

# ISOMORFISMO, EMPARELHAMENTO E COMPONENTES CONEXOS

DCE529 - Algoritmos e Estruturas de Dados III

Atualizado em: 28 de maio de 2023

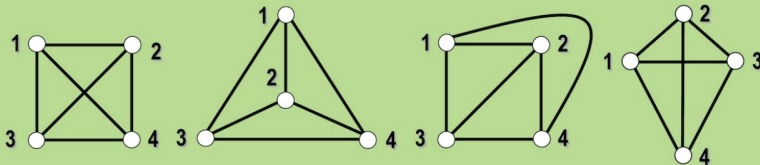
Iago Carvalho

Departamento de Ciência da Computação



## Grafos isomorfos

Dois grafos são isomorfos se e somente se existir uma correspondência um-para-um entre seus vértices e suas arestas de maneira que suas relações de adjacência sejam preservadas



# CONDIÇÕES NECESSÁRIAS PARA ISOMORFISMO

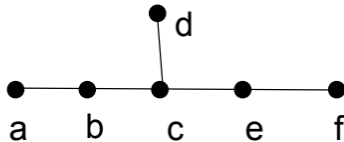
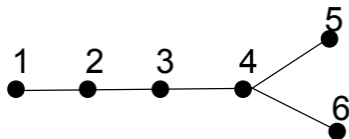
Existem 4 condições necessárias (mas não suficientes) para que dois grafos sejam isomorfos

1. Mesmo número de vértices
2. Mesmo número de arestas
3. Mesmo número de componentes
4. Mesmo número de vértices com mesmo grau

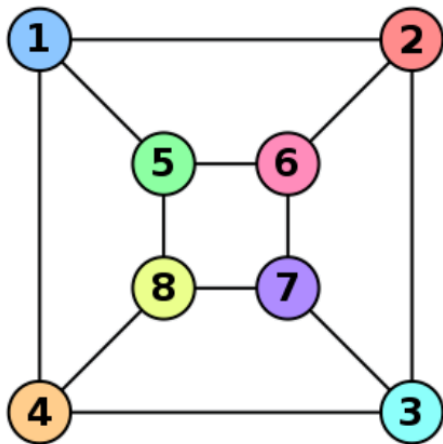
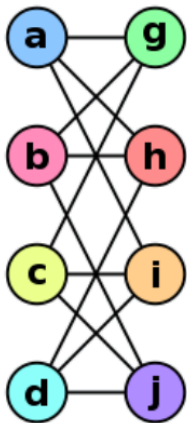
Estas quatro condições podem ser checadas em tempo polinomial

- Elas conseguem nos dizer se um grafo não é isomorfo
- Entretanto, reconhecer a isomorfia não é tão simples assim

# SÃO ISOMORFOS?



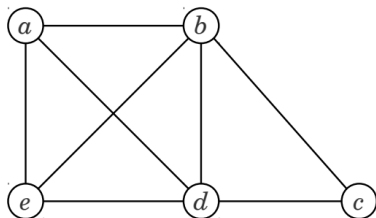
# SÃO ISOMORFOS?



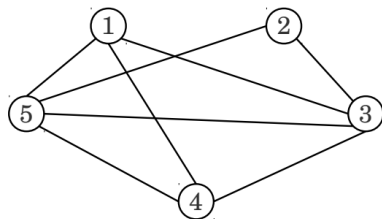
# ALGORITMO PARA RECONHECIMENTO DE ISOMORFISMO

Um algoritmo simples para reconhecimento de isomorfismo tem complexidade  $\mathcal{O}(n!)$

