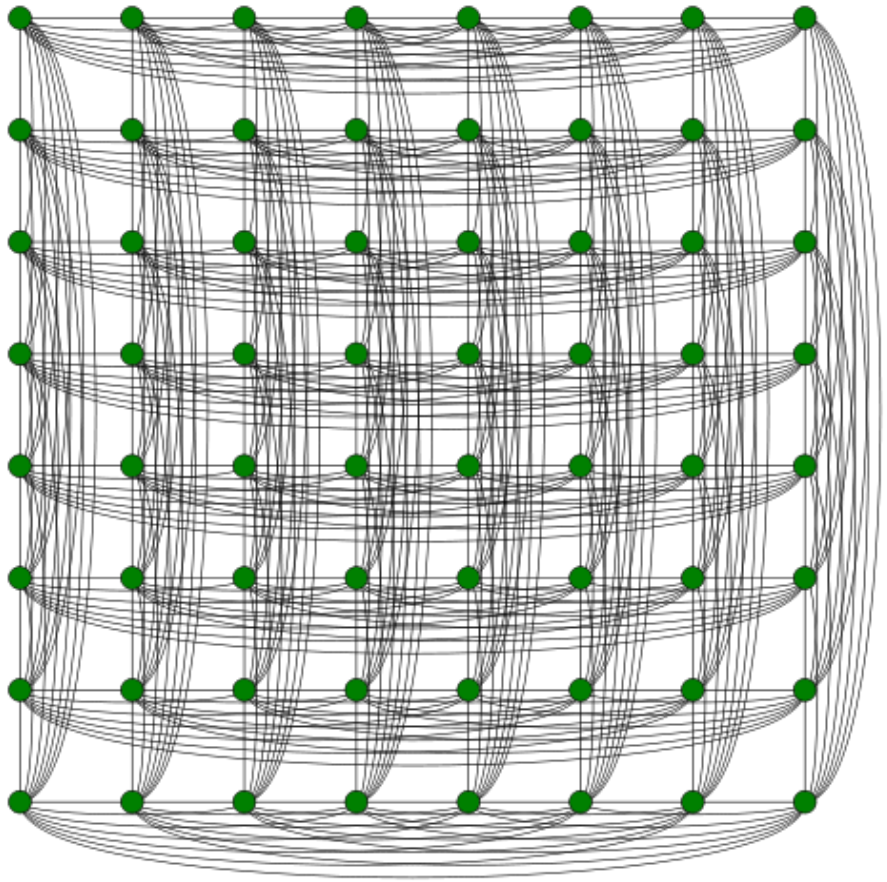
Grafo dos movimentos da torre no xadrez

64 vértices

448 arestas

14-regular

número cromático 8



[Wikipedia]

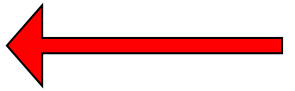
Grafos

$$\sum grau(v) = 2 \times E$$

- Grau médio – Average vertex degree

$$AVD = (2 \times E) \ / \ V$$

- Grafo denso : E em $\Theta(V^2)$ e AVD em $\Theta(V)$
  - Grafos densos vs. grafos esparsos



- Densidade

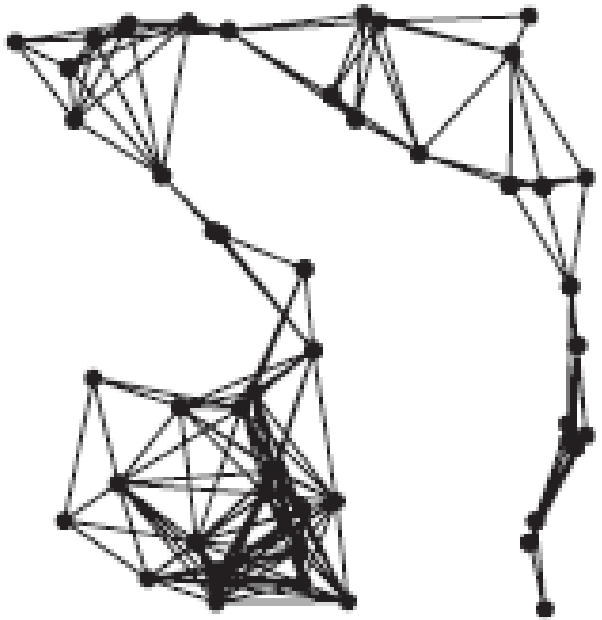
$$D = (2 \times E) \ / \ (V \times (V - 1))$$

Densidade – Exemplos

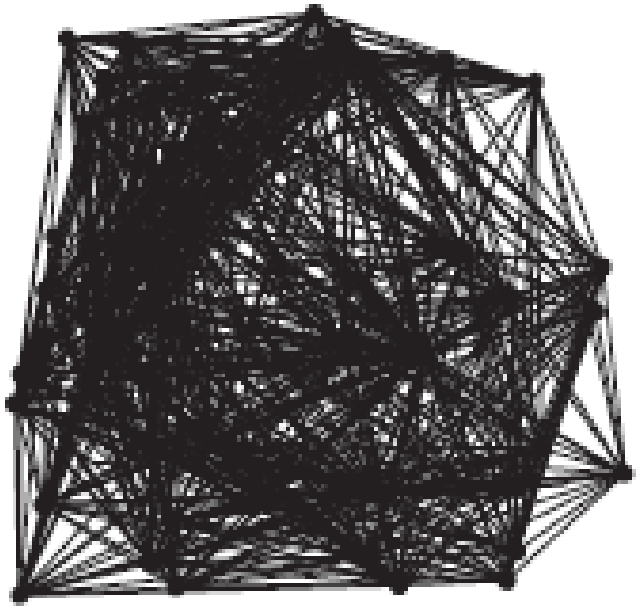
- Grafo dos movimentos da torre no xadrez
  - V = 64, E = 448 → D = 0.22
- Grafo de Clebsch
  - V = 16, E = 40 → D = 0.33
- Grafo de Schläfli
  - V = 27, E = 216 → D = 0.61

Densidade – Exemplo

sparse (E = 200)



dense (E = 1000)

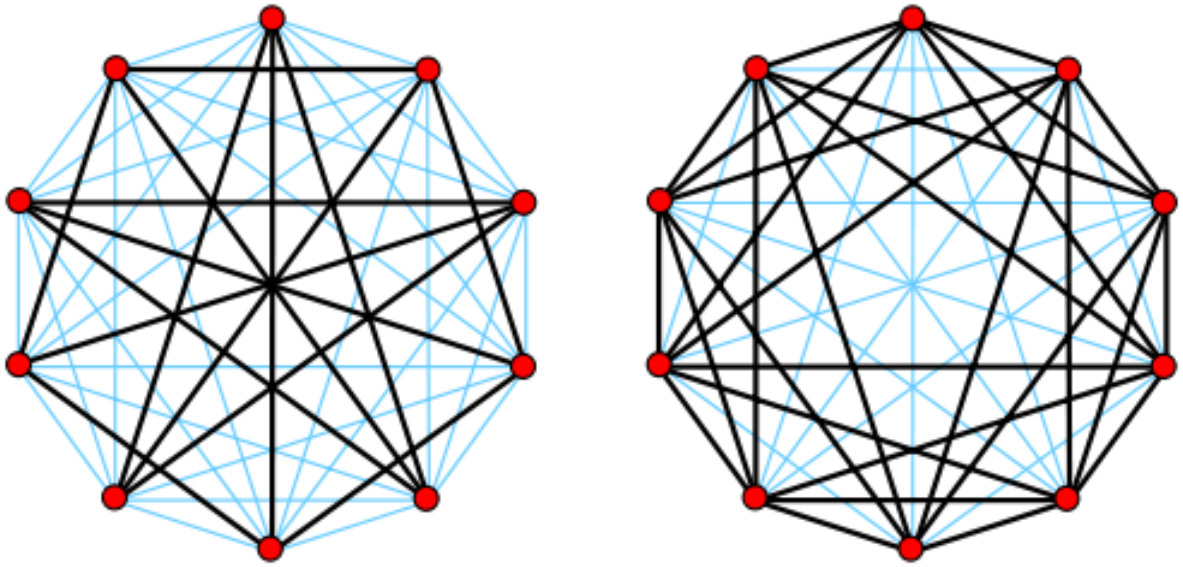


Two graphs (V = 50)

[Sedgewick & Wayne]

Grafo Complementar

- H(V,E') é o grafo complementar do grafo G(V,E)
  - Dois vértices são adjacentes em H sse não são adjacentes em G

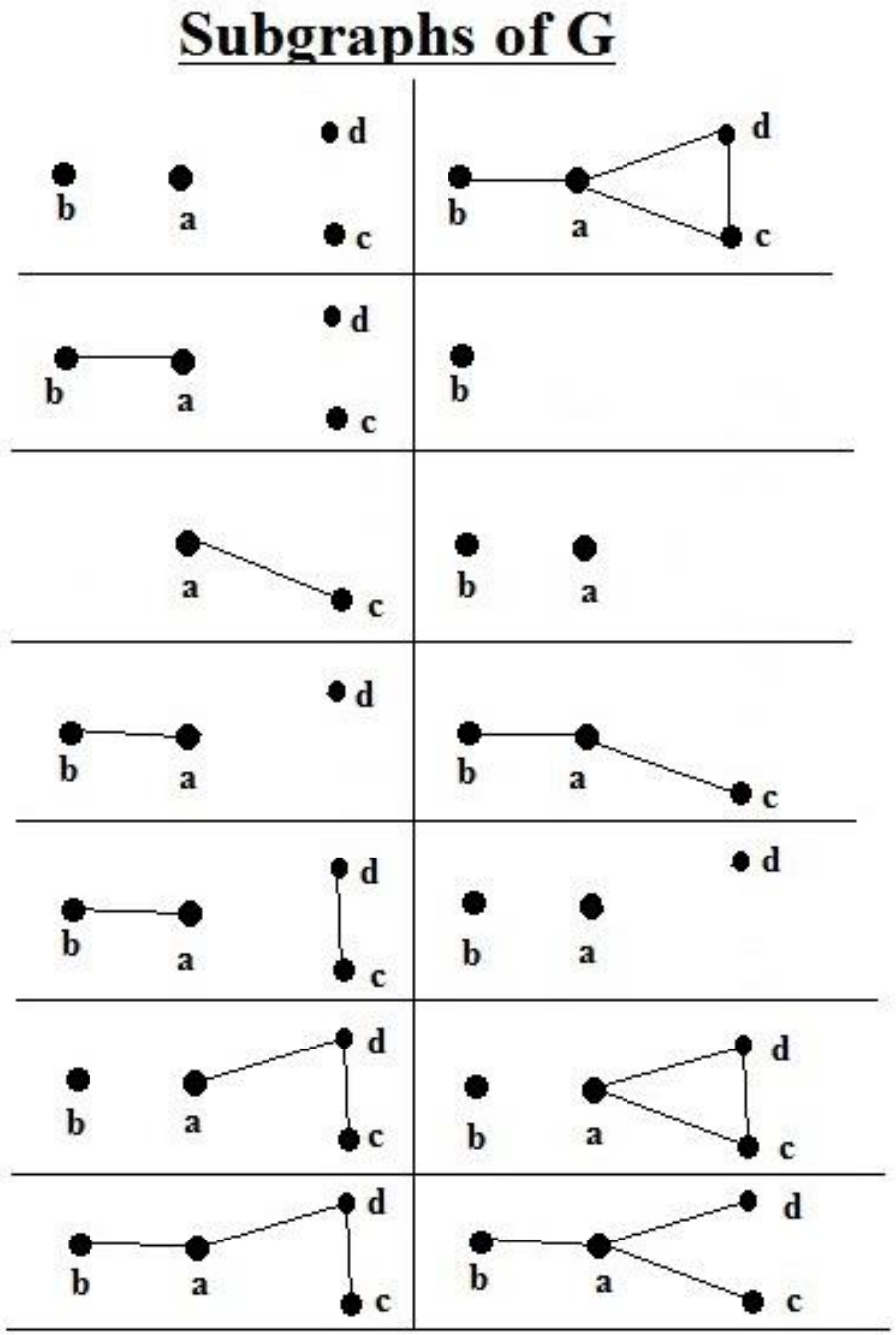


[Wikipedia]

Grafos

- G’(V’,E’) é um subgrafo de G(V,E)
  - E’ é um subconjunto de E
  - As arestas em E’ e os vértices associados constituem um grafo
- G e G’ são idênticos ou isomorfos
  - Renomeando o conjunto dos vértices de G (ou G’) torna o seu conjunto de arestas idêntico a E’ (ou E) ?

