

Algoritmos e Estruturas de Dados II

Grafos

Prof. Tiago Eugenio de Melo

tmelo@uea.edu.br

www.tiagodemelo.info

Observações

- As palavras com a fonte `Courier` indicam as palavras-reservadas da linguagem de programação.

Referências

- **Projetos de Algoritmos – com implementações em Pascal e C.** Nivio Ziviani. 2ª edição. Thomson, 2005.
- Material baseado nas notas de aula do professor Marco Antônio Moreira de Carvalho. Acessado em 01/10/2019:
<http://www.decom.ufop.br/marco/ensino/bcc204/material-das-aulas>

REVISÃO

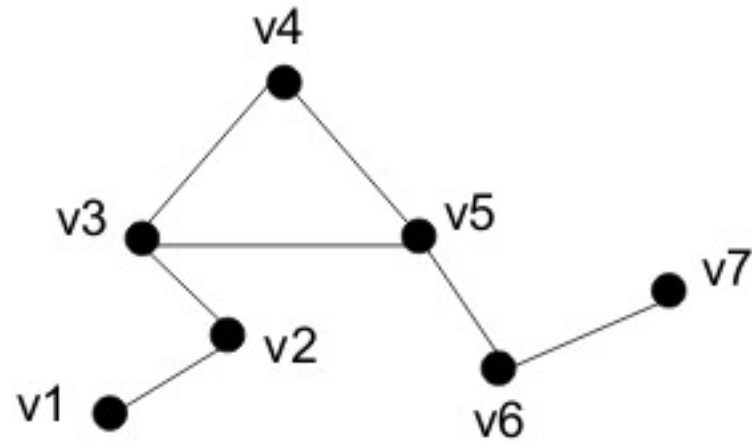
Revisão

Revisão

- Com relação ao grafo abaixo, responda:

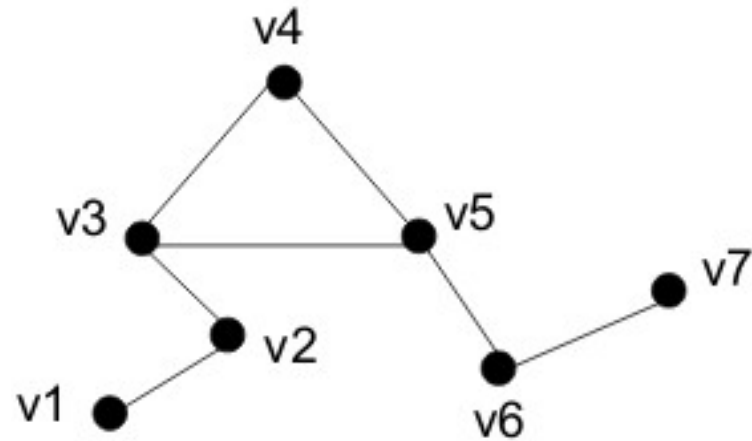
Revisão

- Com relação ao grafo abaixo, responda:



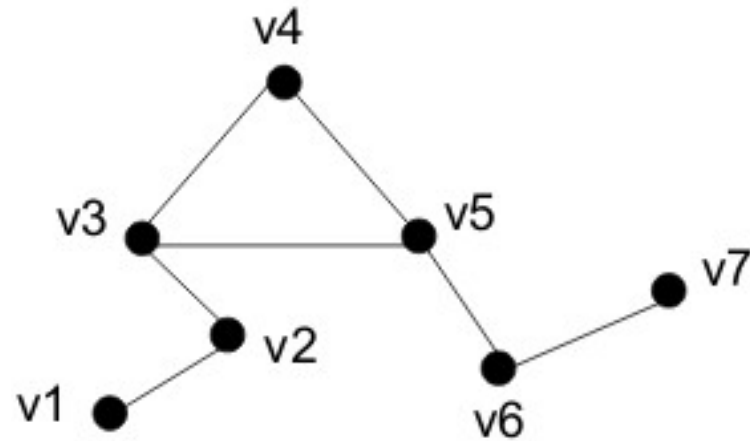
Revisão

- Com relação ao grafo abaixo, responda:
 - O grafo é simples?



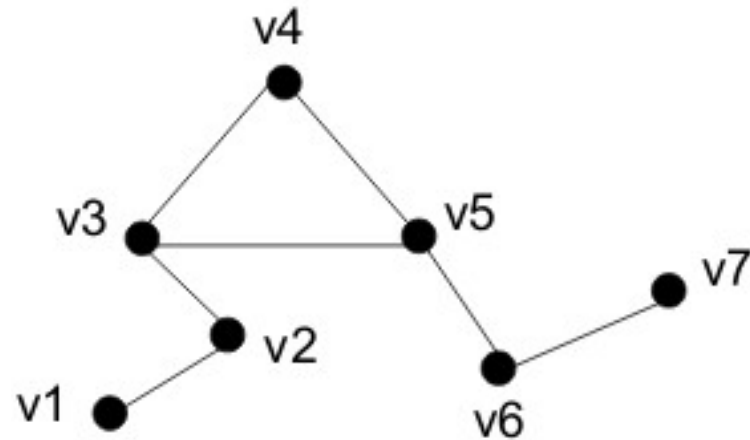
Revisão

- Com relação ao grafo abaixo, responda:
 - O grafo é simples?
 - Completo?



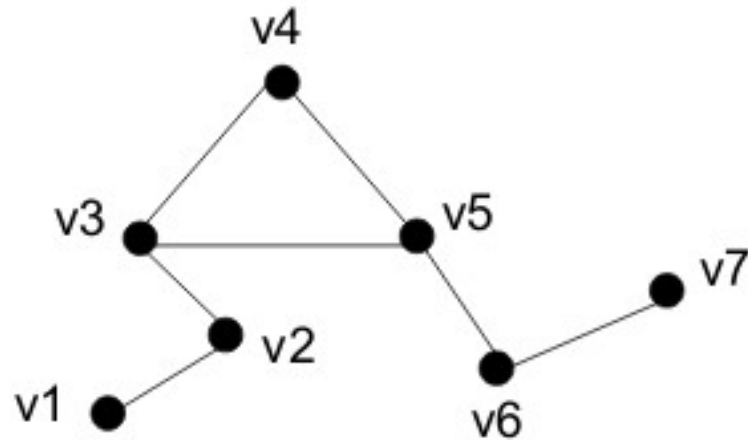
Revisão

- Com relação ao grafo abaixo, responda:
 - O grafo é simples?
 - Completo?
 - Regular?



Revisão

- Com relação ao grafo abaixo, responda:
 - O grafo é simples?
 - Completo?
 - Regular?
 - Conexo?



Revisão

- Com relação ao grafo abaixo, responda:

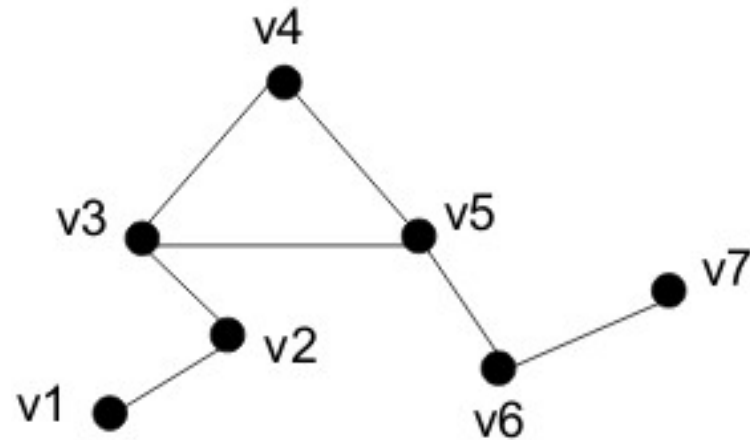
- O grafo é simples?

- Completo?

- Regular?

- Conexo?

- Encontre dois caminhos diferentes entre v_3 e v_6 .



Revisão

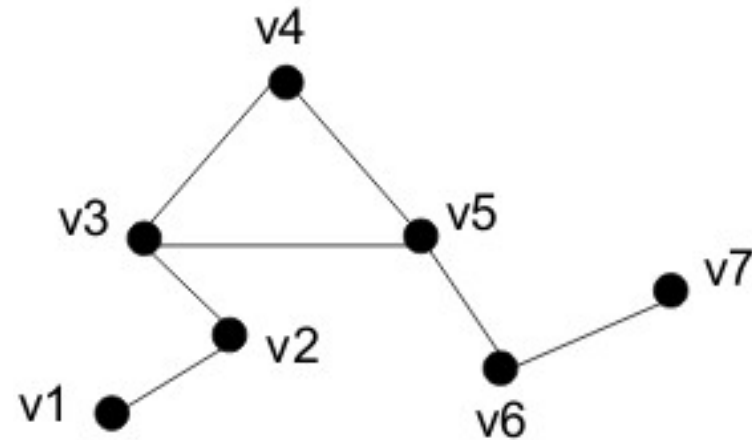
- Com relação ao grafo abaixo, responda:

- O grafo é simples?

- Completo?

- Regular?

- Conexo?

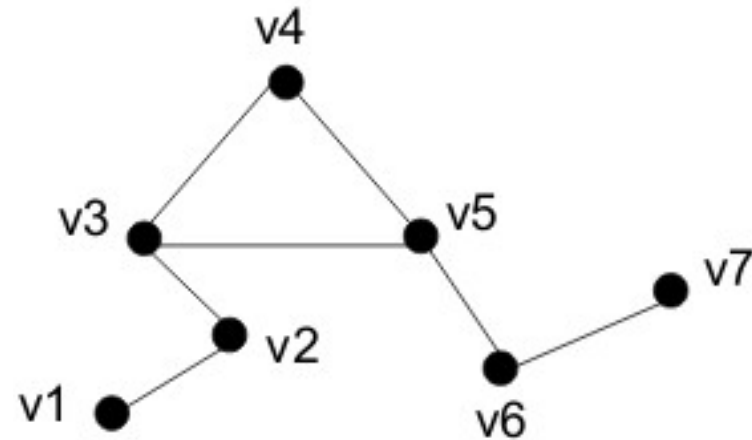


- Encontre dois caminhos diferentes entre v_3 e v_6 .

- Indique uma aresta cuja remoção torna o grafo desconexo.

Revisão

- Com relação ao grafo abaixo, responda:



- O grafo é simples?
- Completo?
- Regular?
- Conexo?
- Encontre dois caminhos diferentes entre v_3 e v_6 .
- Indique uma aresta cuja remoção torna o grafo desconexo.
- Indique a representação deste grafo por listas de adjacências.