- Consiste em verificar todas as permutações de linhas e colunas das matrizes de adjacência de dois grafos



	a	b	c	d	e
a	0	1	0	1	1
b	1	0	1	1	1
c	0	1	0	1	0
d	1	1	1	0	1
e	1	1	0	1	0

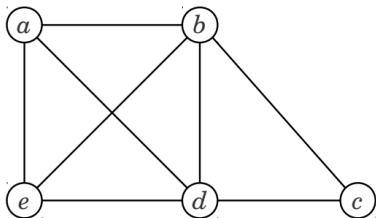


	1	2	3	4	5
1	0	0	1	1	1
2	0	0	1	0	1
3	1	1	0	1	1
4	1	0	1	0	1
5	1	1	1	1	0

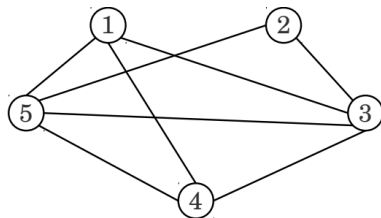
# ALGORITMO PARA RECONHECIMENTO DE ISOMORFISMO

Um algoritmo simples para reconhecimento de isomorfismo tem complexidade  $\mathcal{O}(n!)$

- Consiste em verificar todas as permutações de linhas e colunas das matrizes de adjacência de dois grafos



	$a$	$b$	$c$	$d$	$e$
$a$	0	1	0	1	1
$b$	1	0	1	1	1
$c$	0	1	0	1	0
$d$	1	1	1	0	1
$e$	1	1	0	1	0



	1	5	2	3	4
1	0	1	0	1	1
5	1	0	1	1	1
2	0	1	0	1	0
3	1	1	1	0	1
4	1	1	0	1	0

# ALGORITMO PARA RECONHECIMENTO DE ISOMORFISMO

Outro algoritmo para reconhecimento de isomorfismo [▶ Link](#)

- Complexidade quasipolinomial
  - Complexidade pior que polinomial
  - Mas menor que exponencial

Entretanto, a complexidade do problema ainda não é conhecida

- Provado pertencer a NP
- Entretanto, não se sabe se pertence a P ou é completo para NP

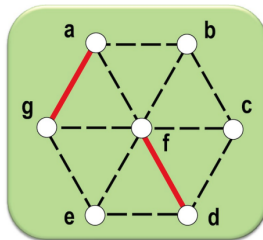
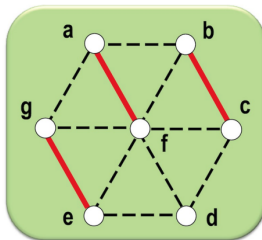
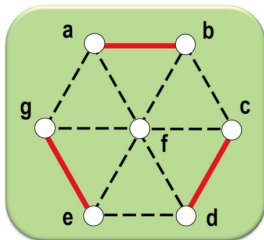


# EMPARELHAMENTO EM GRAFOS

# DEFINIÇÃO

Um emparelhamento (também conhecido como casamento ou *matching*) é um conjunto independente de arestas

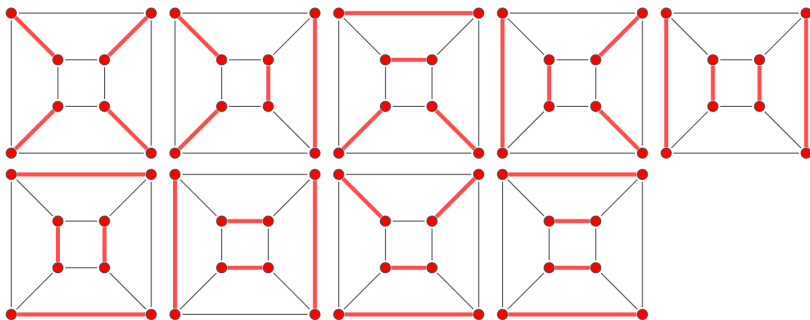
- Conjunto de arestas sem vértices em comum



# EMPARELHAMENTO PERFEITO

Um emparelhamento perfeito inclui todos os vértices do grafo

- Um emparelhamento perfeito é maximal
- Algoritmo de Blossoms -  $\mathcal{O}(|V|^2|E|)$



# EMPARELHAMENTO DE PESO MÁXIMO

Um emparelhamento que considera os pesos das arestas

- Somatório dos pesos das arestas utilizadas é maximal
- Algoritmo de caminhos, árvores e flores
  - Algoritmo de Edmond's
  - Polinomial, de complexidade  $\mathcal{O}(|V|^2|E|)$