Exemplo

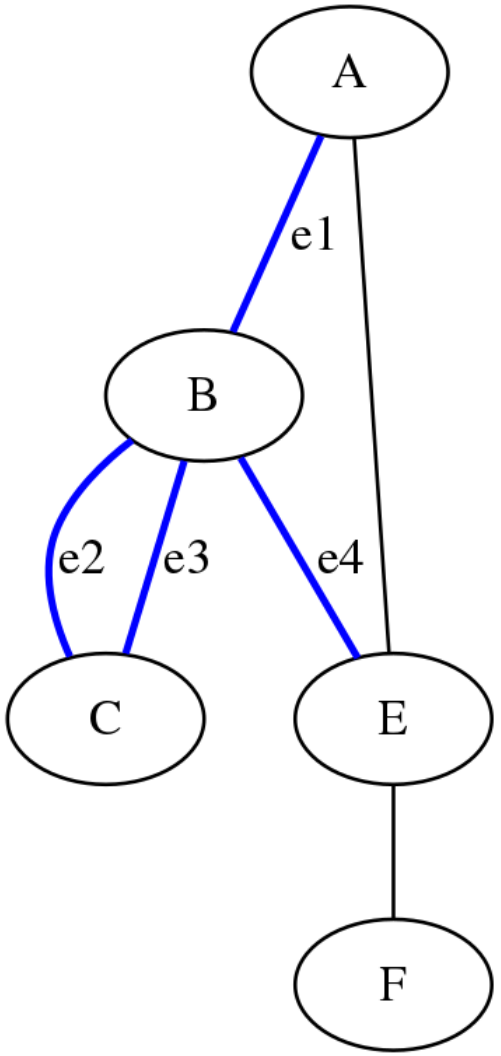


[math.stackexchange.com]

Grafos

- Um passeio é uma qualquer sequência de vértices adjacentes
  - Comprimento do passeio: nº de arestas que o constituem
- Um trajeto é um passeio constituído por arestas distintas
  - Um circuito é um trajeto de comprimento não nulo, que começa e acaba no mesmo vértice
- Um caminho é um passeio constituído por arestas e vertices distintos
  - Um ciclo é um caminho de comprimento não nulo, que começa e acaba no mesmo vértice

Exemplo

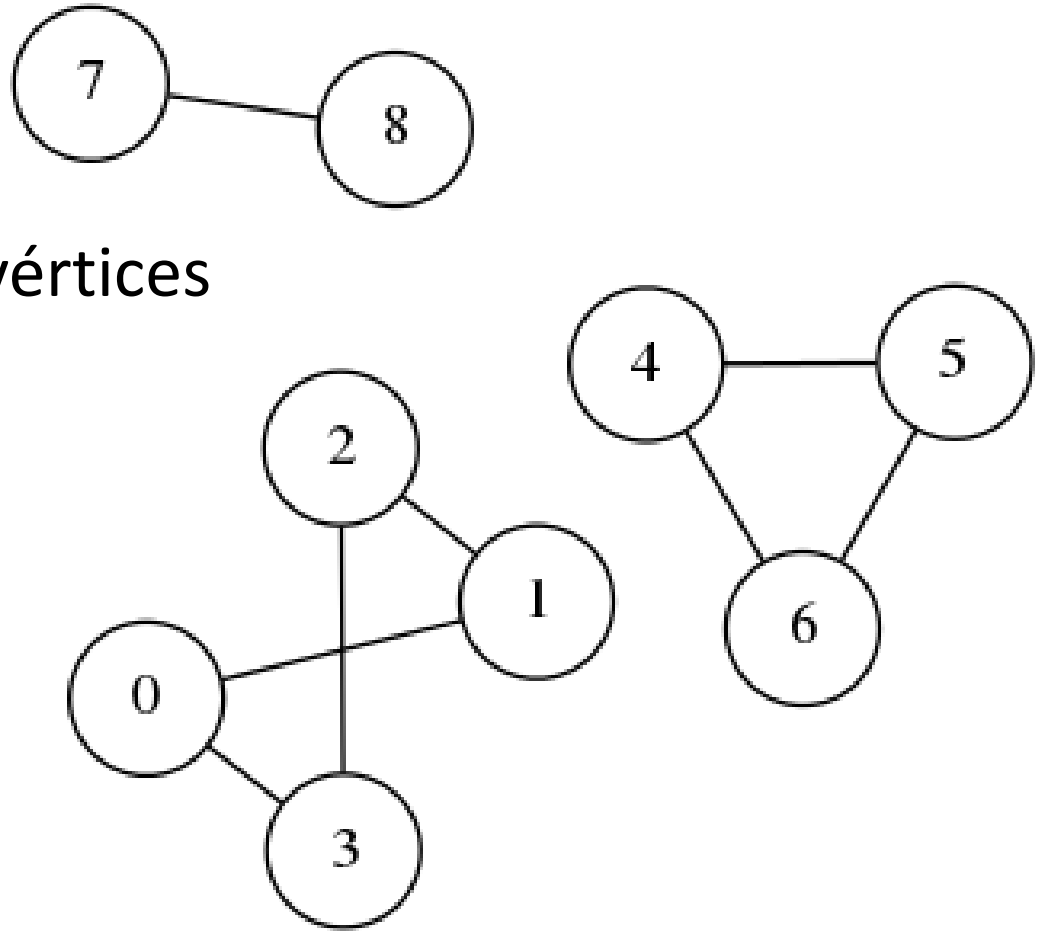


Trail from A to E, but not path (B vertex repeated)

[Wikipedia]

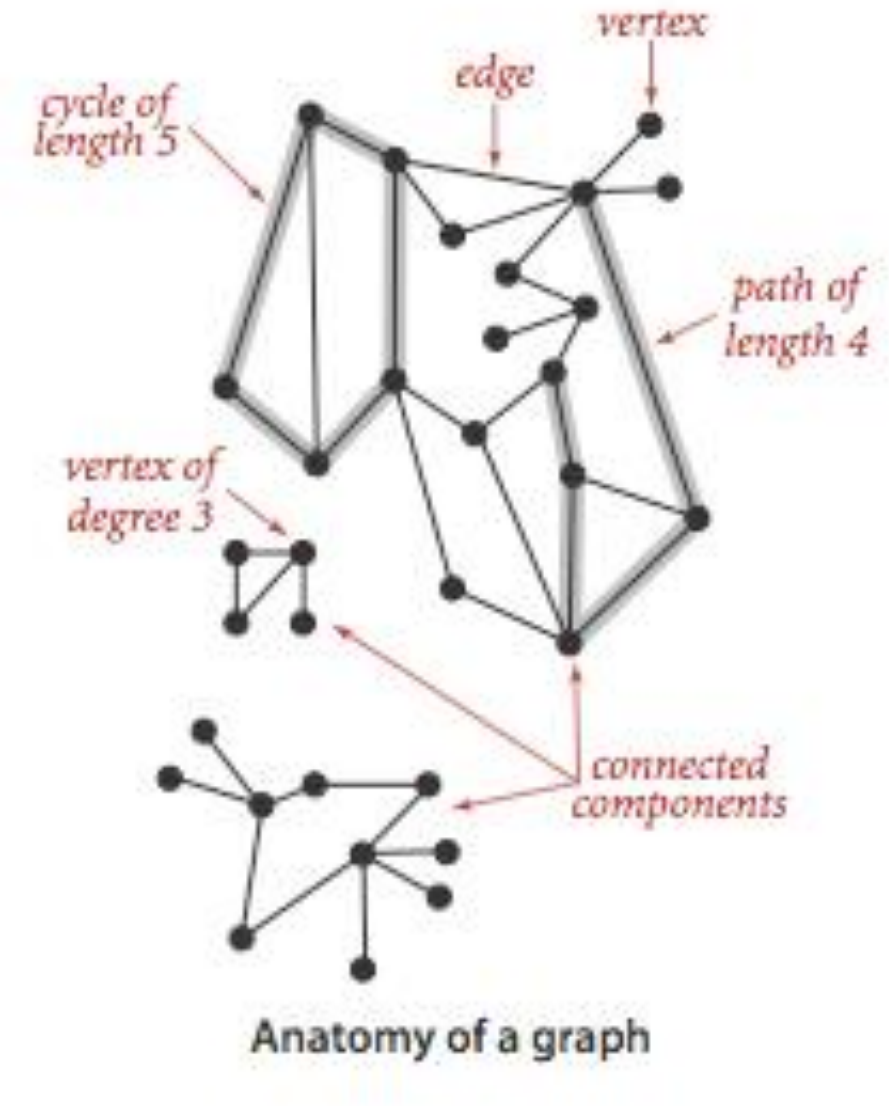
Grafos

- Grafo conexo
  - Existe um caminho entre cada par de vértices
  - Um único componente conexo



[martinbroadhurst.com/]

Grafos



[Sedgewick & Wayne]