ISOMORFISMO

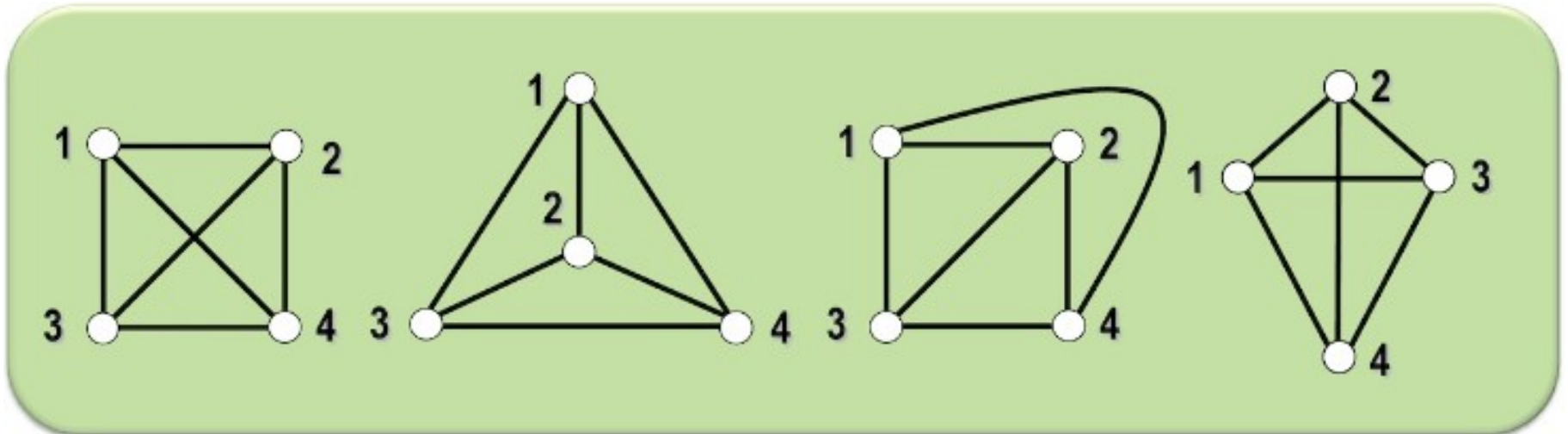
Isomorfismo

Isomorfismo

- Dois grafos G e H são considerados **isomorfos** se existir uma correspondência um-para-um entre seus vértices e entre suas arestas, de maneira que as relações de incidência sejam preservadas.

Isomorfismo

- Dois grafos G e H são considerados **isomorfos** se existir uma correspondência um-para-um entre seus vértices e entre suas arestas, de maneira que as relações de incidência sejam preservadas.



Isomorfismo

Isomorfismo

- Condições necessárias, mas não suficientes para o isomorfismo:

Isomorfismo

- Condições necessárias, mas não suficientes para o isomorfismo:
 - Mesmo número de vértices.

Isomorfismo

- Condições necessárias, mas não suficientes para o isomorfismo:
 - Mesmo número de vértices.
 - Mesmo número de arestas.

Isomorfismo

- Condições necessárias, mas não suficientes para o isomorfismo:
 - Mesmo número de vértices.
 - Mesmo número de arestas.
 - Mesmo número de componentes.

Isomorfismo

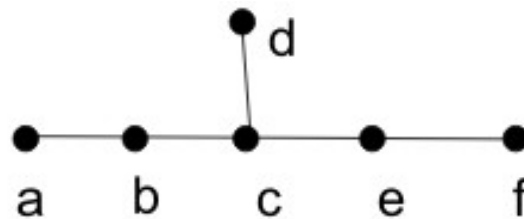
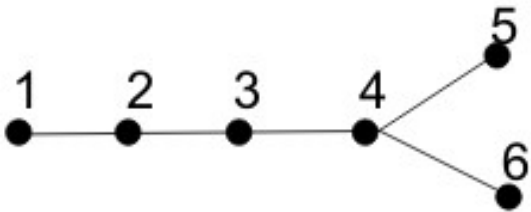
- Condições necessárias, mas não suficientes para o isomorfismo:
 - Mesmo número de vértices.
 - Mesmo número de arestas.
 - Mesmo número de componentes.
 - Mesmo número de vértices com o mesmo grau.

Isomorfismo

- Condições necessárias, mas não suficientes para o isomorfismo:
 - Mesmo número de vértices.
 - Mesmo número de arestas.
 - Mesmo número de componentes.
 - Mesmo número de vértices com o mesmo grau.
- Exemplos:

Isomorfismo

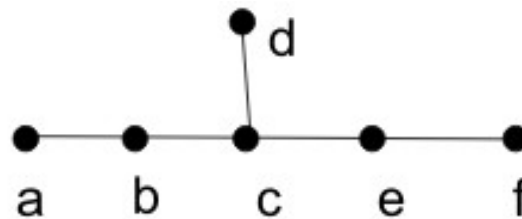
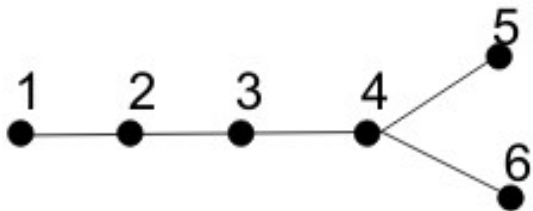
- Condições necessárias, mas não suficientes para o isomorfismo:
 - Mesmo número de vértices.
 - Mesmo número de arestas.
 - Mesmo número de componentes.
 - Mesmo número de vértices com o mesmo grau.
- Exemplos:



Isomorfismo

- Condições necessárias, mas não suficientes para o isomorfismo:
 - Mesmo número de vértices.
 - Mesmo número de arestas.
 - Mesmo número de componentes.
 - Mesmo número de vértices com o mesmo grau.

- Exemplos:



Não existem algoritmos comprovadamente eficientes para determinar se dois grafos são isomorfos.



Isomorfismo

1. Mais lento que polinomial, mas significativamente mais rápido que exponencial.
2. <https://jeremykun.com/2015/11/12/a-quasipolynomial-time-algorithm-for-graph-isomorphism-the-details>
3. Mais lento que o quasipolinomial, mas mais rápido que o exponencial.
4. <http://people.cs.uchicago.edu/~laci/update.html>

Isomorfismo

- Complexidade

1. Mais lento que polinomial, mas significativamente mais rápido que exponencial.
2. <https://jeremykun.com/2015/11/12/a-quasipolynomial-time-algorithm-for-graph-isomorphism-the-details>
3. Mais lento que o quasipolinomial, mas mais rápido que o exponencial.
4. <http://people.cs.uchicago.edu/~laci/update.html>

Isomorfismo

- Complexidade
 - O algoritmo intuitivo para testes de isomorfismo consiste em analisar as permutações de linhas e colunas de matrizes de equivalência, em busca de uma relação um-para-um, ou seja, $O(n!)$.

1. Mais lento que polinomial, mas significativamente mais rápido que exponencial.

2. <https://jeremykun.com/2015/11/12/a-quasipolynomial-time-algorithm-for-graph-isomorphism-the-details>

3. Mais lento que o quasipolinomial, mas mais rápido que o exponencial.

4. <http://people.cs.uchicago.edu/~laci/update.html>

Isomorfismo

- Complexidade

- O algoritmo intuitivo para testes de isomorfismo consiste em analisar as permutações de linhas e colunas de matrizes de equivalência, em busca de uma relação um-para-um, ou seja, $O(n!)$.
- Em outubro de 2015, László Babai, da Universidade de Chicago, anunciou um algoritmo quasipolinomial¹ para o teste do isomorfismo!²

1. Mais lento que polinomial, mas significativamente mais rápido que exponencial.

2. <https://jeremykun.com/2015/11/12/a-quasipolynomial-time-algorithm-for-graph-isomorphism-the-details>

3. Mais lento que o quasipolinomial, mas mais rápido que o exponencial.

4. <http://people.cs.uchicago.edu/~laci/update.html>

Isomorfismo

- Complexidade

- O algoritmo intuitivo para testes de isomorfismo consiste em analisar as permutações de linhas e colunas de matrizes de equivalência, em busca de uma relação um-para-um, ou seja, $O(n!)$.
- Em outubro de 2015, László Babai, da Universidade de Chicago, anunciou um algoritmo quasipolinomial¹ para o teste do isomorfismo!²
- Em 4 de outubro de 2017 foi descoberto um erro na prova, reclassificando o algoritmo como subexponencial³.

1. Mais lento que polinomial, mas significativamente mais rápido que exponencial.

2. <https://jeremykun.com/2015/11/12/a-quasipolynomial-time-algorithm-for-graph-isomorphism-the-details>

3. Mais lento que o quasipolinomial, mas mais rápido que o exponencial.

4. <http://people.cs.uchicago.edu/~laci/update.html>

Isomorfismo

- Complexidade
 - O algoritmo intuitivo para testes de isomorfismo consiste em analisar as permutações de linhas e colunas de matrizes de equivalência, em busca de uma relação um-para-um, ou seja, $O(n!)$.
 - Em outubro de 2015, László Babai, da Universidade de Chicago, anunciou um algoritmo quasipolinomial¹ para o teste do isomorfismo!²
 - Em 4 de outubro de 2017 foi descoberto um erro na prova, reclassificando o algoritmo como subexponencial³.
 - Em 9 de janeiro de 2018, o erro da prova foi anunciado como contornado⁴.

1. Mais lento que polinomial, mas significativamente mais rápido que exponencial.

2. <https://jeremykun.com/2015/11/12/a-quasipolynomial-time-algorithm-for-graph-isomorphism-the-details>

3. Mais lento que o quasipolinomial, mas mais rápido que o exponencial.

4. <http://people.cs.uchicago.edu/~laci/update.html>

Isomorfismo

Isomorfismo

- Algoritmo Básico:

Isomorfismo

- Algoritmo Básico:
 - Mesmo número de vértices.

Isomorfismo

- Algoritmo Básico:
 - Mesmo número de vértices.
 - Mesmo número de arestas.

Isomorfismo

- Algoritmo Básico:
 - Mesmo número de vértices.
 - Mesmo número de arestas.
 - Mesmo número de componentes.

Isomorfismo

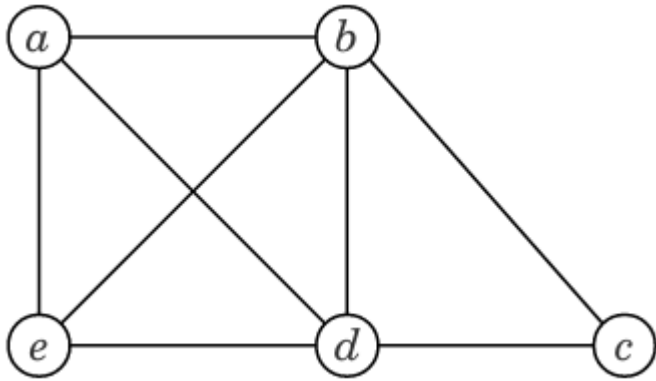
- Algoritmo Básico:
 - Mesmo número de vértices.
 - Mesmo número de arestas.
 - Mesmo número de componentes.
 - Mesmo número de vértices com o mesmo grau.