Um caso especial (e muito útil) é definido em grafos bipartidos

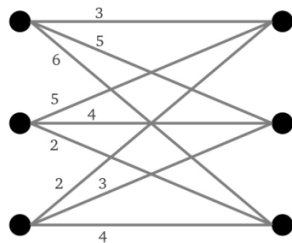
- Problema da atribuição
- Algoritmo Húngaro, de complexidade  $\mathcal{O}(|V|^3)$

# PROBLEMA DA ATRIBUIÇÃO

Conjunto de trabalhadores  $\times$  conjunto de tarefas

- Cada trabalhador gasta um tempo diferente para cada tarefa
- Como dividir as tarefas entre os trabalhadores?
- Desejável gastar o menor tempo o possível

Operário \ Máquina	1	2	3
1	3	5	6
2	5	4	2
3	2	3	4



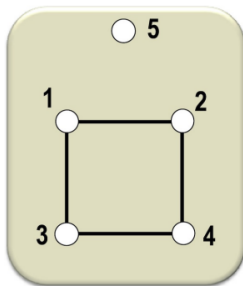
COMPONENTES CONEXOS

A conectividade tem a ver com a alcançabilidade (ou atingibilidade) em grafos

- Fecho transitivo dos vértices
- Um vértice  $u$  é alcançável a partir de  $v$  se existir um caminho entre  $u$  e  $v$  em  $G$
- É possível ir do ponto A até o ponto B?
- O computador X está conectado na mesma rede que o computador Y?
- Existe um barramento de dados entre dois chips de um computador?
- É possível enviar energia de uma usina hidroelétrica para uma cidade?

# GRAFOS DESCONEXOS

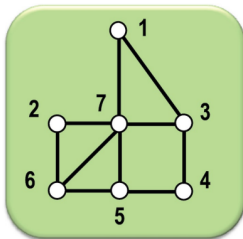
Um grafo é desconexo se existir pelo menos um par de vértices não alcançáveis entre si





# GRAFO NÃO-ORIENTADO CONEXO

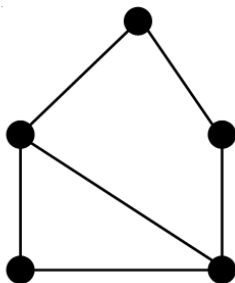
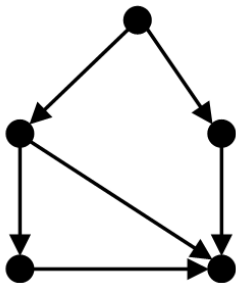
Um grafo não orientado é conexo se todos os vértices são alcançáveis a partir de um vértice qualquer



# CONECTIVIDADE EM GRAFOS ORIENTADOS

Em grafos orientados, nós temos diferentes conceitos de conectividade

- Ele é considerado conexo se seu grafo não-orientado subjacente é conexo



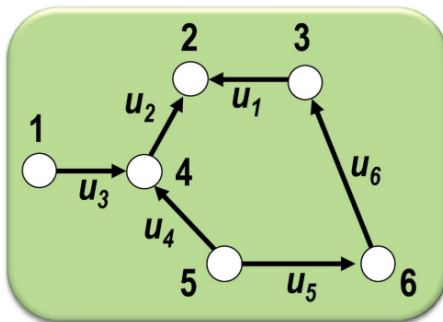
Além disso, dizemos que estes grafos podem ser divididos em três categorias

1. Simplemente conexo
2. Semi-fortemente conexo
3. Fortemente conexo

Cada categoria depende da alcançabilidade de seus vértices

# SIMPLESMENTE CONEXO

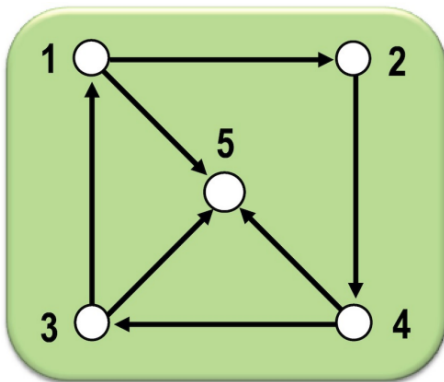
Todo grafo conexo é simplesmente conexo (**s-conexo**)