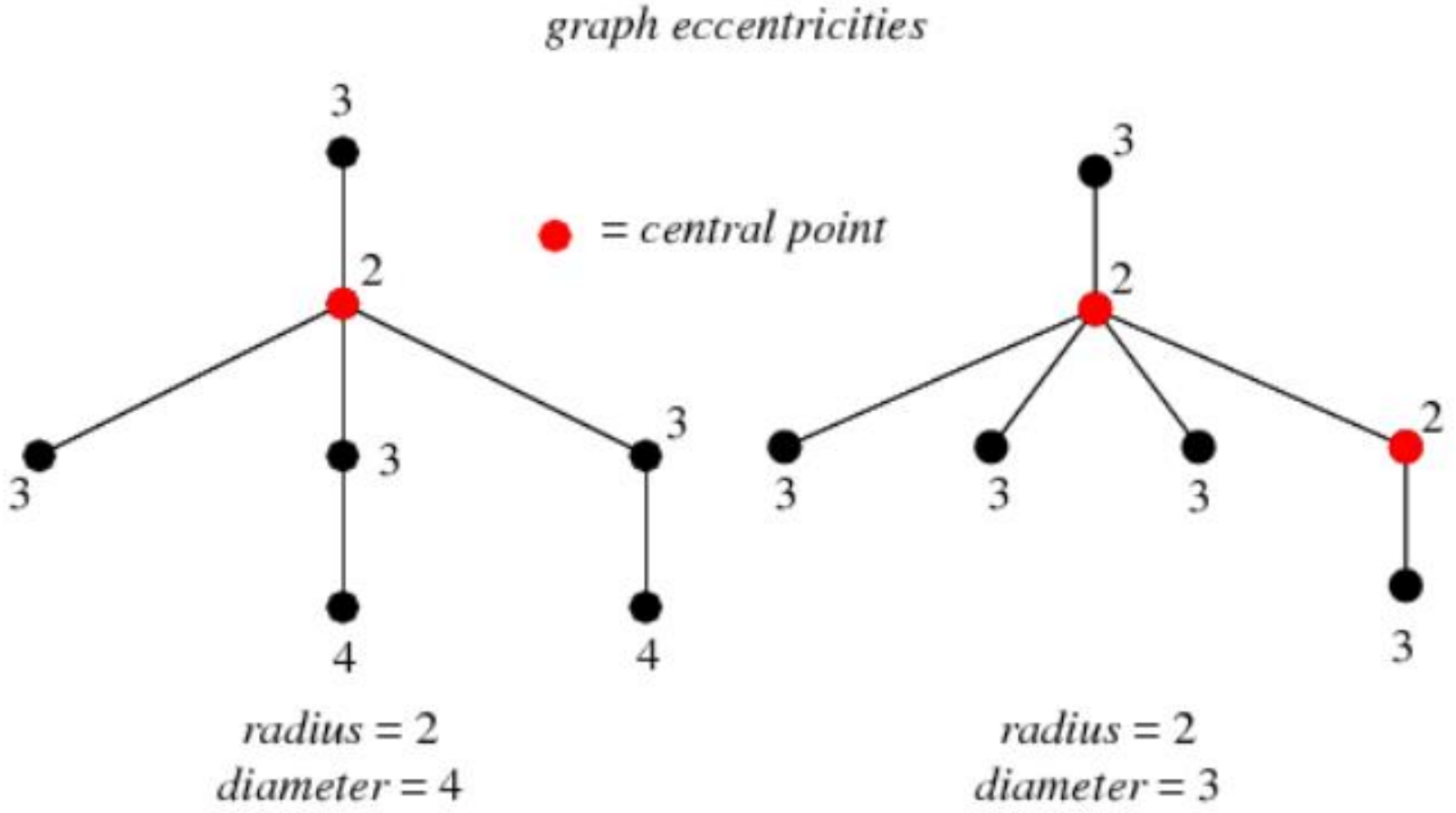
Grafos – Distâncias

- Distância entre dois vértices v_j e v_k
  - Número de arestas definindo um caminho mais curto ligando os dois vértices, caso exista
- Cintura ("girth") de um grafo G
  - Comprimento de um ciclo mais curto definido em G, caso exista
- Excentricidade do vértice v_j : $\varepsilon (v_j)$
  - Maior distância para qualquer outro vértice v_k

Grafos – Distâncias

- Raio de um grafo G
  - Menor valor de excentricidade dos vértices de G
- Diâmetro de um grafo G
  - Maior valor de excentricidade dos vértices de G
  - I.e., a maior distância entre qualquer par de vértices
- v_j é um vértice central num grafo de raio *r*
  - $\varepsilon(v_j) = r$
- v_j é um vértice periférico num grafo de diâmetro *d*
  - $\varepsilon(v_j) = d$

Exemplo



[Mathworld]

Grafo completo

K_7

7 vértices, 21 arestas

raio 1

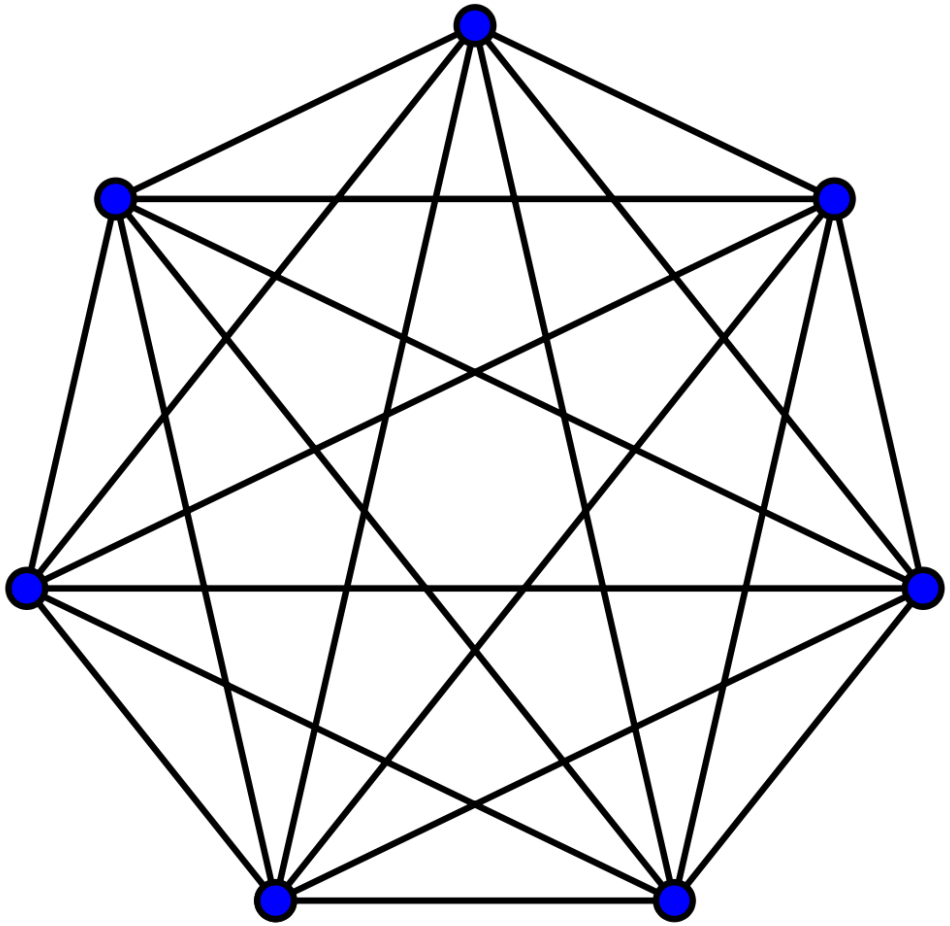
diâmetro 1

cintura 3

6-regular

número cromático 7

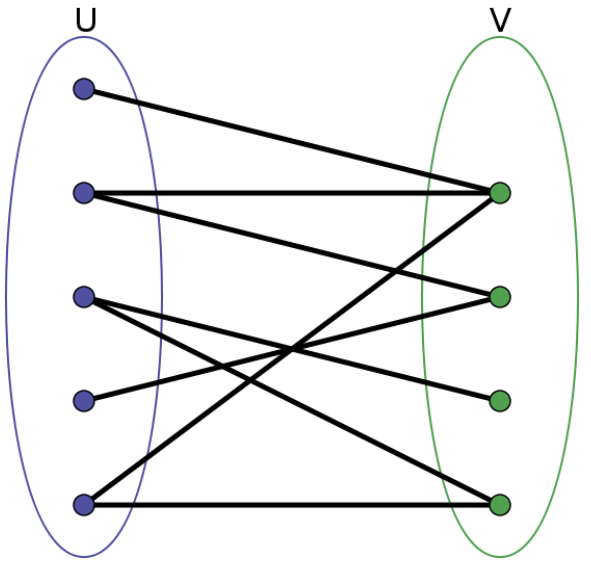
índice cromático 7



[Wikipedia]

Grafo bipartido

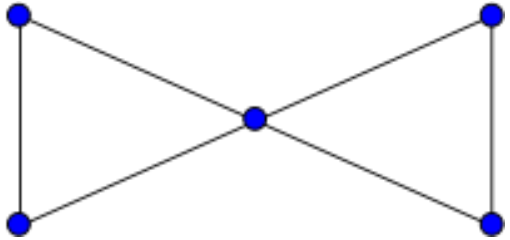
- Vértices divididos em dois conjuntos disjuntos
- Todas as arestas ligam um vértice de um dos conjuntos a um vértice do outro conjunto
- Aplicação:
  - Modelar problemas de emparelhamento / afetação
- Grafo bipartido completo ?

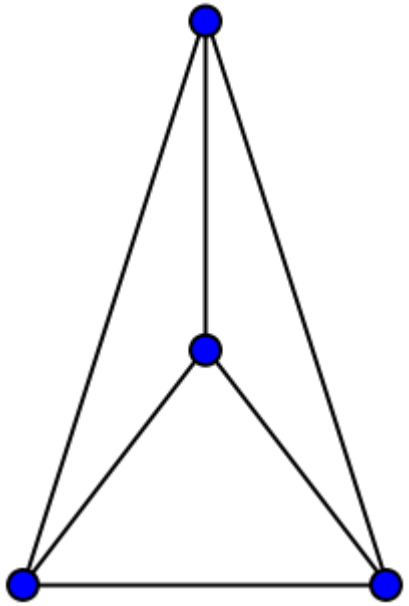


[Wikipedia]

Grafo planar

- Um grafo que pode ser embebido no plano !

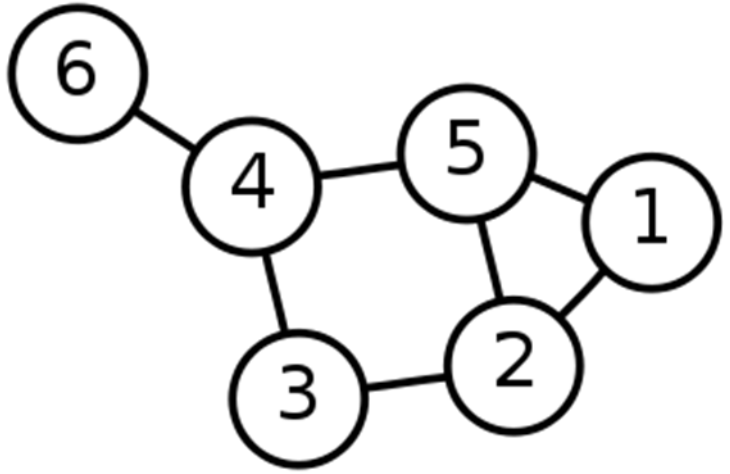




[Wikipedia]

- Aplicações
  - Projeto de circuitos VLSI
  - …

Grafos



[Wikipedia]

– Alguns problemas

Caminhos

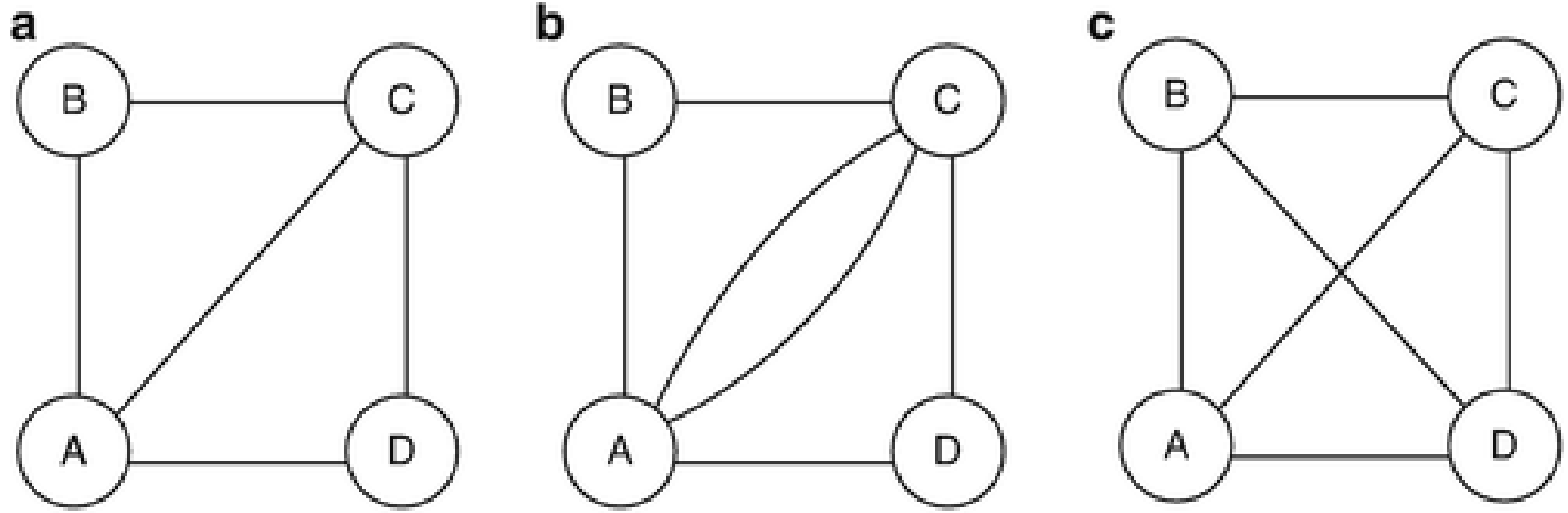
- Há um caminho entre os vértices s e t ?
  - Usar a procura em profundidade – Depth-First Search (DFS)