Isomorfismo

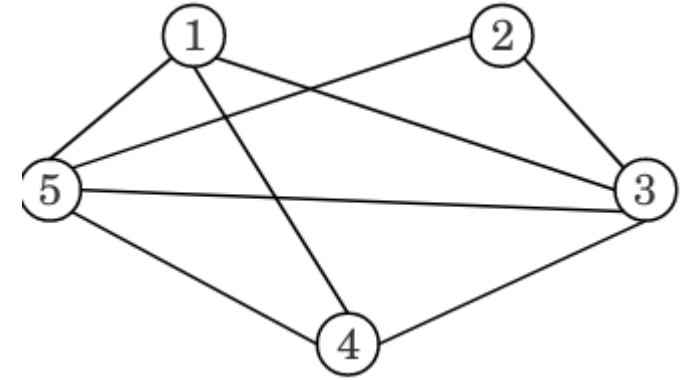
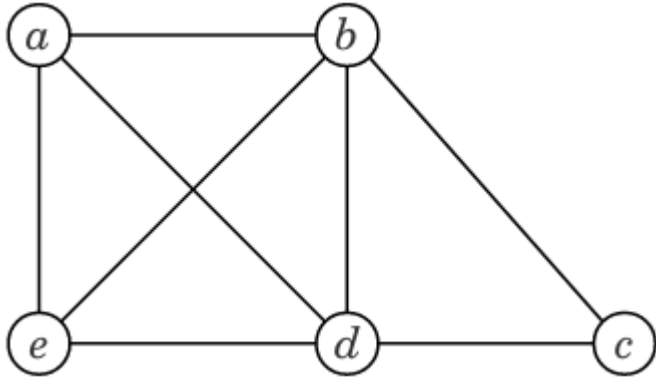
- Algoritmo Básico:
 - Mesmo número de vértices.
 - Mesmo número de arestas.
 - Mesmo número de componentes.
 - Mesmo número de vértices com o mesmo grau.
- Em seguida, efetuar a combinação das matrizes de adjacência dos grafos, verificando se são semelhantes.

Isomorfismo e Matrizes de Adjacência

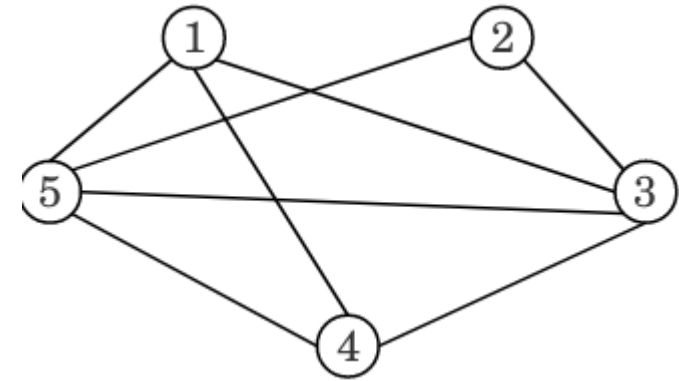
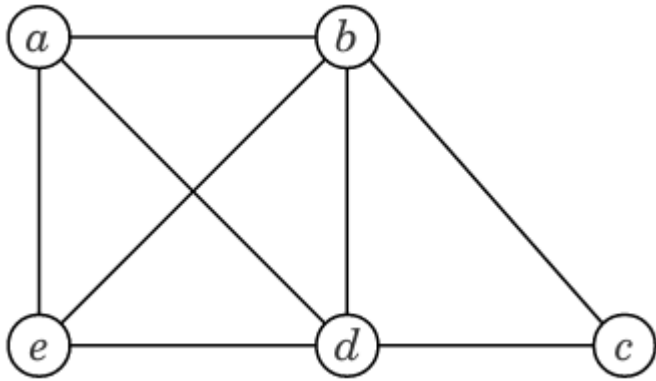
Isomorfismo e Matrizes de Adjacência



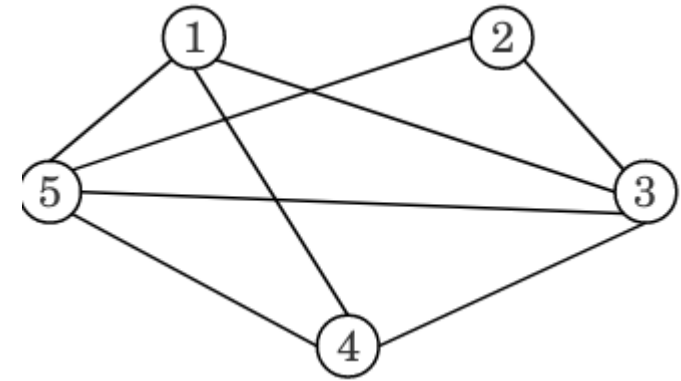
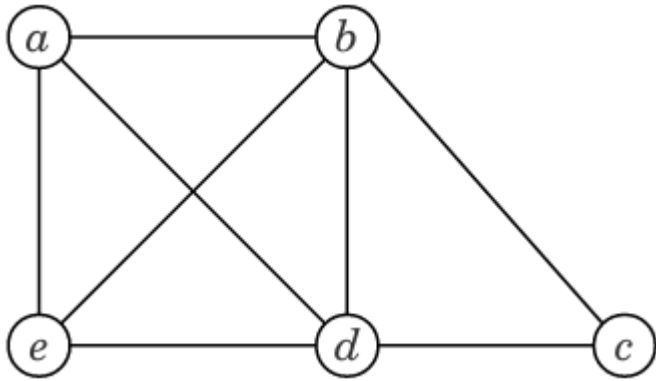
Isomorfismo e Matrizes de Adjacência



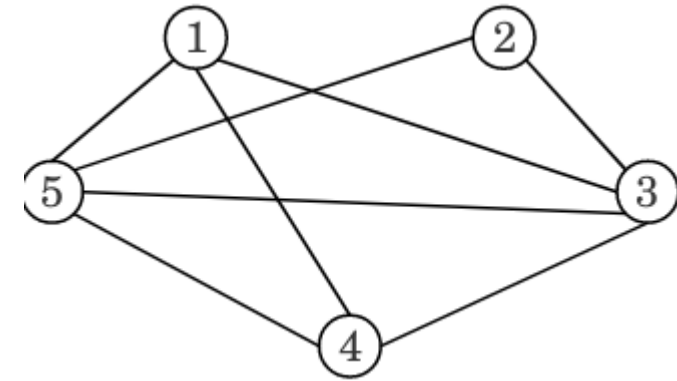
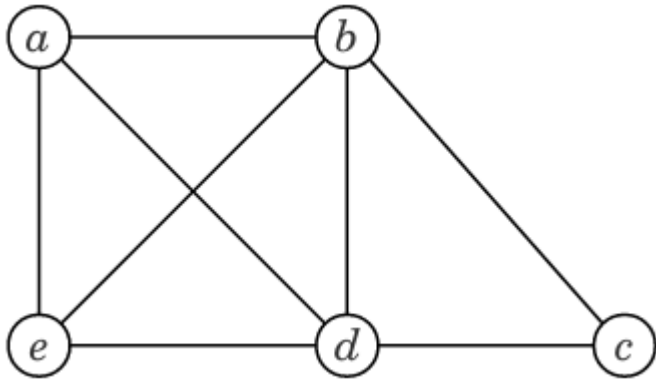
Isomorfismo e Matrizes de Adjacência



Isomorfismo e Matrizes de Adjacência



Isomorfismo e Matrizes de Adjacência



Matrizes de Adjacência

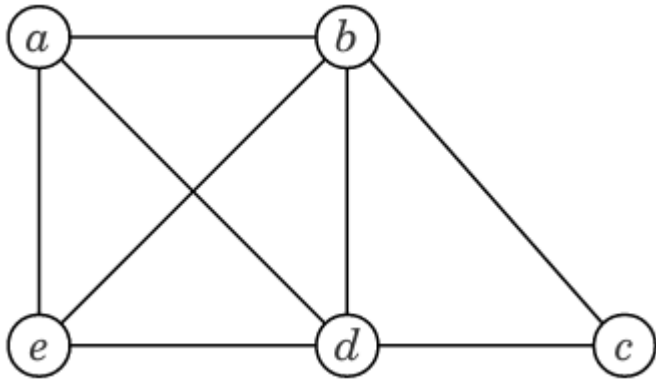


	a	b	c	d	e
a	0	1	0	1	1
b	1	0	1	1	1
c	0	1	0	1	0
d	1	1	1	0	1
e	1	1	0	1	0

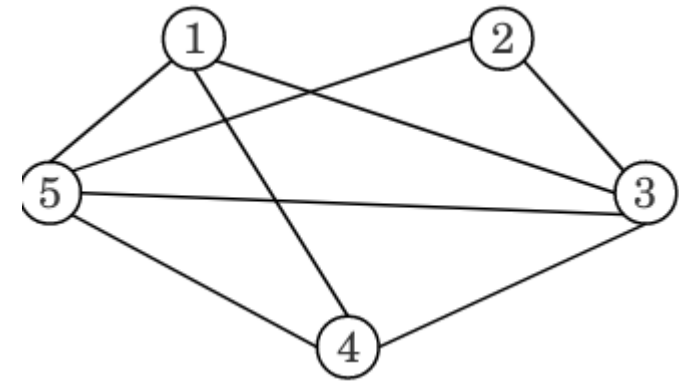
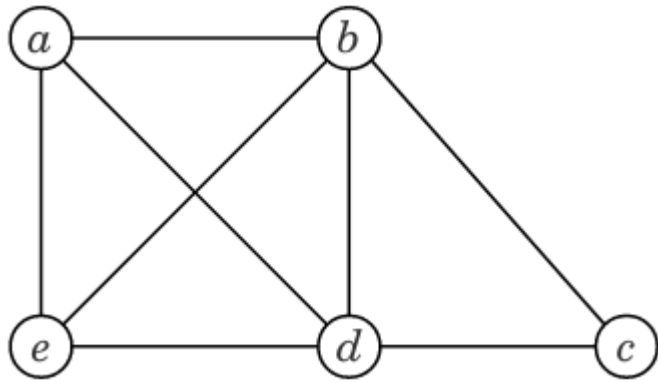
	1	2	3	4	5
1	0	0	1	1	1
2	0	0	1	0	1
3	1	1	0	1	1
4	1	0	1	0	1
5	1	1	1	1	0

Isomorfismo e Matrizes de Adjacência

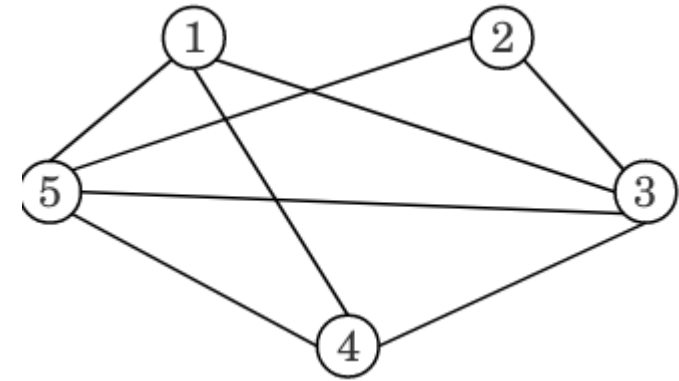
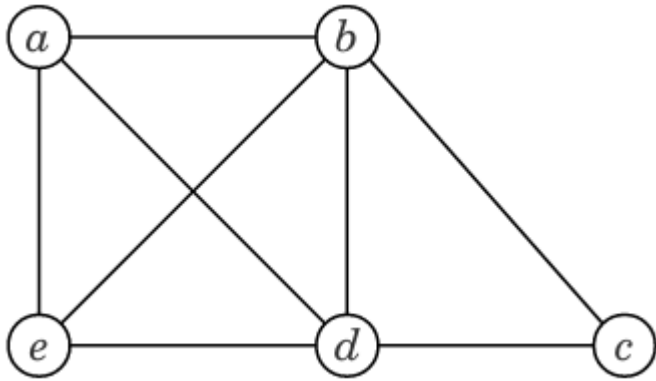
Isomorfismo e Matrizes de Adjacência