# SEMI-FORTEMENTE CONEXO

Um grafo é semi-fortemente conexo (**sf-conexo**) se

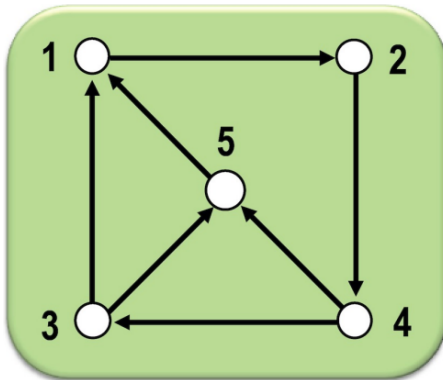
- Para todo par de vértices  $u, v \in V$
- Existe um caminho de  $u$  para  $v$  **OU** de  $v$  para  $u$



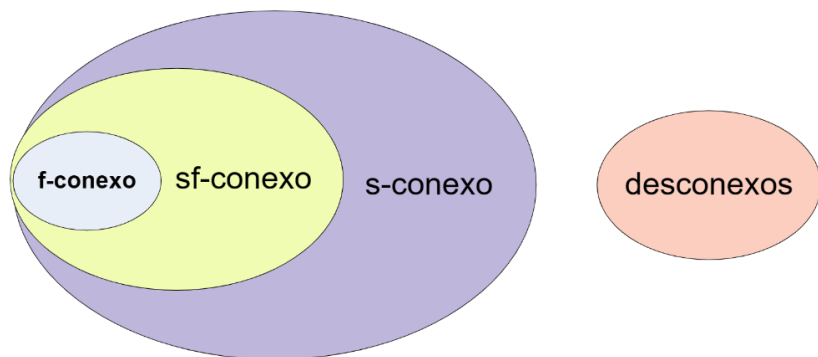
# FORTEMENTE CONEXO

Um grafo é fortemente conexo (**f-conexo**) se

- Para todo par de vértices  $u, v \in V$
- Existe um caminho de  $u$  para  $v$  E de  $v$  para  $u$



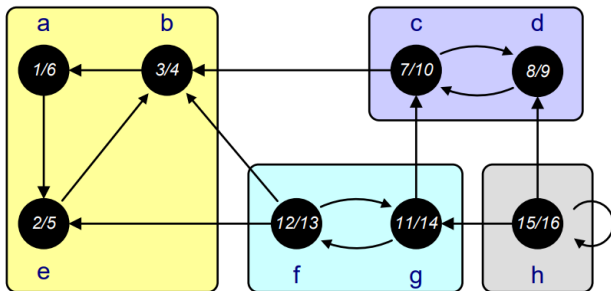
## CONECTIVIDADE EM GRAFOS DIRECIONADOS



# COMPONENTES CONEXOS

Um componente conexo é um subgrafo conexo maximal

- Maior subgrafo induzido conexo
- Grafos conexos possuem somente um componente
- O número de componentes de um grafo é denotado por  $c$