- Qual é o caminho mais curto entre s e t ?
  - Caminho usando o menor número de arestas
  - Usar a procura por níveis – Breadth-First Search (BFS)

Ciclos / Circuitos

- G(V,E) tem algum ciclo ?
  - Usar DFS !
- G(V,E) tem um circuito que usa, uma única vez, cada uma das arestas de G ?
  - Euler Tour
- G(V,E) tem um ciclo que usa, uma única vez, cada um dos vértices de G ?
  - Hamilton Tour

Euler Path / Euler Circuit



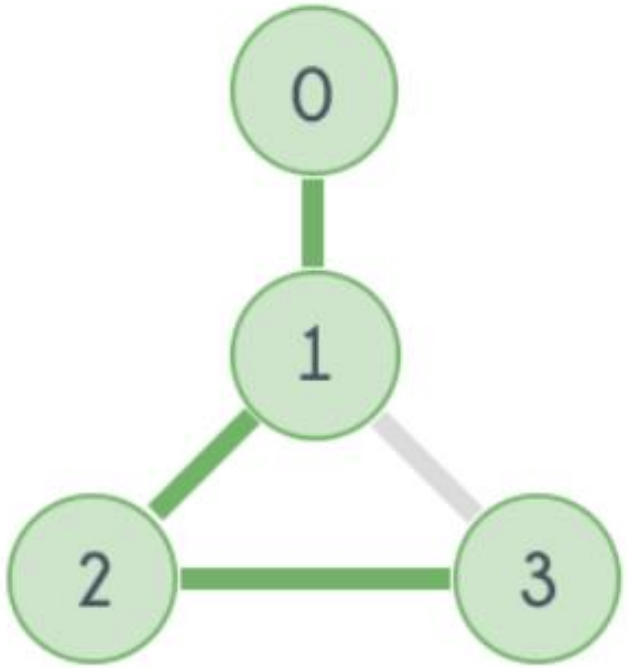
Eulerian Path:
A–B–C–D–A–C

Eulerian Cycle:
A–B–C–D–A–C–A

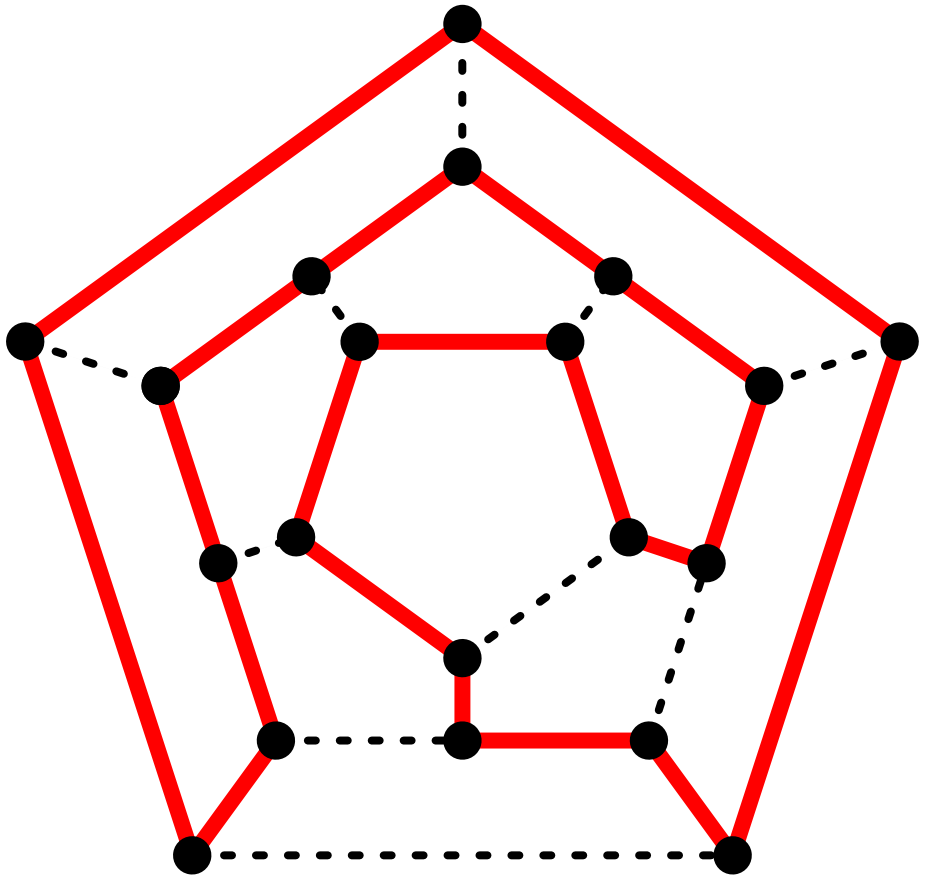
Neither eulerian path
nor cycle exist

[SpringerLink]

Hamiltonian Path / Hamiltonian Cycle



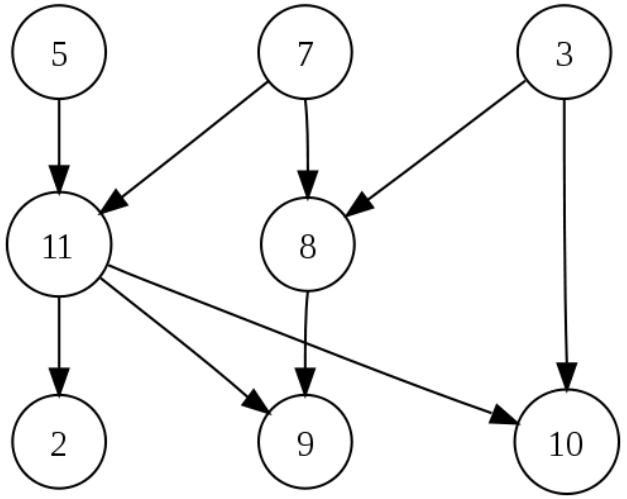
[hackerearth.com]



[Wikipedia]

Grafos Orientados

Grafos orientados



[Wikipedia]

- G(V,E)
- Grafo orientado
  - As arestas orientadas definem uma adjacência unidirecional
- $e_i = (v_j, v_k)$
  - v_j é o vértice origem e v_k o vértice destino
  - v_k é adjacente a v_j
  - e_i é incidente em v_k

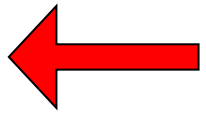
Aplicações

Digraph	Vertex	Directed Edge
transportation	street intersection	one-way street
web	web page	hyperlink
food web	species	predator-prey relationship
scheduling	task	precedence constraint
financial	bank	transaction
cell phone	person	placed call
infectious disease	person	infection
game	board position	legal move
citation	journal article	citation
object class	object	pointer
inheritance hierarchy	class	inherits from
control flow	code block	jump

[Sedgewick/Wayne]

Grafos orientados

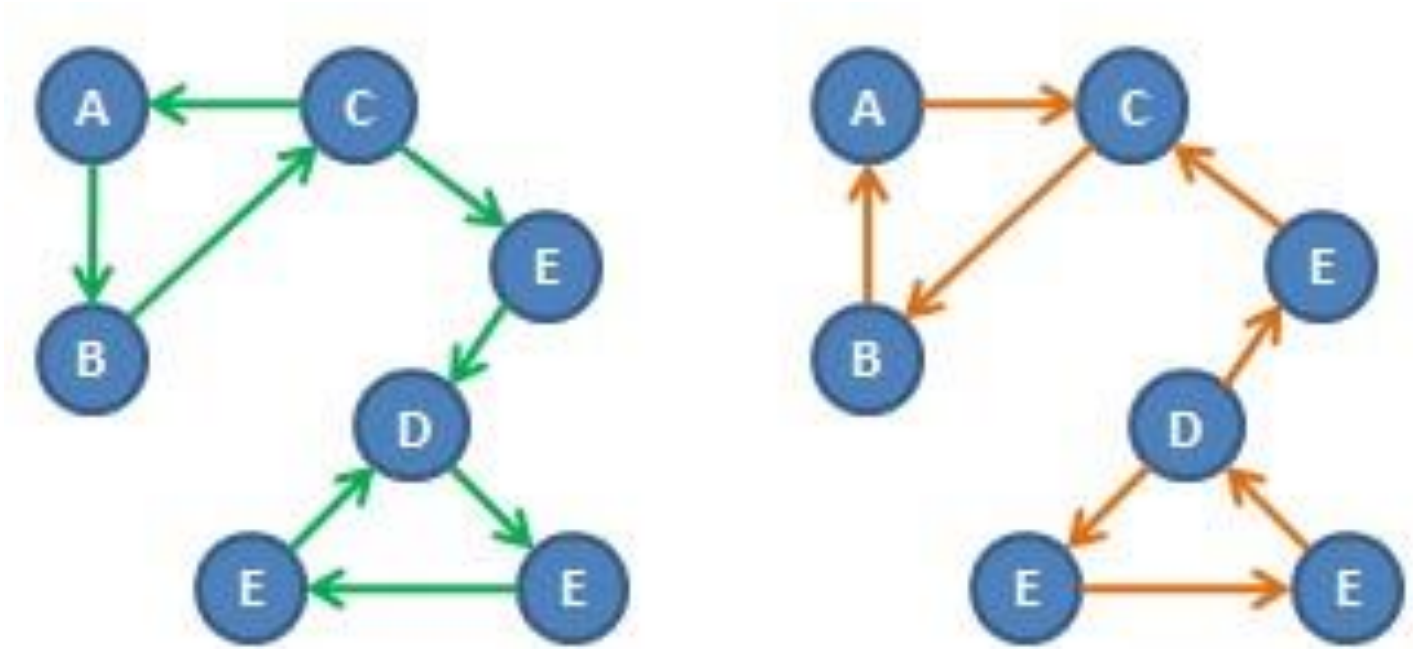
- Nº máximo de arestas = V x (V – 1)
  - Grafo orientado completo
- In-Degree e Out-Degree associado a cada vértice
  - Vértice fonte ?
  - Vértice sumidouro ?
- Densidade de um grafo orientado
  - Grafos orientados densos vs. esparsos



Grafos orientados

- G’(V,E’) é o grafo transposto de um grafo orientado G(V,E)
  - Para cada aresta orientada $e_i = (v_j, v_k)$ em E, a sua aresta simétrica $e_i’ = (v_k, v_j)$ existe em E’
- Grafo orientado simétrico
  - Para cada aresta orientada $e_i = (v_j, v_k)$, a sua aresta simétrica $e_i’ = (v_k, v_j)$ também existe
  - Permite representar um grafo como um grafo orientado