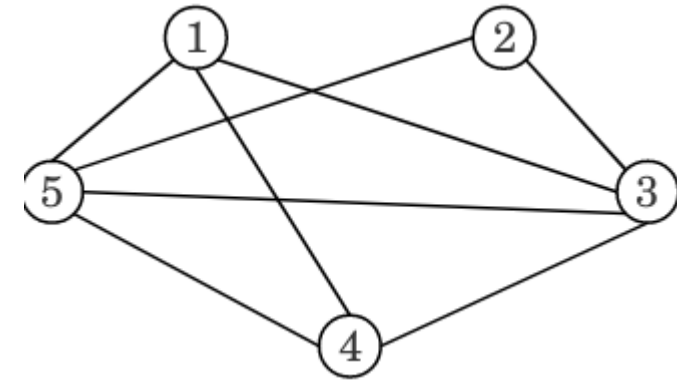
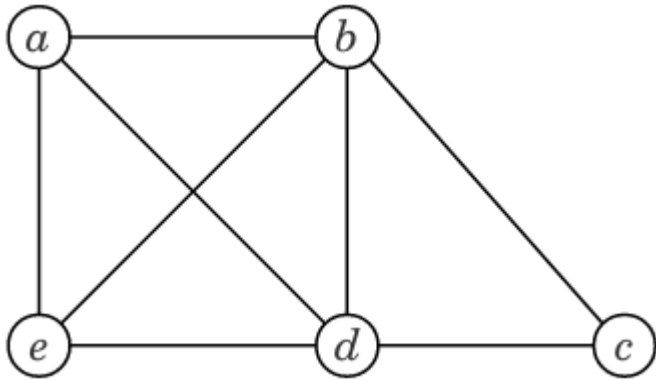
Isomorfismo e Matrizes de Adjacência



Isomorfismo e Matrizes de Adjacência



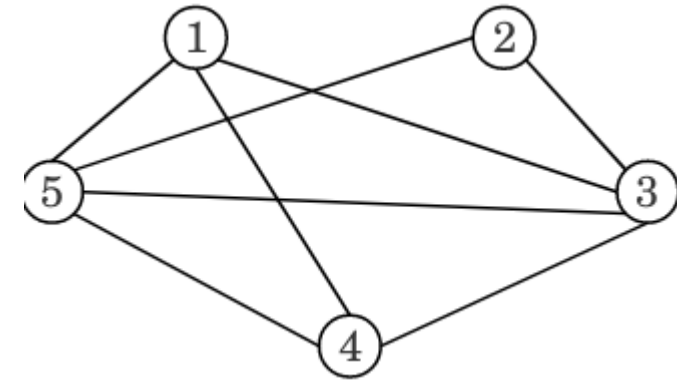
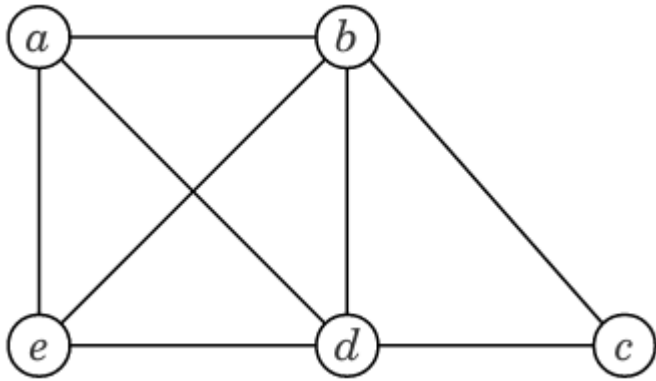
Isomorfismo e Matrizes de Adjacência



Matrizes de Adjacência



Isomorfismo e Matrizes de Adjacência



Matrizes de Adjacência



	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
<i>a</i>	0	1	0	1	1
<i>b</i>	1	0	1	1	1
<i>c</i>	0	1	0	1	0
<i>d</i>	1	1	1	0	1
<i>e</i>	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

SUBGRAFOS

Subgrafo

Subgrafo

- Um grafo $G_s = (V_s, A_s)$ é considerado um **subgrafo** de um grafo $G = (V, A)$ se todos os vértices e todas as arestas de G_s estão em G , ou seja, se $V_s \subseteq V$ e $A_s \subseteq A$.

Subgrafo

- Um grafo $G_s = (V_s, A_s)$ é considerado um **subgrafo** de um grafo $G = (V, A)$ se todos os vértices e todas as arestas de G_s estão em G , ou seja, se $V_s \subseteq V$ e $A_s \subseteq A$.
- Observações:

Subgrafo

- Um grafo $G_s = (V_s, A_s)$ é considerado um **subgrafo** de um grafo $G = (V, A)$ se todos os vértices e todas as arestas de G_s estão em G , ou seja, se $V_s \subseteq V$ e $A_s \subseteq A$.
- Observações:
 - Todo grafo é subgrafo de si próprio.

Subgrafo

- Um grafo $G_s = (V_s, A_s)$ é considerado um **subgrafo** de um grafo $G = (V, A)$ se todos os vértices e todas as arestas de G_s estão em G , ou seja, se $V_s \subseteq V$ e $A_s \subseteq A$.
- Observações:
 - Todo grafo é subgrafo de si próprio.
 - O subgrafo G_{s_2} de um subgrafo G_s de G também é subgrafo de G .

Subgrafo

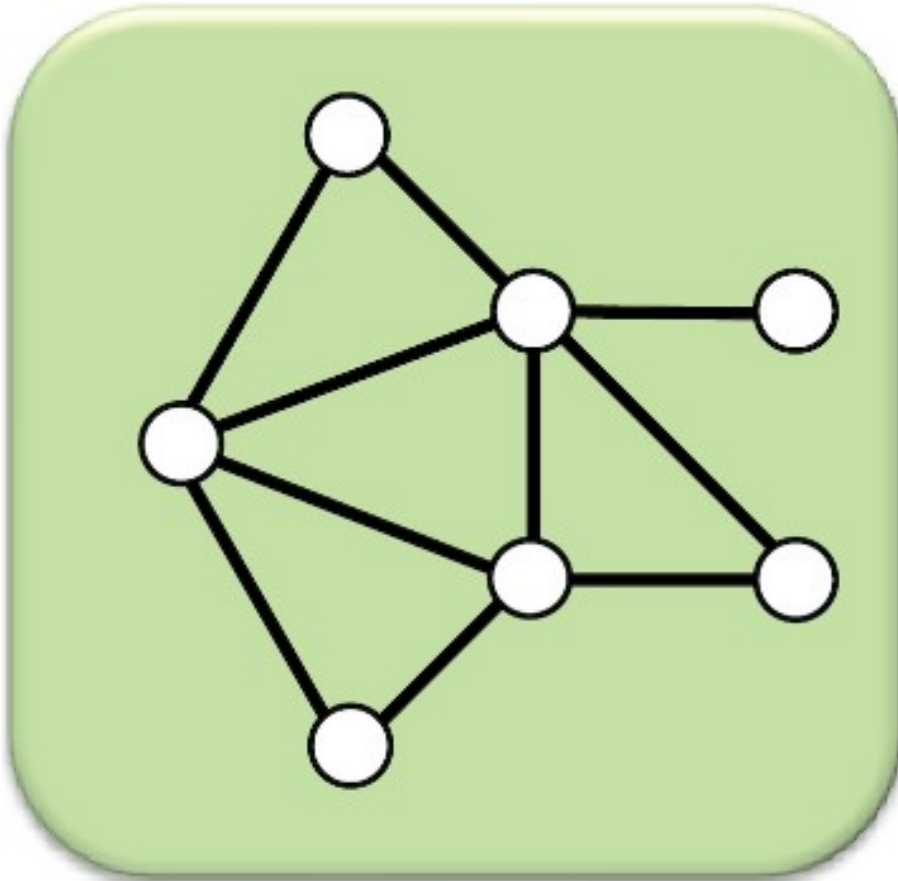
- Um grafo $G_s = (V_s, A_s)$ é considerado um **subgrafo** de um grafo $G = (V, A)$ se todos os vértices e todas as arestas de G_s estão em G , ou seja, se $V_s \subseteq V$ e $A_s \subseteq A$.
- Observações:
 - Todo grafo é subgrafo de si próprio.
 - O subgrafo G_{s_2} de um subgrafo G_s de G também é subgrafo de G .
 - Um vértice simples de G é um subgrafo de G .

Subgrafo

- Um grafo $G_s = (V_s, A_s)$ é considerado um **subgrafo** de um grafo $G = (V, A)$ se todos os vértices e todas as arestas de G_s estão em G , ou seja, se $V_s \subseteq V$ e $A_s \subseteq A$.
- Observações:
 - Todo grafo é subgrafo de si próprio.
 - O subgrafo G_{s2} de um subgrafo G_s de G também é subgrafo de G .
 - Um vértice simples de G é um subgrafo de G .
 - Uma aresta simples de G (juntamente com as suas extremidades) é um subgrafo de G .

Subgrafo

- Encontre todos os subgrafos:



CONNECTIVIDADE E CAMINHOS

Definições

Definições

- Passeio

Definições

- Passeio
 - Um passeio é uma sequência finita de vértices e arestas.

Definições

- Passeio
 - Um passeio é uma sequência finita de vértices e arestas.
 - Cada vértice da sequência é incidente a aresta que o precede e a aresta seguinte.

Definições

- Passeio
 - Um passeio é uma sequência finita de vértices e arestas.
 - Cada vértice da sequência é incidente a aresta que o precede e a aresta seguinte.
 - Essa sequência deve acabar e iniciar em um vértice (não necessariamente o mesmo).