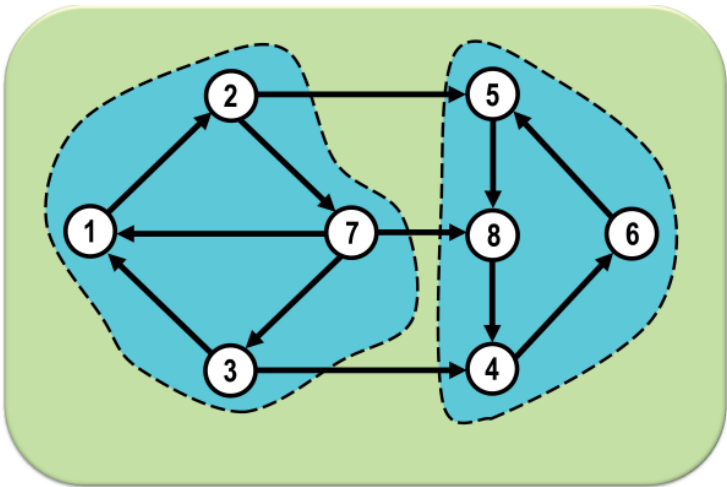


Grafo com  $c = 4$



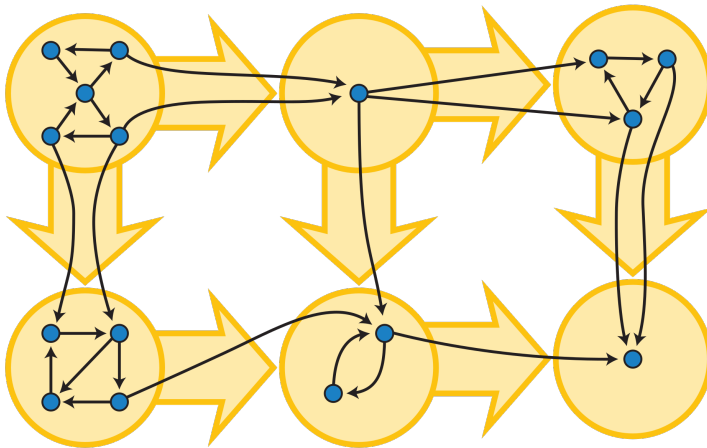
# COMPONENTES FORTEMENTE CONEXOS

Em um grafo direcionado, componentes fortemente conexos são subgrafos maximais **f-conexos**



# GRAFOS REDUZIDOS

Um grafo reduzido é construído mostrando somente as conexões entre os seus componentes conexos

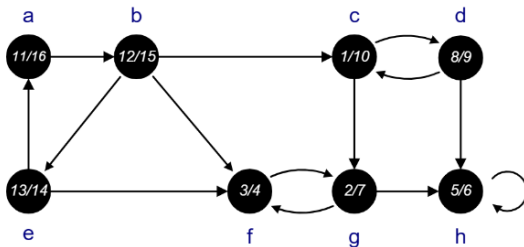
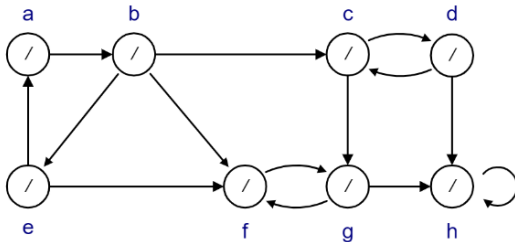


# COMO ENCONTRAR COMPONENTES FORTEMENTE CONEXOS

Algoritmo baseado no algoritmo de busca em profundidade

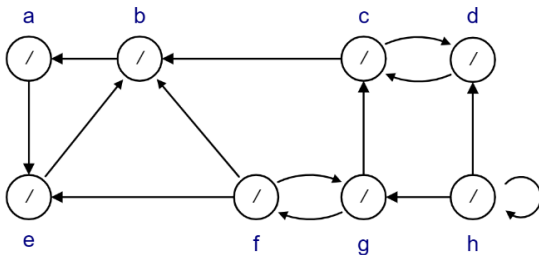
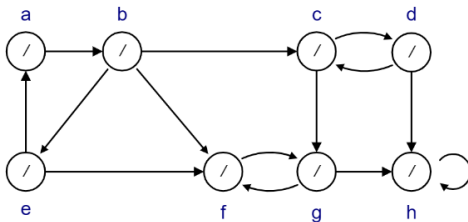
- Complexidade de  $\Theta(|V| + |A|)$
- 1. Faça uma busca em profundidade a partir de  $u$  e salve os tempos de fechamento de cada vértice
  - O tempo que ele é colorido de preto
- 2. Gere o grafo transposto  $G^T$  do grafo original  $G$ 
  - Arcos com sentido invertido
- 3. Faça uma busca em profundidade em  $G^T$ 
  - Considerando a ordem decrescente do tempo de fechamento dos vértices encontrado na etapa 1

# COMO ENCONTRAR COMPONENTES FORTEMENTE CONEXOS



$[a, b, e, c, d, g, h, f]$

# COMO ENCONTRAR COMPONENTES FORTEMENTE CONEXOS



# COMO ENCONTRAR COMPONENTES FORTEMENTE CONEXOS