Exemplo



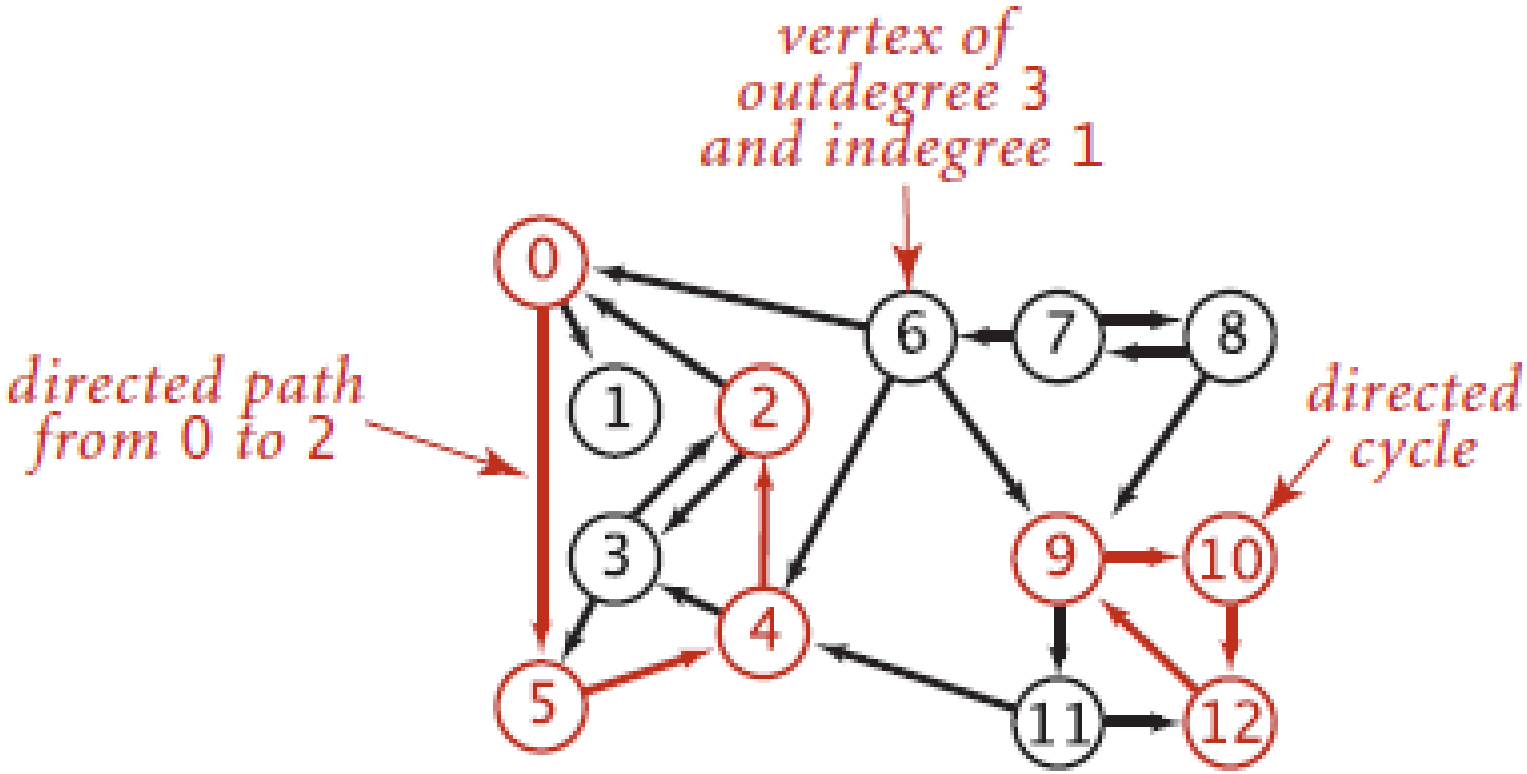
[Wikipedia]

Grafos orientados

- Um passeio orientado é uma sequência de vértices
  - Cada vértice (exceto o primeiro) é adjacente ao seu predecessor
- Caminho orientado : arestas e vértices distintos
- Ciclo orientado : caminho orientado com o mesmo vértice inicial e final

- Vértice t é alcançável a partir do vértice s ?
  - Existe um caminho de s para t

Grafo orientado

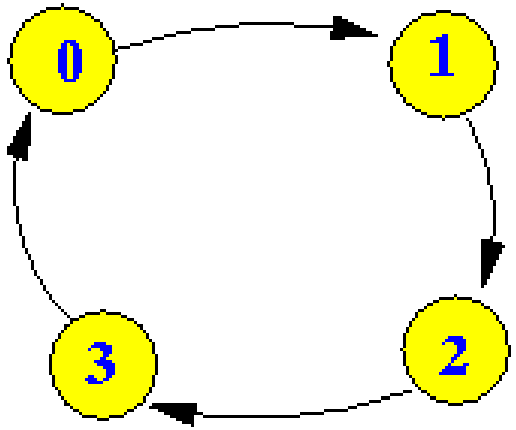


[Sedgewick/Wayne]

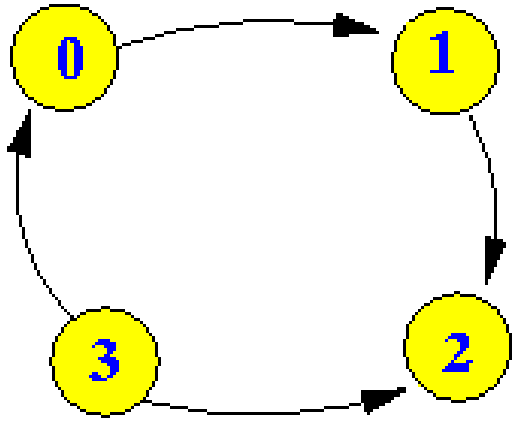
Grafos orientados

- Grafo orientado fracamente conexo
  - Substituir as arestas orientadas por arestas não-orientadas
  - O grafo resultante é conexo
- Grafo orientado fortemente conexo
  - Existe um caminho entre cada par de vértices
    - Vértices mutuamente alcançáveis !!
  - Um único componente fortemente conexo

Exemplo



Strongly connected

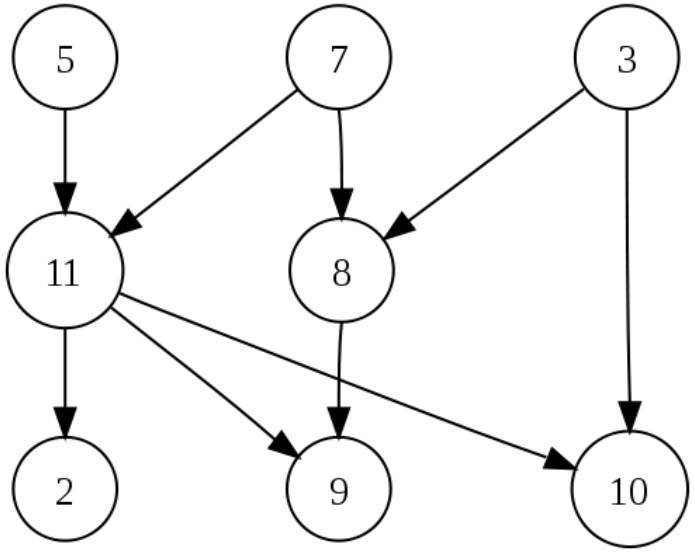


Not strongly connected

[cs.emory.edu]

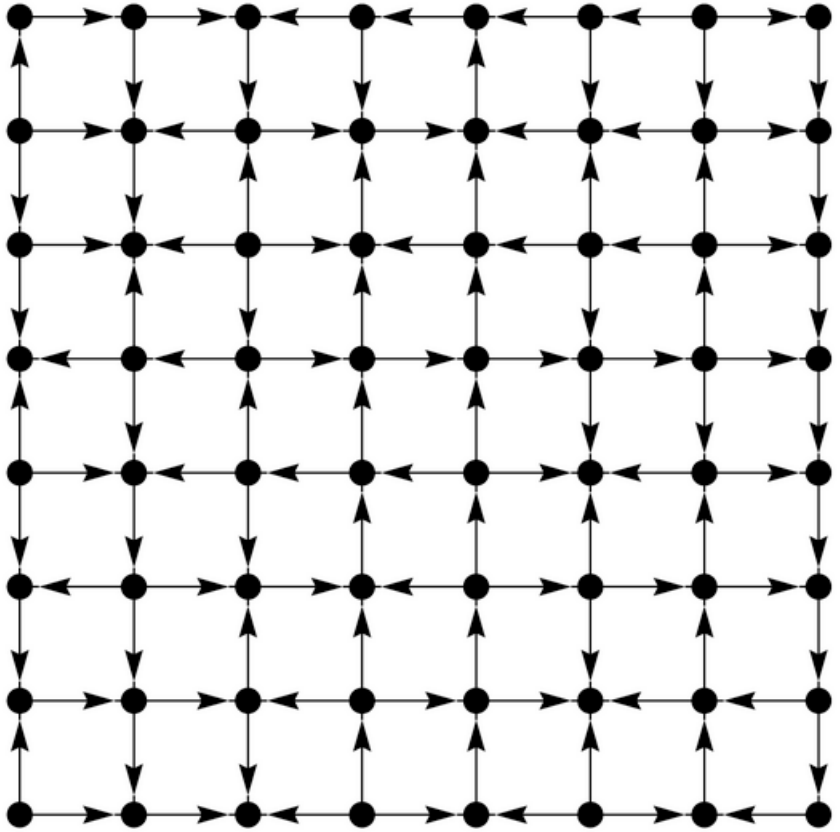
Grafos orientados acíclicos

- Directed Acyclic Graphs (DAGs)
- Um grafo orientado que não contém qualquer ciclo !
  - Relações de precedência



[Wikipedia]

Grid Digraph



[Sedgewick/Wayne]

Grafos Orientados – Alguns problemas

Caminhos

- Existe um caminho orientado entre os vértices s e t ?

- Qual é o caminho orientado mais curto entre s e t ?
  - Caminho usando o menor número de arestas