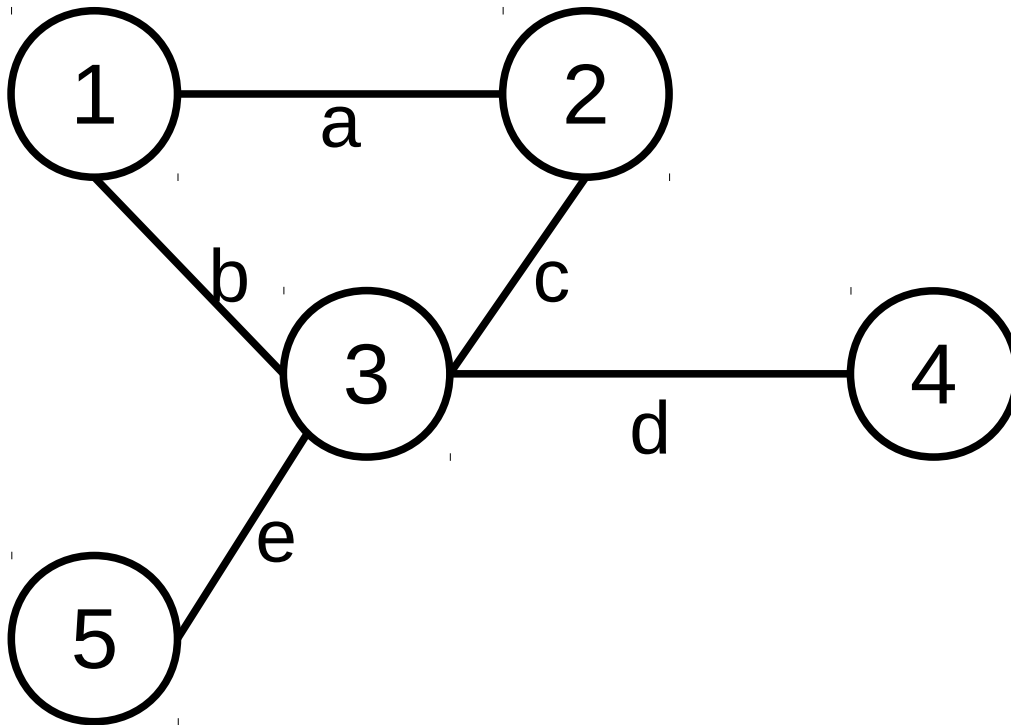
Passeio

Passeio

- Exemplo:

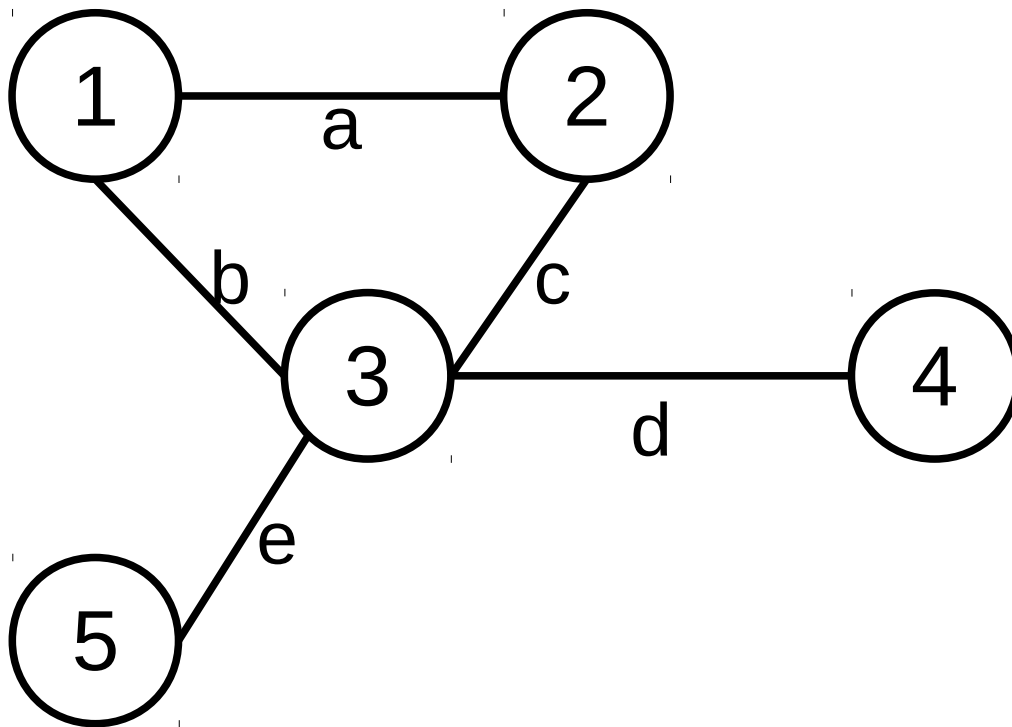
Passeio

- Exemplo:



Passeio

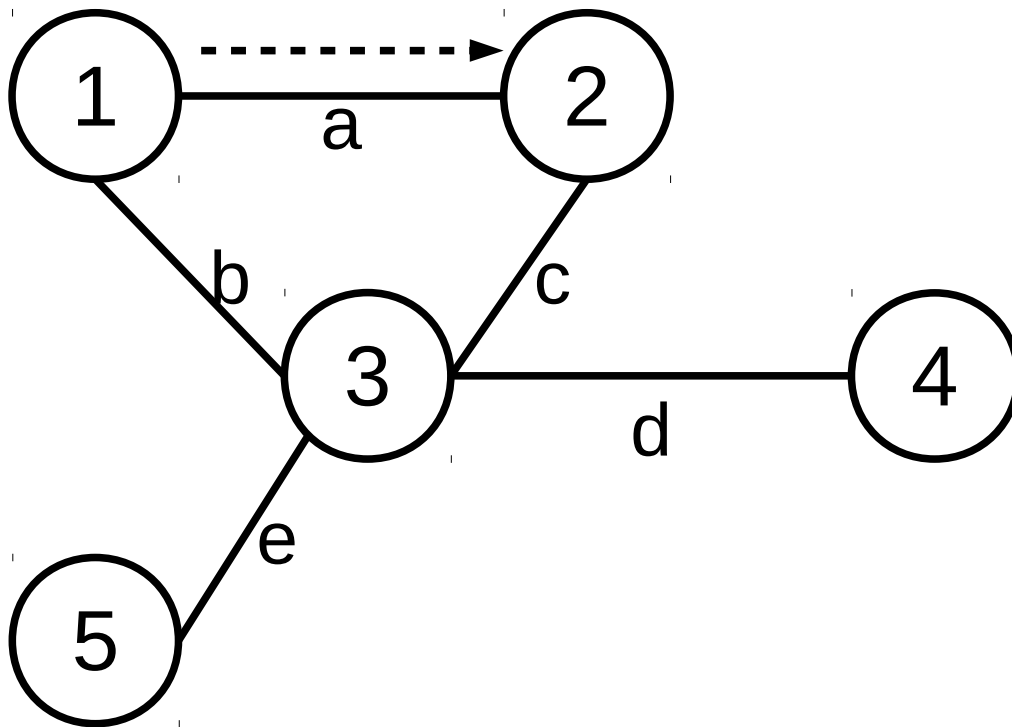
- Exemplo:



1 – a – 2

Passeio

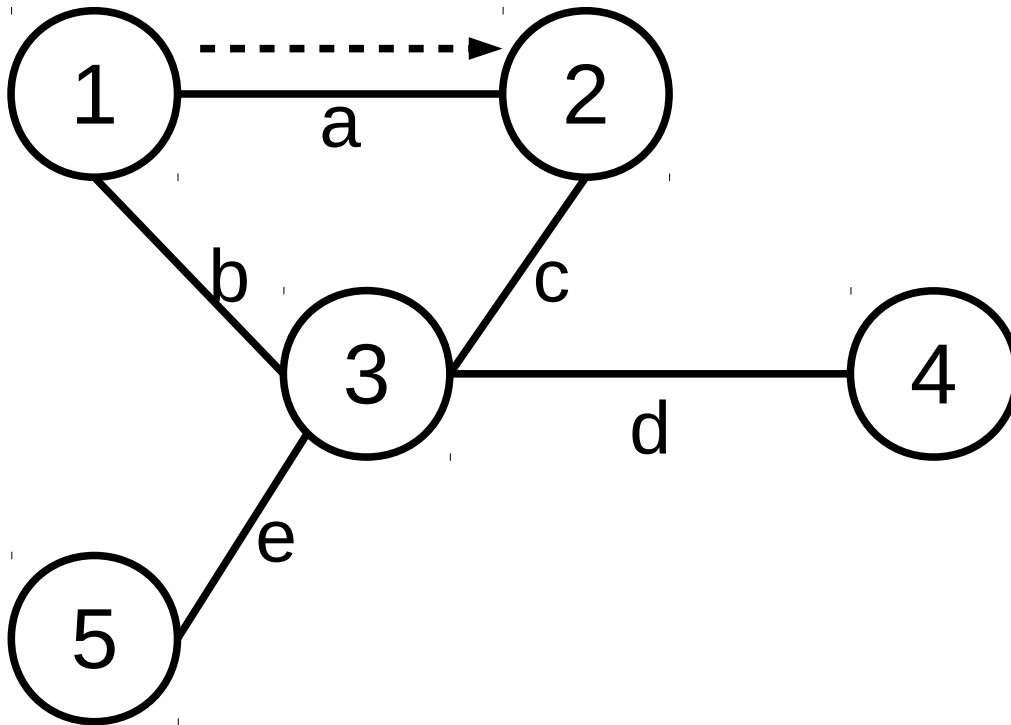
- Exemplo:



1 – a – 2

Passeio

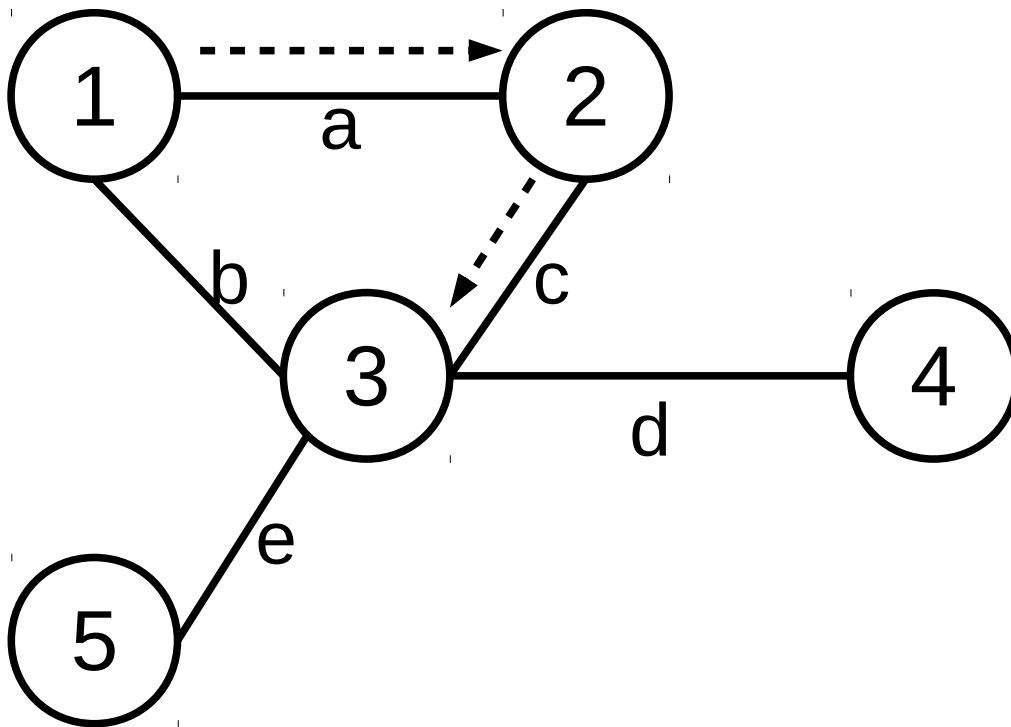
- Exemplo:



1 – a – 2
2 – c – 3

Passeio

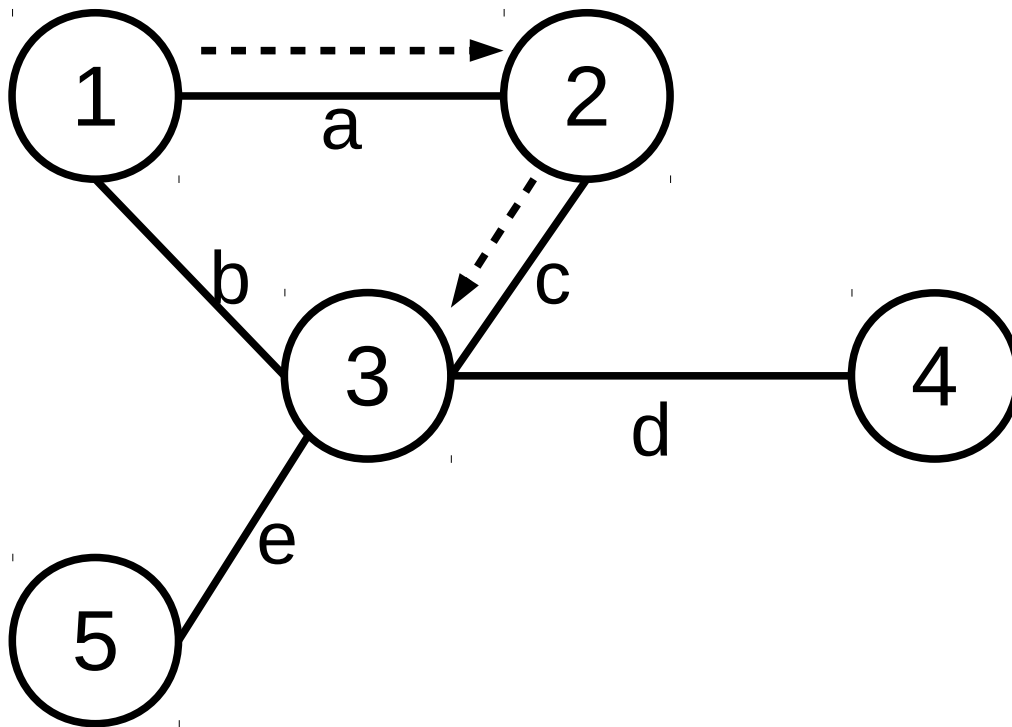
- Exemplo:



1 – a – 2
2 – c – 3

Passeio

- Exemplo:



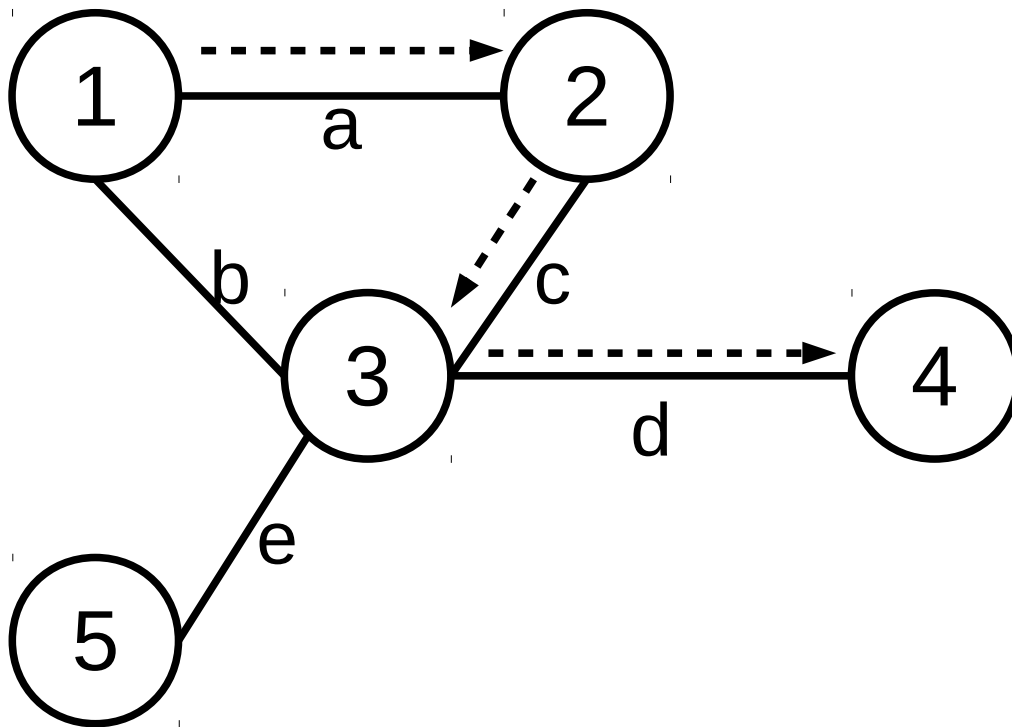
1 – a – 2

2 – c – 3

3 – d – 4

Passeio

- Exemplo:



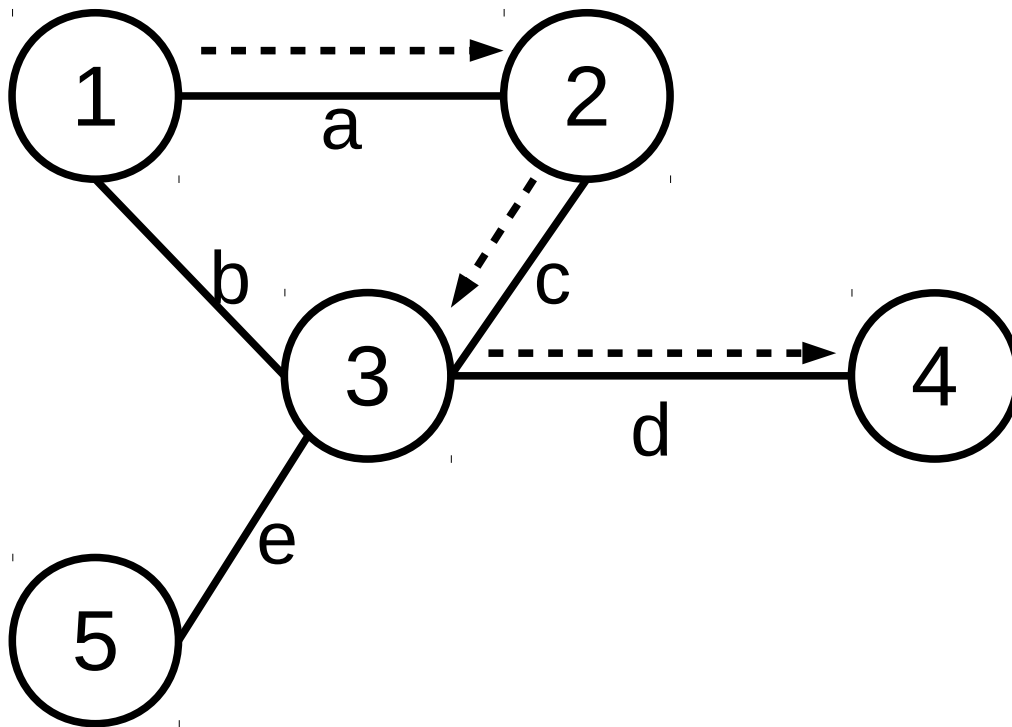
1 – a – 2

2 – c – 3

3 – d – 4

Passeio

- Exemplo:



1 – a – 2

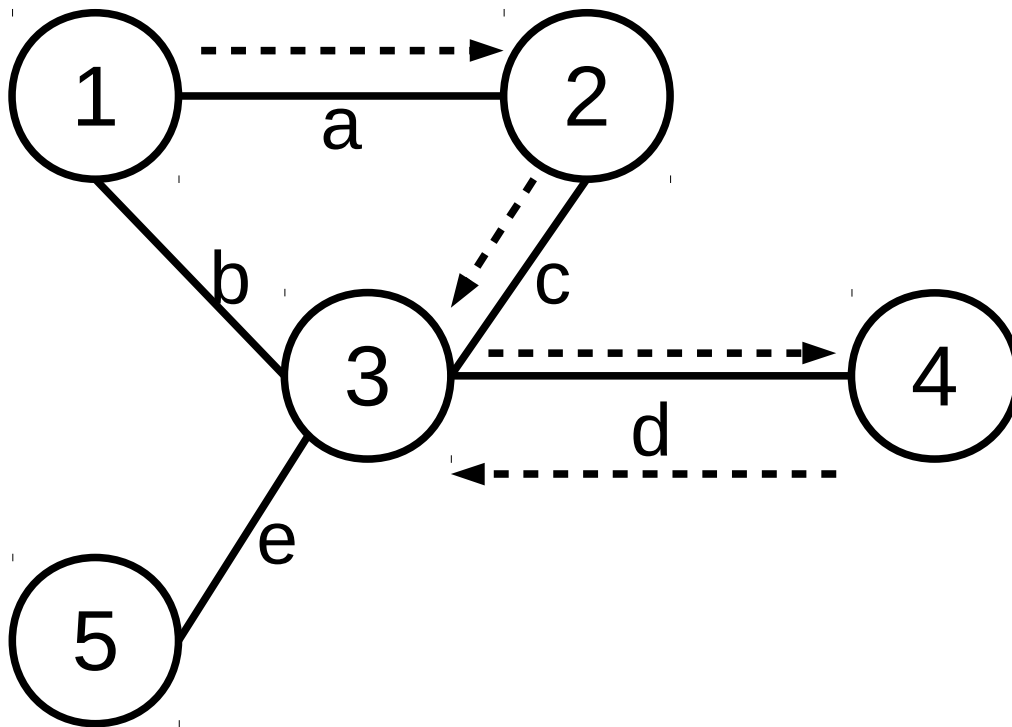
2 – c – 3

3 – d – 4

4 – d – 3

Passeio

- Exemplo:



1 – a – 2

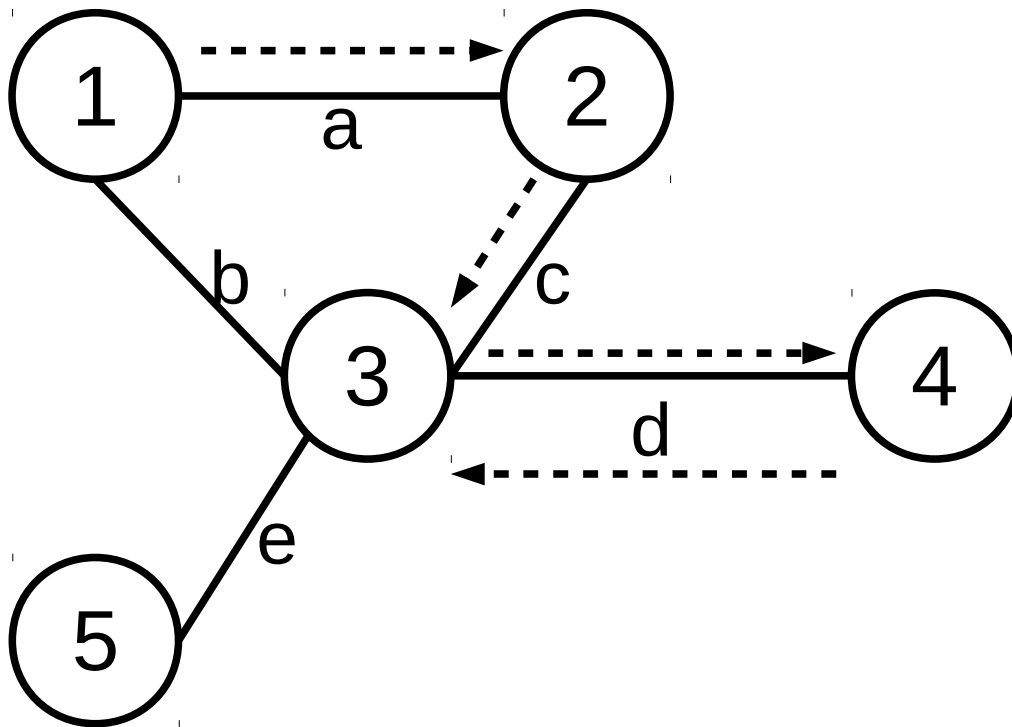
2 – c – 3

3 – d – 4

4 – d – 3

Passeio

- Exemplo:



1 – a – 2

2 – c – 3

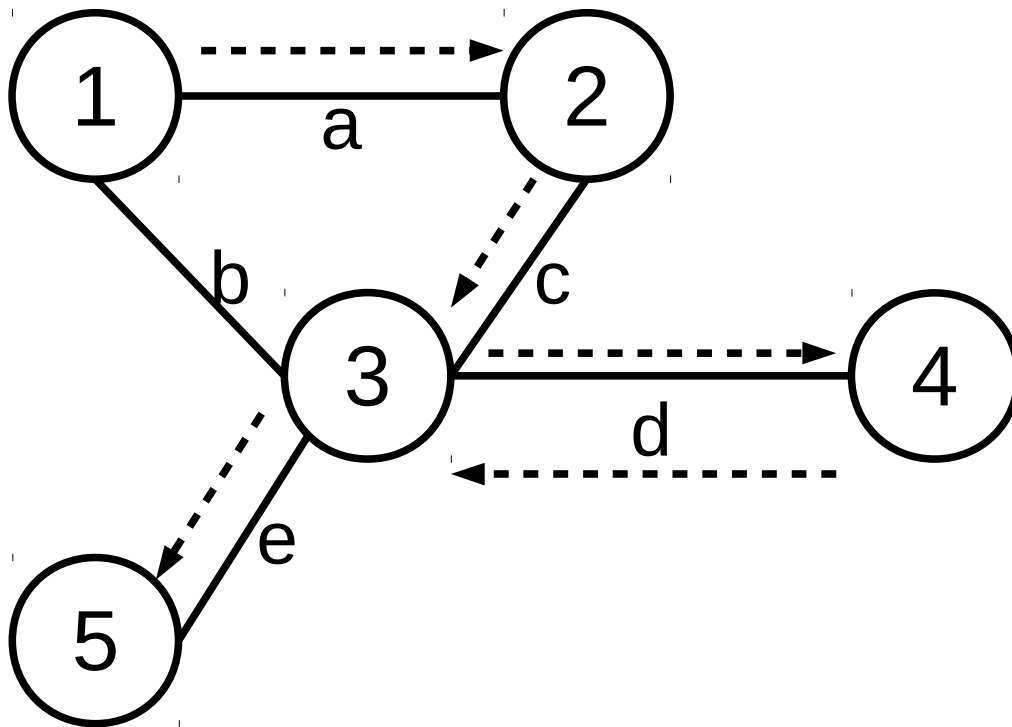
3 – d – 4

4 – d – 3

3 – e – 5

Passeio

- Exemplo:



1 – a – 2

2 – c – 3

3 – d – 4

4 – d – 3

3 – e – 5

Passeio

Passeio

- O passeio pode ser:

Passeio

- O passeio pode ser:
 - Aberto: quando inicia e acaba em vértices **diferentes** (o caso anterior).

Passeio

- O passeio pode ser:
 - Aberto: quando inicia e acaba em vértices **diferentes** (o caso anterior).
 - Fechado: quando inicia e acaba no **mesmo** vértice.
Exemplo: $1 - 2 - 3 - 4 - 5 - 3 - 1$.

Passeio

Passeio

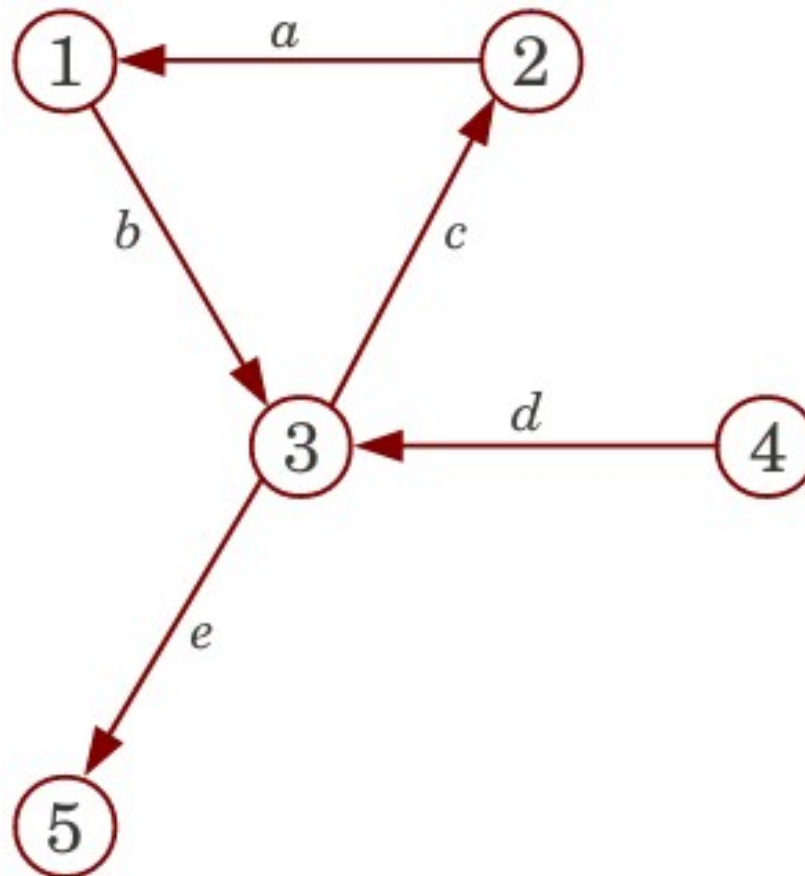
- Cadeia é um passeio que não repete arestas.

Passeio

- Cadeia é um passeio que não repete arestas.
- Exemplo: $4 - 3 - 2 - 1 - 3 - 5$.

Passeio

- Cadeia é um passeio que não repete arestas.
- Exemplo: 4 – 3 – 2 – 1 – 3 – 5.



Passeio

Passeio

- Caminho é uma cadeia sem repetição de vértices.

Passeio

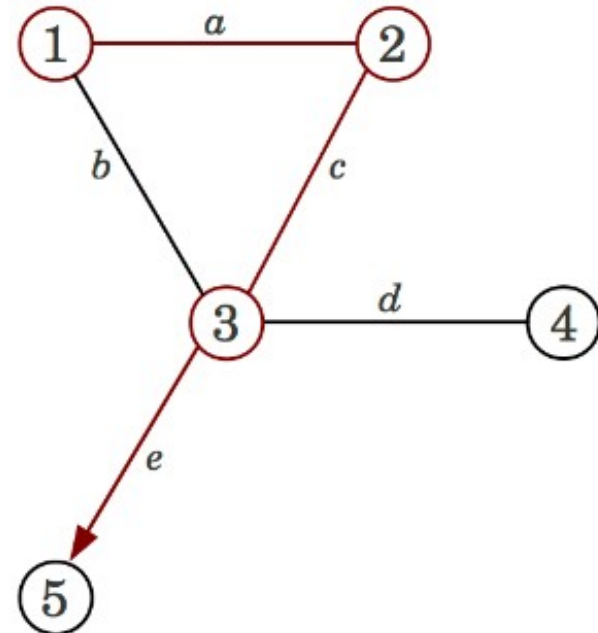
- Caminho é uma cadeia sem repetição de vértices.
- Exemplo: 1 – 2 – 3 – 5.

Passeio

- Caminho é uma cadeia sem repetição de vértices.
- Exemplo: 1 – 2 – 3 – 5.
- O comprimento de um caminho é o número de arestas que este inclui.

Passeio

- Caminho é uma cadeia sem repetição de vértices.
- Exemplo: 1 – 2 – 3 – 5.
- O comprimento de um caminho é o número de arestas que este inclui.



Ciclos

Ciclos

- Um **ciclo** é um caminho fechado.

Ciclos

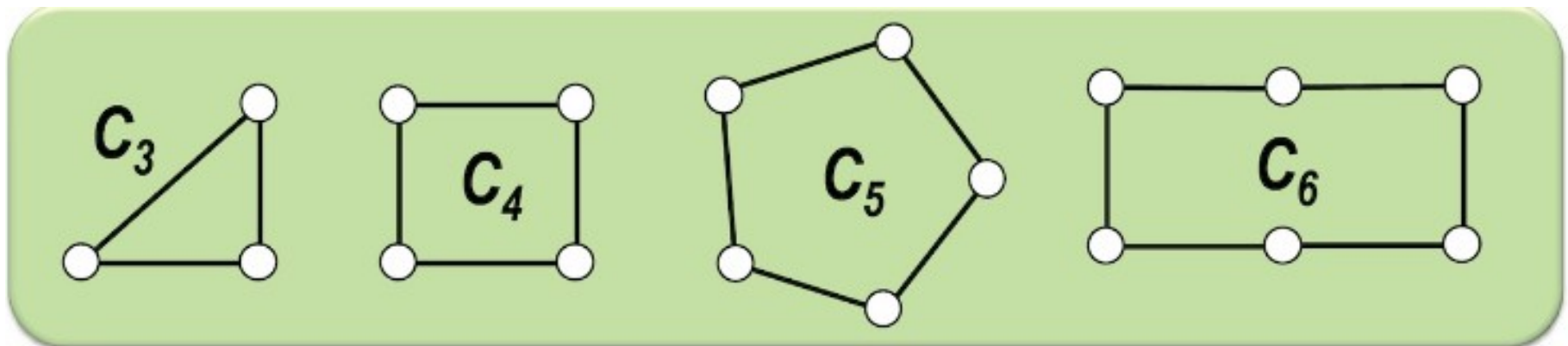
- Um **ciclo** é um caminho fechado.
- Alguns autores, utilizam o termo circuito para o caso de grafos orientados.