Vértices Alcançáveis

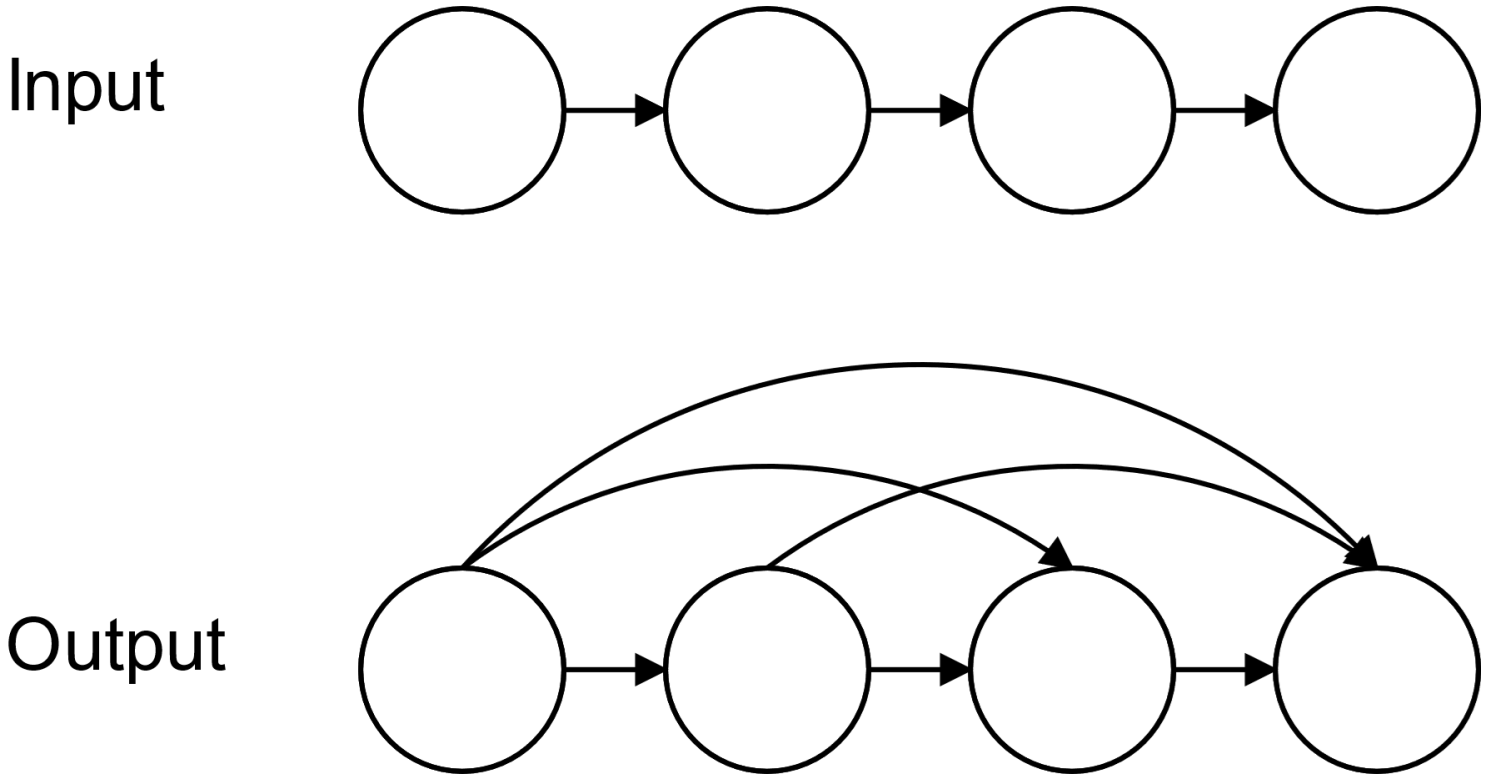
- Encontrar todos os vértices alcançáveis a partir de um vértice s ?
  - Usar DFS !

- Todos os vértices são mutuamente alcançáveis ?
  - Usar, repetidamente, DFS !

Fecho Transitivo

- Para que vértices v e w há um caminho de v para w ?
  - Representar a resposta como um grafo orientado !
  - Usar, repetidamente, DFS !

Exemplo



[Wikipedia]

Ordenação Topológica

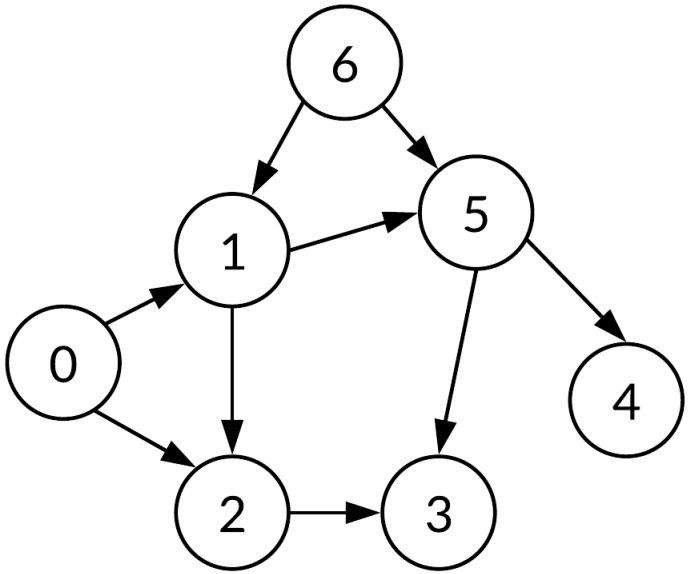
- Podemos desenhar um dado grafo orientado de maneira a que todas as arestas apontem para o mesmo lado ?

- Dado um conjunto de tarefas a realizar, e as respetivas precedências, qual a ordem pela qual devem ser escalonadas ?
  - Usar BFS ou DFS !
  - Representar a solução com um grafo orientado acíclico !

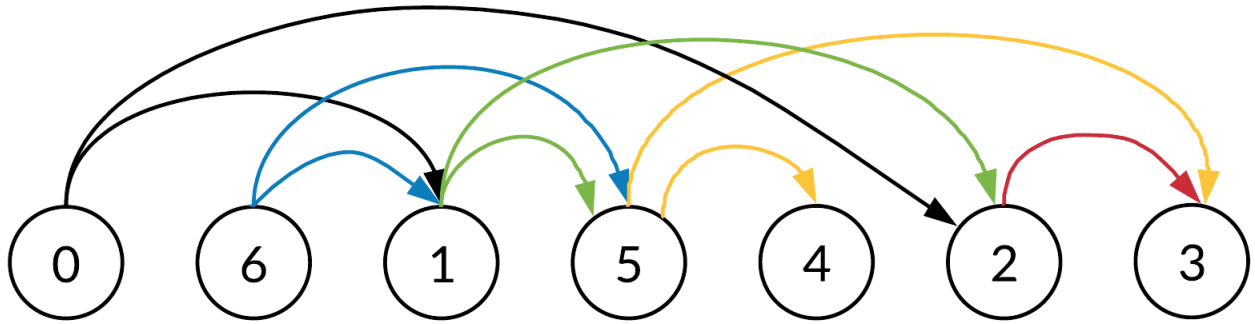
- Usar para verificar se um grafo orientado é acíclico ou não

Exemplo

Unsorted graph



Topologically sorted graph

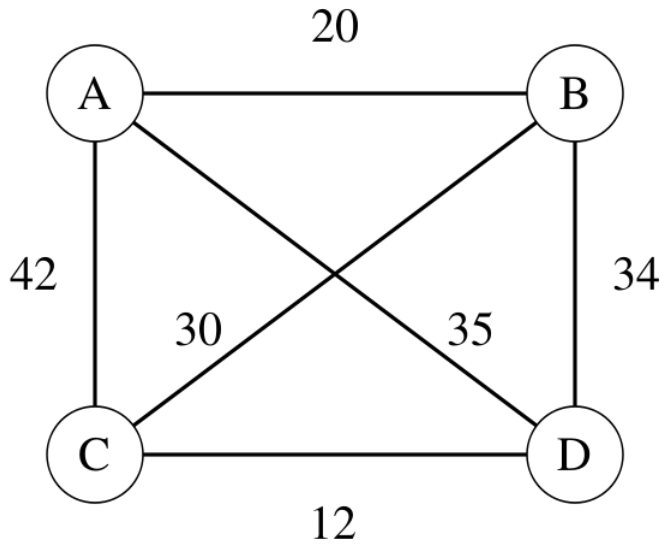


[guides.codepath.com]

Redes

Rede

- Uma rede é um grafo / grafo orientado com "pesos" associados às suas arestas
  - Weighted graph / digraph
  - Associar um ou mais valores a cada aresta
  - Custo, distância, capacidade, …



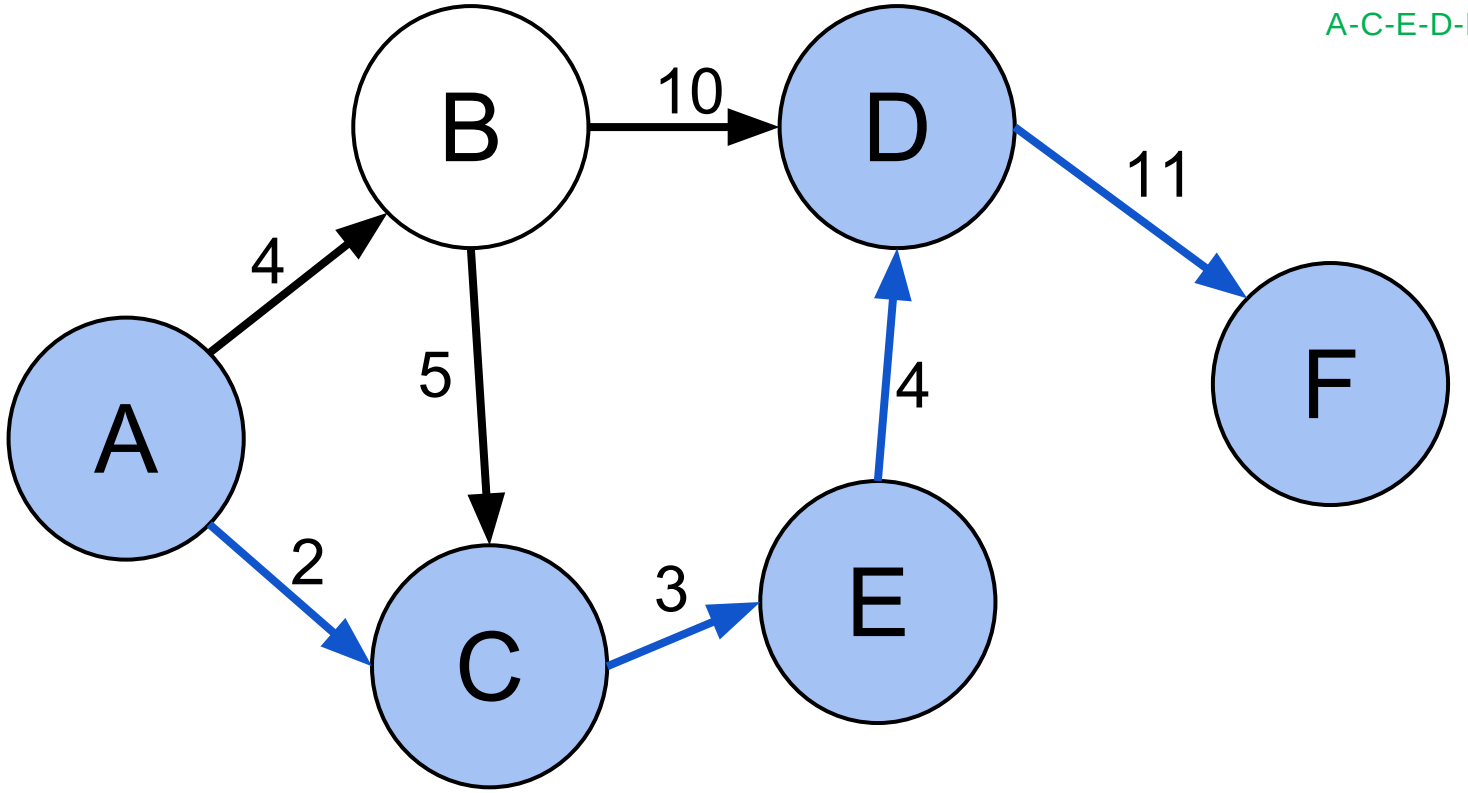
[Wikipedia]

Caminhos

- Existe um caminho entre os vértices s e t ?

- Qual é o caminho mais curto entre s e t ?
  - Soma das distâncias associadas a cada aresta

Caminho mais curto entre A e F – Custo ?



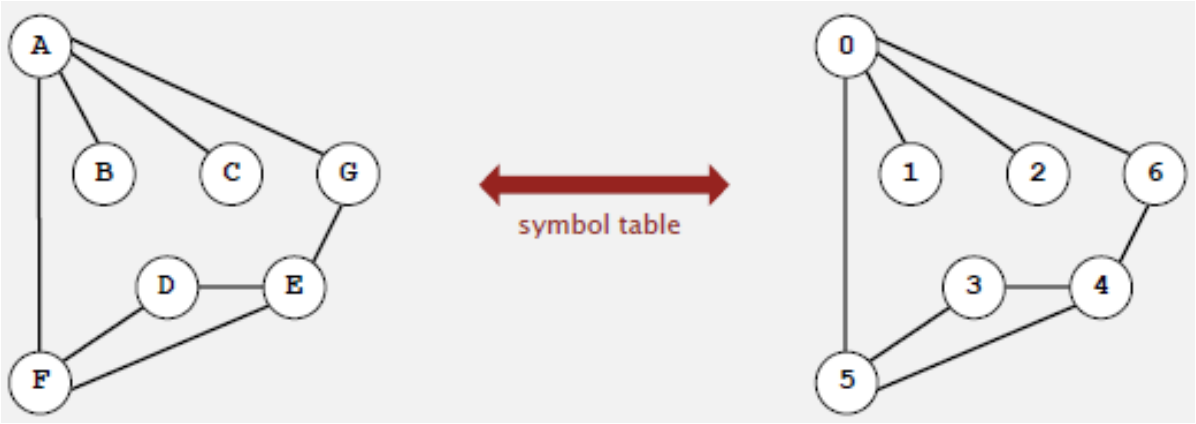
A-C-E-D-F -> custo = 24

[Wikipedia]

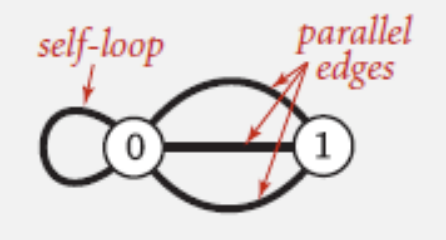
Como representar?
Que funcionalidades?

Vértices

- Identificados por um valor inteiro de 0 a V – 1
- Usar dicionários para mapear esses IDs noutros identificadores



- Não são permitidos lacetes nem arestas paralelas



[Sedgewick/Wayne]

TAD Grafo

```
Graph* GraphCreate(unsigned int V);          // Apenas com V vértices
GraphDestroy(Graph** g);
Graph* GraphCopy(Graph* g);
Graph* GraphFromFile(FILE* f);
unsigned int GraphGetNumVertices(Graph* g);
unsigned int GraphGetNumEdges(Graph* g);
…
```

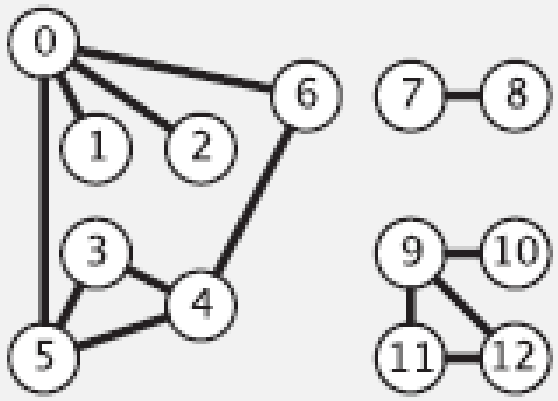
TAD Grafo

```
…
int GraphGetVertexDegree(Graph* g, unsigned int v);
int GraphGetDegree(Graph* g);
int GraphGetAverageDegree(Graph* g);
…
int GraphAddEdge(Graph* g, unsigned int v, unsigned int w);
List* GraphGetAdjacentTo(Graph*g, unsigned int v);
…
```

Representação em ficheiro

tinyG.txt

V → 13
E → 13
0 5
4 3
0 1
9 12
6 4
5 4
0 2
11 12
9 10
0 6
7 8
9 11
5 3

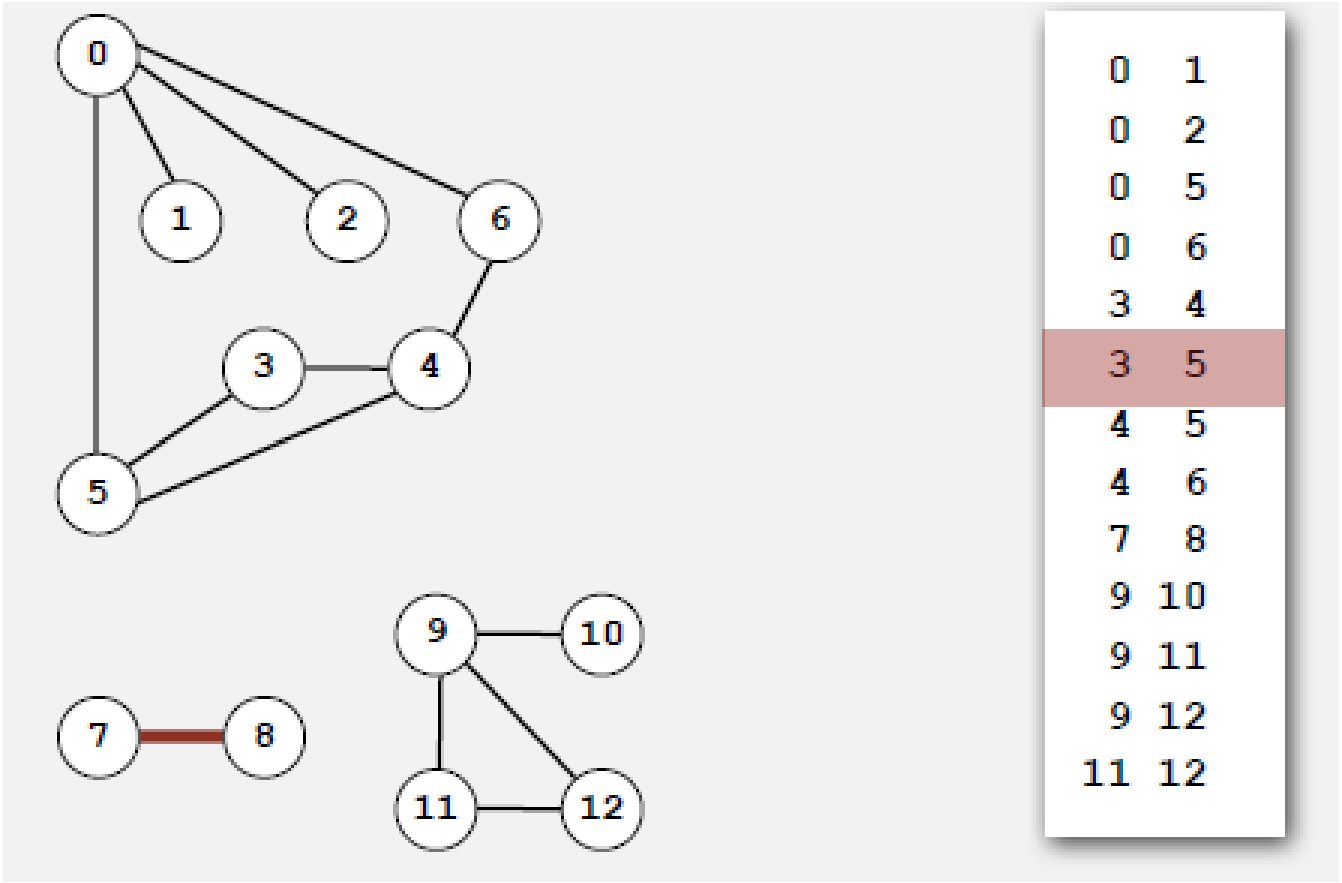



```
0-6
0-2
0-1
0-5
1-0
2-0
3-5
3-4
...
12-11
12-9
```

[Sedgewick/Wayne]