Ciclos

- Um **ciclo** é um caminho fechado.
- Alguns autores, utilizam o termo circuito para o caso de grafos orientados.
- Grafo Ciclo: é um grafo com n vértices formado por apenas um ciclo passando por todos os vértices.

Ciclos

- Um **ciclo** é um caminho fechado.
- Alguns autores, utilizam o termo circuito para o caso de grafos orientados.
- Grafo Ciclo: é um grafo com n vértices formado por apenas um ciclo passando por todos os vértices.



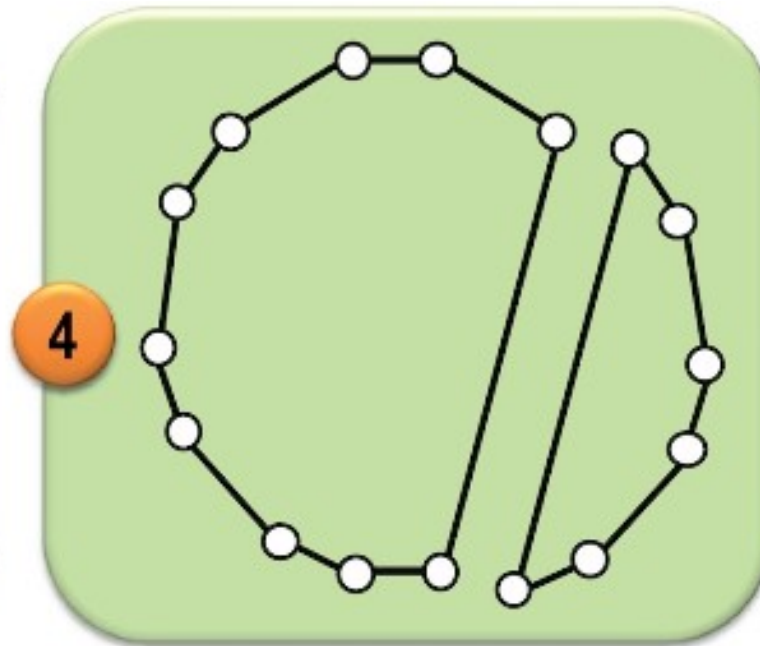
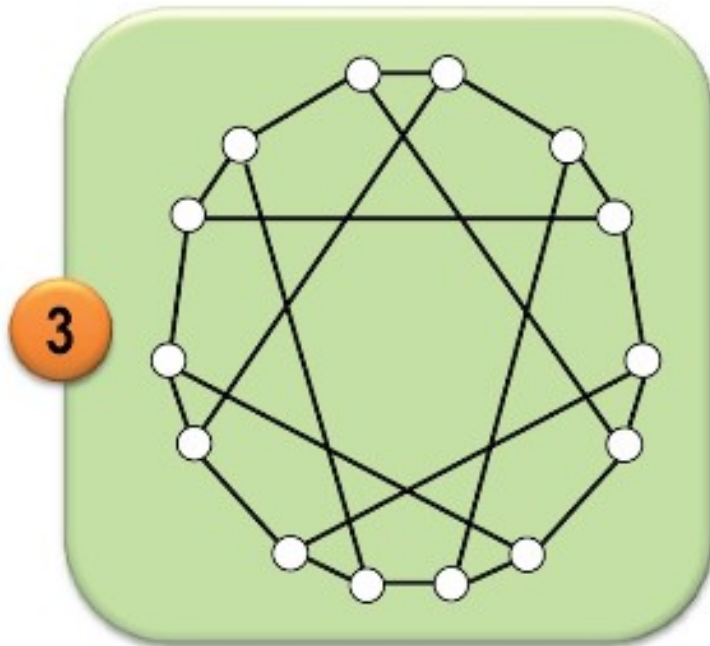
Ciclos

Ciclos

- A **cintura** de um grafo é o comprimento do menor ciclo existente no mesmo.

Ciclos

- A **cintura** de um grafo é o comprimento do menor ciclo existente no mesmo.



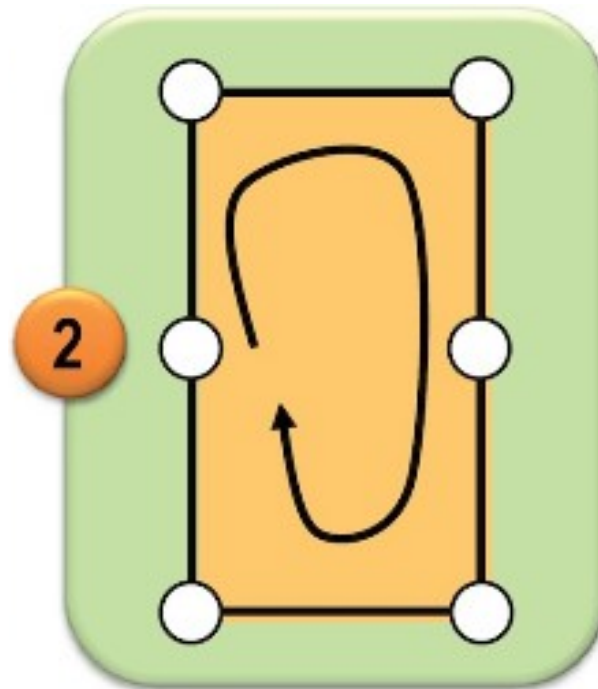
Ciclos

Ciclos

- A **circunferência** de um grafo é o comprimento do maior ciclo existente no mesmo.

Ciclos

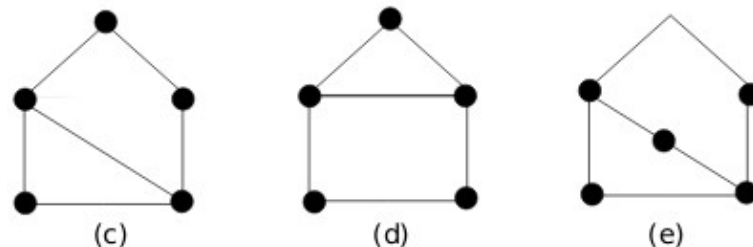
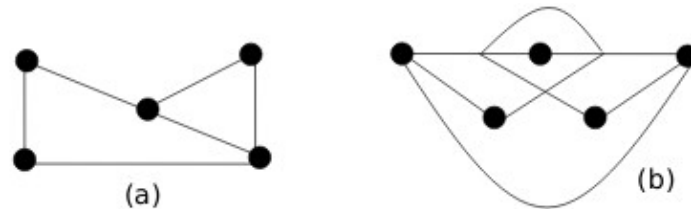
- A **circunferência** de um grafo é o comprimento do maior ciclo existente no mesmo.



EXERCÍCIOS

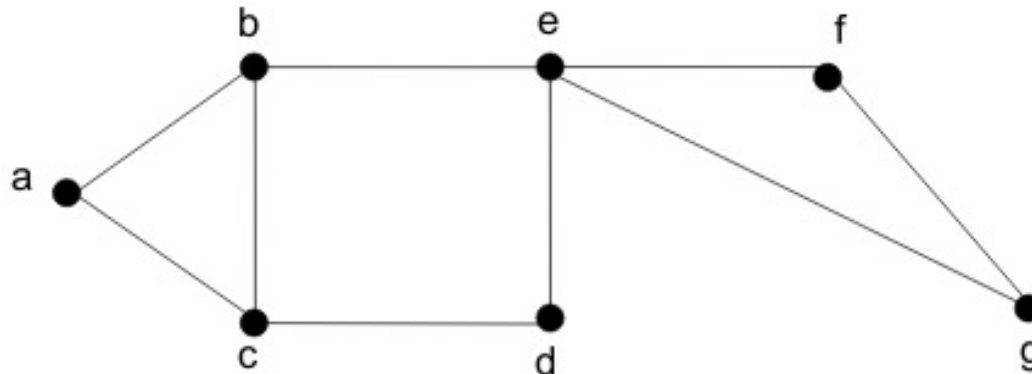
Exercícios

- Escreva um grafo com 4 vértices que seja isomorfo ao seu complemento (ou seja, auto-complementar).
- Qualé o número de arestas de um grafo que é isomorfo ao seu complemento?
- Considere os grafos abaixo. Qual deles é diferente dos demais?



Exercícios

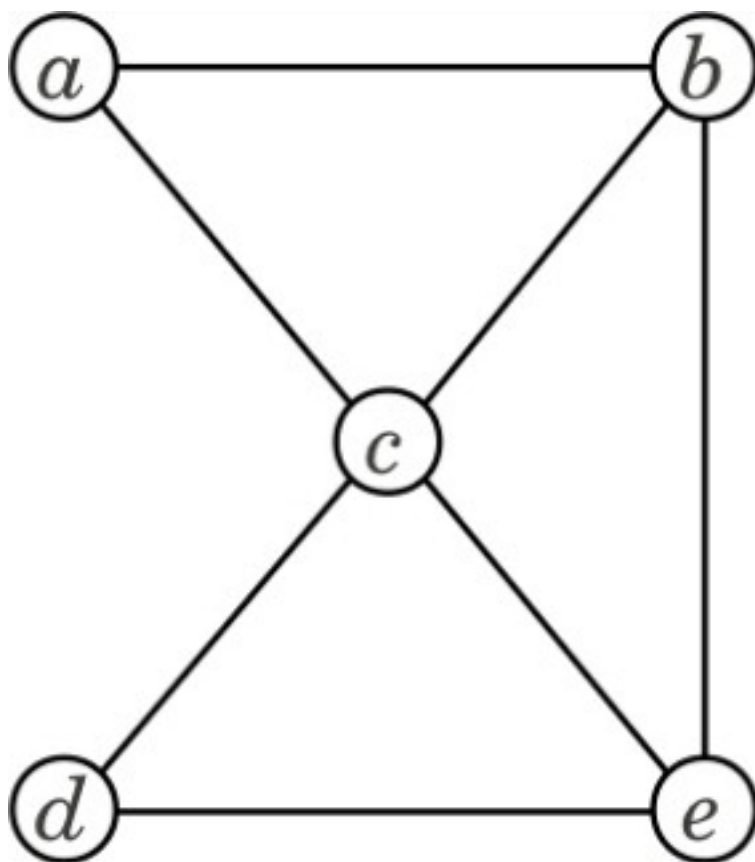
- Quantos caminhos existem entre os vértices b e f?



- Dê um exemplo de grafo conexo G cuja remoção de qualquer aresta torna G desconexo.
- Quantas arestas possui um grafo com essas características?

Exercícios

- Quantos grafos ciclos (não isomorfos) são subgrafos do grafo abaixo?



Exercícios

- Qual é a cintura e a circunferência do grafo abaixo?