Representação – Conj. ordenado de arestas

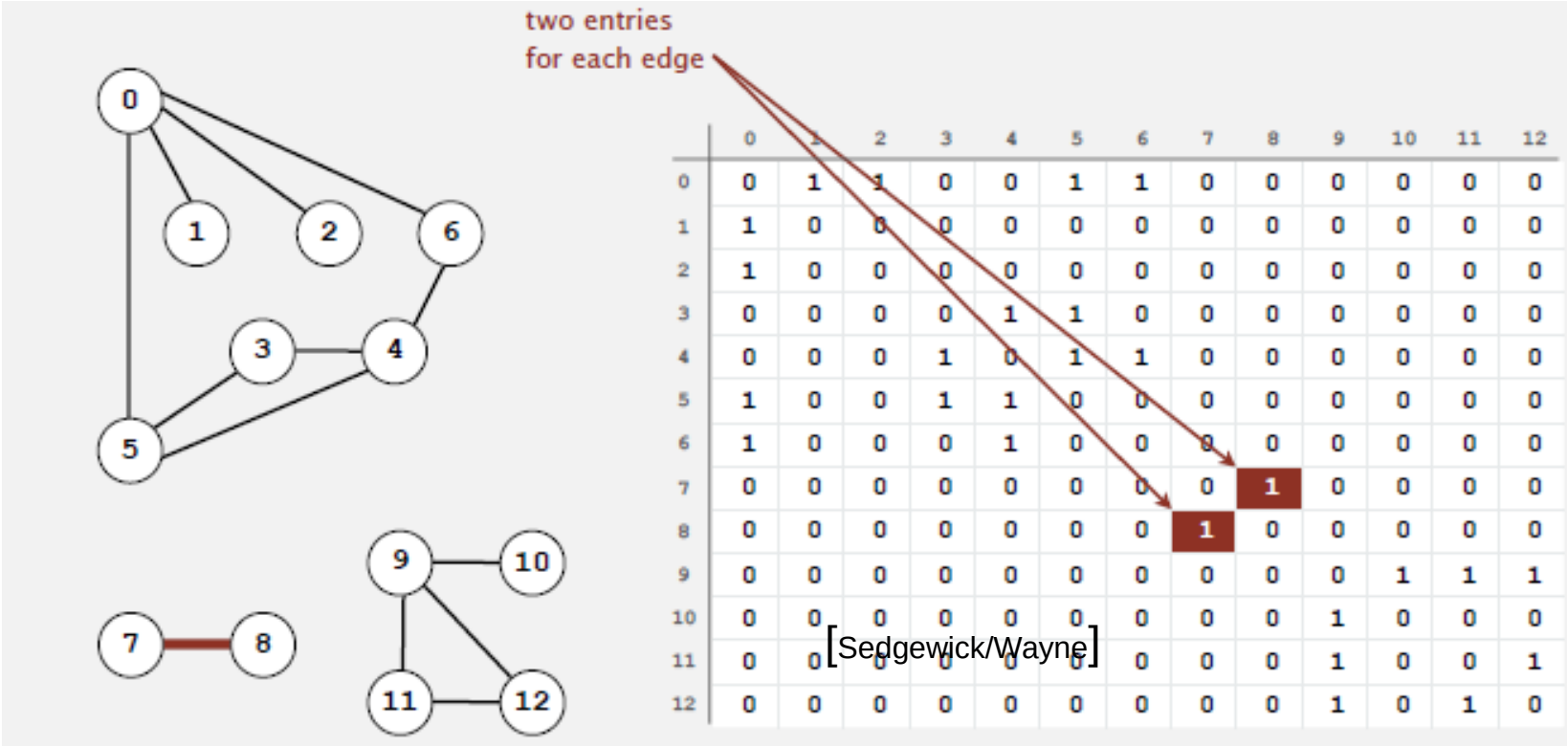
- Lista ligada de arestas



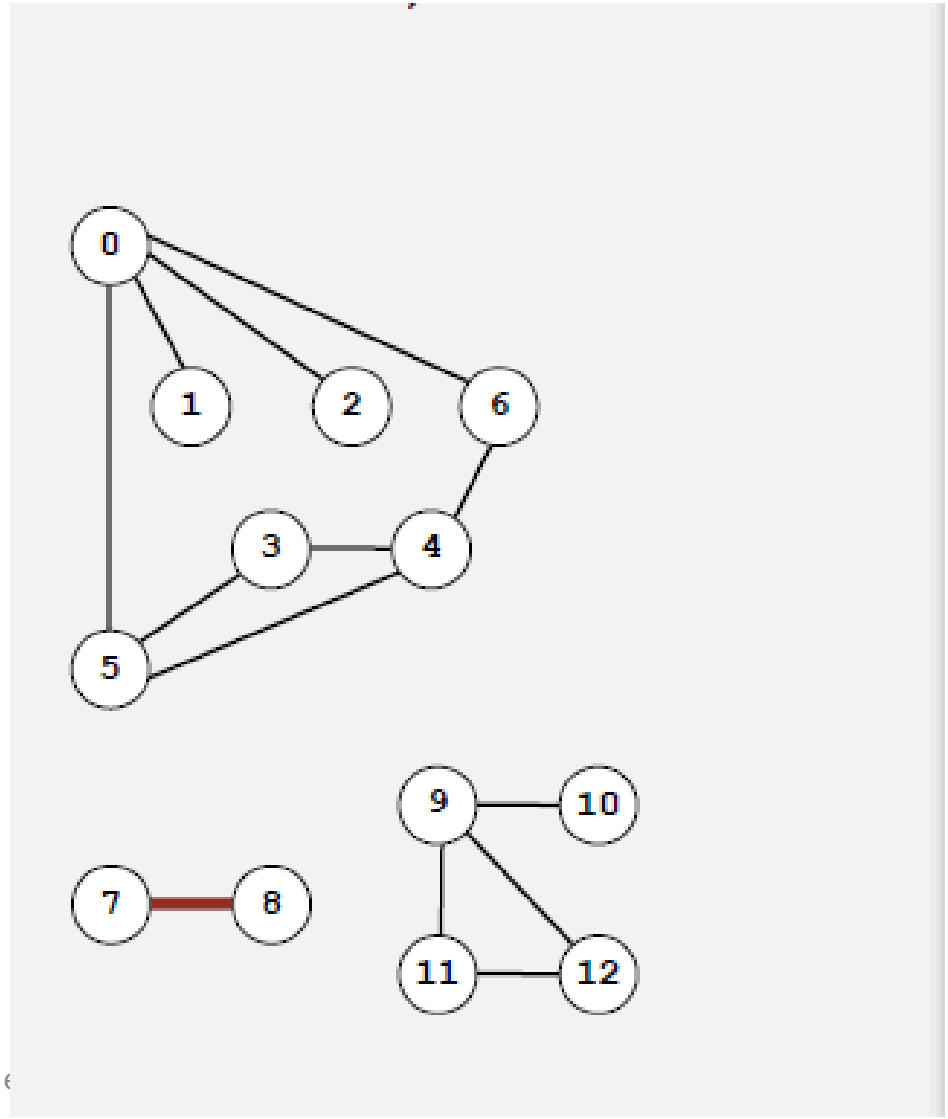
[Sedgewick/Wayne]

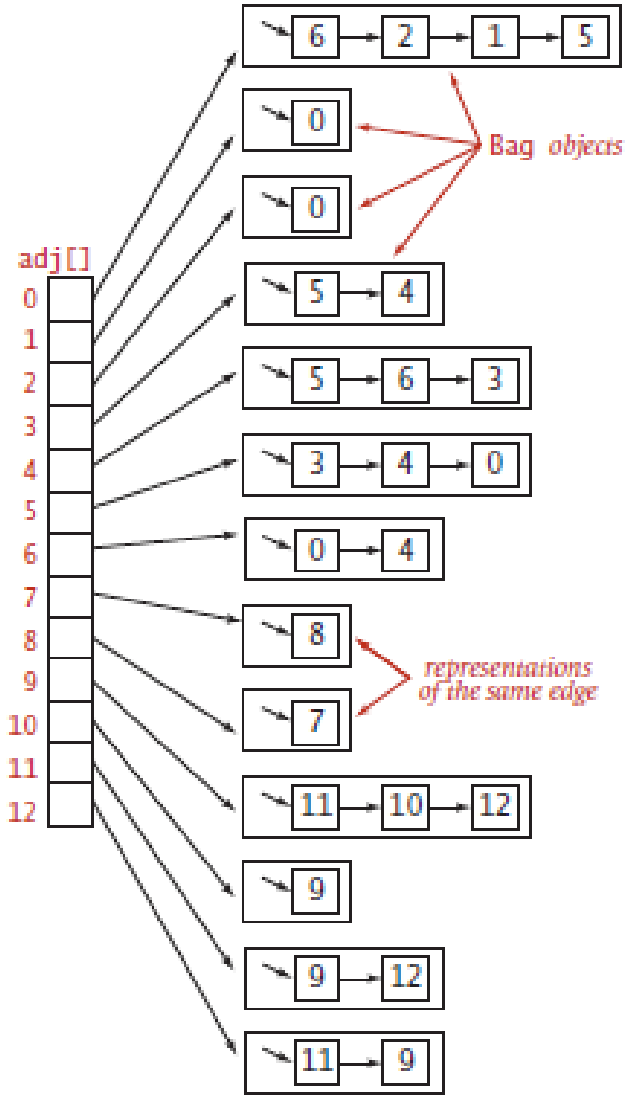
Representação – Matriz de adjacência

- Array de V^2 booleanos
- Cada aresta é representada duas vezes
  - Porquê ?



Representação – Listas de adjacências





[Sedgewick/Wayne]

Desempenho

- Na prática: usar a representação em listas de adjacências
- Os grafos do mundo real são habitualmente esparsos !!
- Algoritmos iteram sobre os vértices adjacentes a um vértice dado

representation	space	add edge	edge between v and w?	iterate over vertices adjacent to v?
list of edges	E	1	E	E
adjacency matrix	V^2	1 *	1	V
adjacency lists	E + V	1	degree(v)	degree(v)

* disallows parallel edges

[Sedgewick/Wayne]

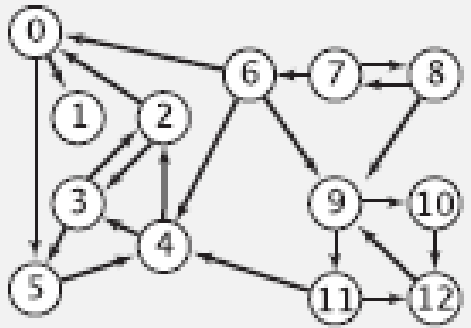
TAD Grafo Orientado

```
Digraph* DigraphCreate(unsigned int V);      // Apenas com V vértices
DigraphDestroy(Digraph** g);
Digraph* DigraphCopy(Digraph* g);
Digraph* DigraphCreateReverse(Digraph* g);
Digraph* DigraphFromFile(FILE* f);
unsigned int DigraphGetNumVertices(Digraph* g);
unsigned int DigraphGetNumEdges(Digraph* g);
…
```

Representação em ficheiro

tinyDG.txt

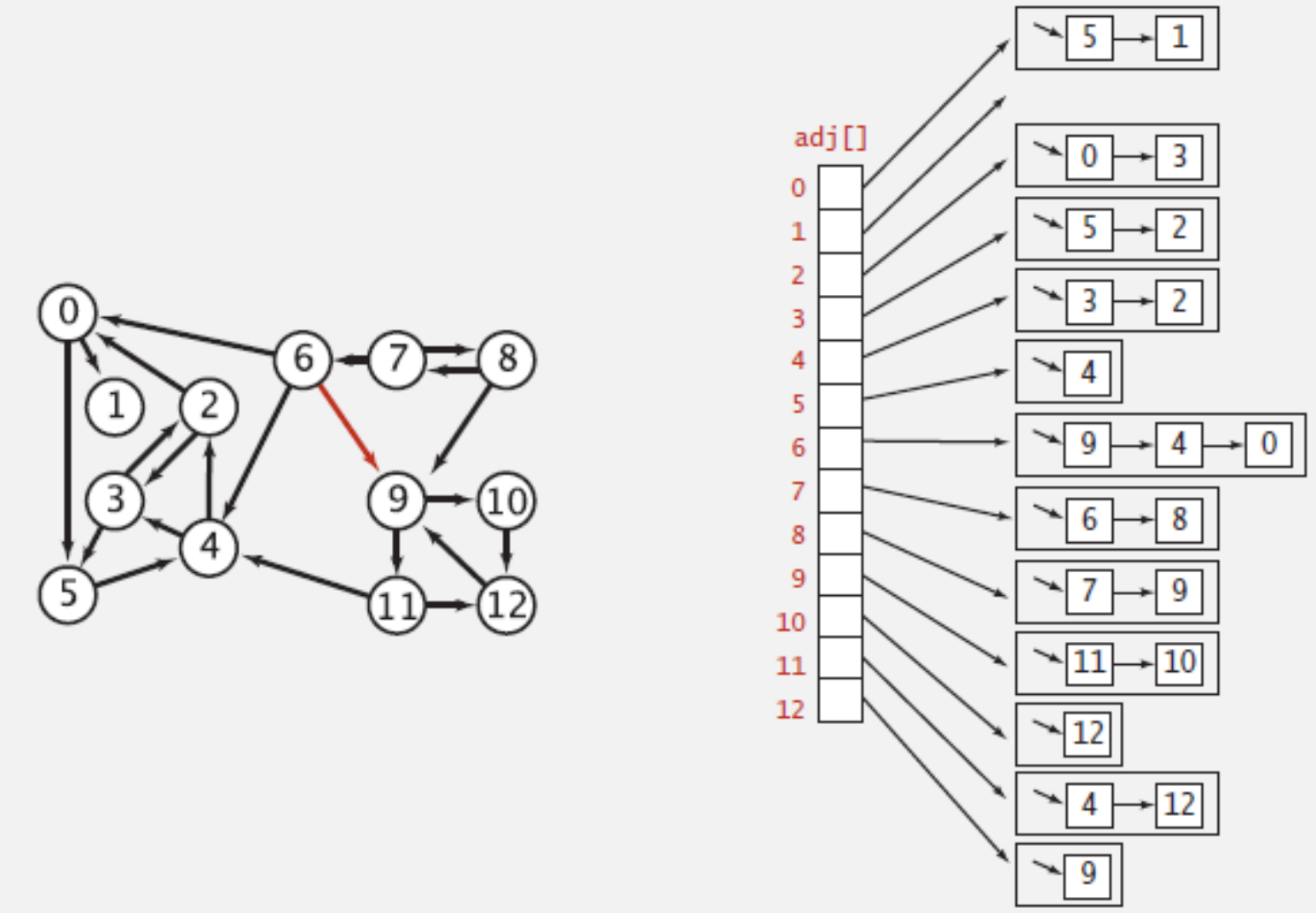
```
V → 13
E → 22
 4  2
 2  3
 3  2
 6  0
 0  1
 2  0
11 12
12  9
 9 10
 9 11
 8  9
10 12
11  4
 4  3
 3  5
 7  8
 8  7
 5  4
 0  5
 6  4
 6  9
 7  6
```




```
0->5
0->1
2->0
2->3
3->5
4->3
4->2
5->4
6->9
6->4
6->0
...
11->4
11->12
12-9
```

[Sedgewick/Wayne]

Representação – Listas de adjacências



[Sedgewick/Wayne]

Desempenho

- Na prática: usar a representação em listas de adjacências
- Os grafos orientados do mundo real são habitualmente esparsos !!
- Algoritmos iteram sobre os vértices adjacentes a um vértice dado

representation	space	insert edge from v to w	edge from v to w?	iterate over vertices adjacent from v?
list of edges	E	1	E	E
adjacency matrix	V^2	1 †	1	V
adjacency list	E + V	1	outdegree(v)	outdegree(v)

† disallows parallel edges

[Sedgewick/Wayne]

Sugestão de leitura

- R. Sedgewick and K. Wayne, "*Algorithms*", 4th. Ed., Addison-Wesley, 2011
  - Chapter 4