

Tema 4-2

Representación, isomorfismos e invariantes



UNIVERSIDAD
DE CHILE

Arturo Merino



Recuerdo/Plan

■ Recuerdo y plan

- **Recuerdo:**

Recuerdo y plan

- **Recuerdo:**
 - Grafos, vecindades y grados.

Recuerdo y plan

- **Recuerdo:**

- Grafos, vecindades y grados.
- Familias de grafos, subgrafos y complemento.

Recuerdo y plan

- **Recuerdo:**

- Grafos, vecindades y grados.
- Familias de grafos, subgrafos y complemento.

- **Plan:**

Recuerdo y plan

- **Recuerdo:**

- Grafos, vecindades y grados.
- Familias de grafos, subgrafos y complemento.

- **Plan:**

- Representaciones de grafos.

Recuerdo y plan

- **Recuerdo:**

- Grafos, vecindades y grados.
- Familias de grafos, subgrafos y complemento.

- **Plan:**

- Representaciones de grafos.
- Isomorfismos: misma estructura.

Recuerdo y plan

- **Recuerdo:**

- Grafos, vecindades y grados.
- Familias de grafos, subgrafos y complemento.

- **Plan:**

- Representaciones de grafos.
- Isomorfismos: misma estructura.
- Invariantes.

Representación

Matriz de adyacencia

- ¿Cómo almacenar un grafo de manera eficiente?

Matriz de adyacencia

- ¿Cómo almacenar un grafo de manera eficiente?

Def.

Sea $G = (V, E)$ un grafo simple con $V = \{v_1, \dots, v_n\}$.

Matriz de adyacencia

- ¿Cómo almacenar un grafo de manera eficiente?

Def.

Sea $G = (V, E)$ un grafo simple con $V = \{v_1, \dots, v_n\}$.

La *matriz de adyacencia* de G es la matriz $A = (a_{ij}) \in \{0, 1\}^{n \times n}$ definida por

Matriz de adyacencia

- ¿Cómo almacenar un grafo de manera eficiente?

Def.

Sea $G = (V, E)$ un grafo simple con $V = \{v_1, \dots, v_n\}$.

La *matriz de adyacencia* de G es la matriz $A = (a_{ij}) \in \{0, 1\}^{n \times n}$ definida por

$$a_{ij} = \begin{cases} 1 & \text{si } v_i v_j \in E, \\ 0 & \text{si } v_i v_j \notin E. \end{cases}$$

Matriz de adyacencia

- ¿Cómo almacenar un grafo de manera eficiente?

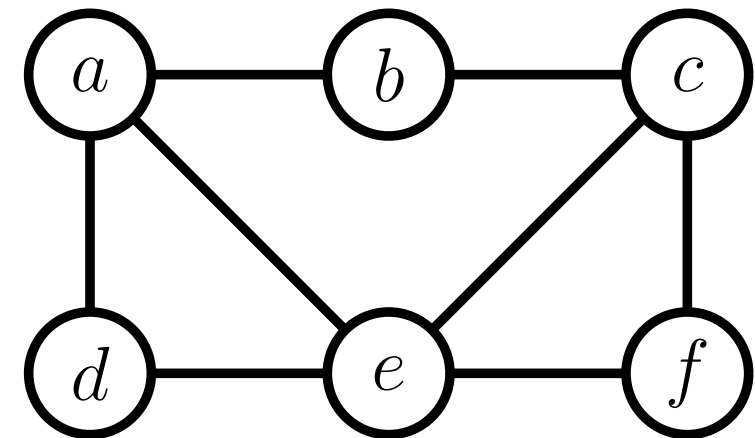
Def.

Sea $G = (V, E)$ un grafo simple con $V = \{v_1, \dots, v_n\}$.

La *matriz de adyacencia* de G es la matriz $A = (a_{ij}) \in \{0, 1\}^{n \times n}$ definida por

$$a_{ij} = \begin{cases} 1 & \text{si } v_i v_j \in E, \\ 0 & \text{si } v_i v_j \notin E. \end{cases}$$

- **Ej:** ¿Matriz de adyacencia en el orden a, b, c, d, e, f ?



Matriz de adyacencia

- ¿Cómo almacenar un grafo de manera eficiente?

Def.

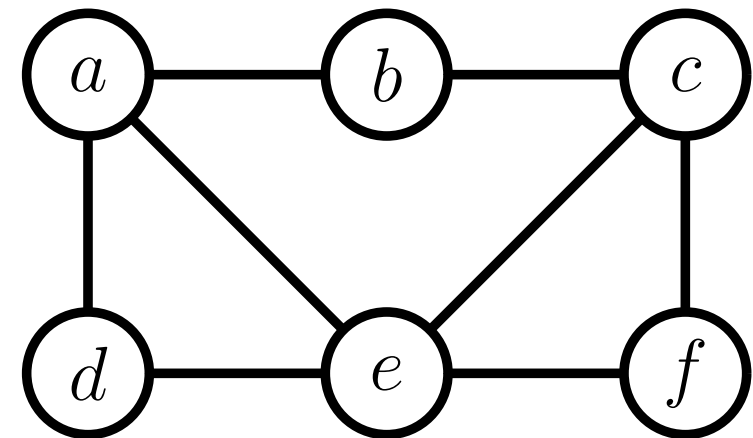
Sea $G = (V, E)$ un grafo simple con $V = \{v_1, \dots, v_n\}$.

La *matriz de adyacencia* de G es la matriz $A = (a_{ij}) \in \{0, 1\}^{n \times n}$ definida por

$$a_{ij} = \begin{cases} 1 & \text{si } v_i v_j \in E, \\ 0 & \text{si } v_i v_j \notin E. \end{cases}$$

- **Ej:** ¿Matriz de adyacencia en el orden a, b, c, d, e, f ?

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$



Matriz de adyacencia

- ¿Cómo almacenar un grafo de manera eficiente?

Def.

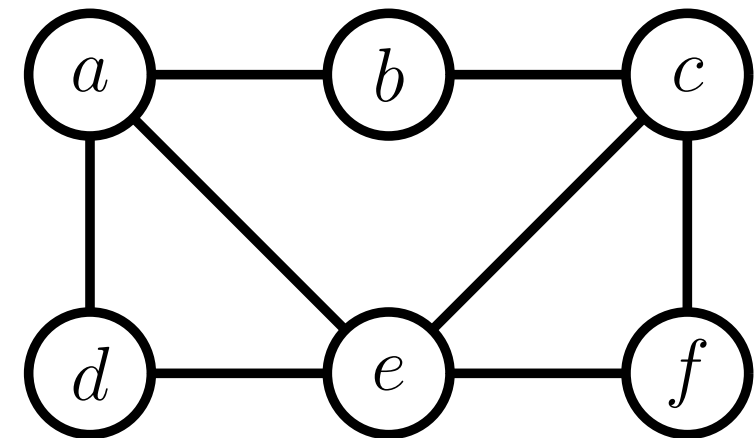
Sea $G = (V, E)$ un grafo simple con $V = \{v_1, \dots, v_n\}$.

La *matriz de adyacencia* de G es la matriz $A = (a_{ij}) \in \{0, 1\}^{n \times n}$ definida por

$$a_{ij} = \begin{cases} 1 & \text{si } v_i v_j \in E, \\ 0 & \text{si } v_i v_j \notin E. \end{cases}$$

- **Ej:** ¿Matriz de adyacencia en el orden a, b, c, d, e, f ?

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \end{pmatrix}.$$



Matriz de adyacencia

- ¿Cómo almacenar un grafo de manera eficiente?

Def.

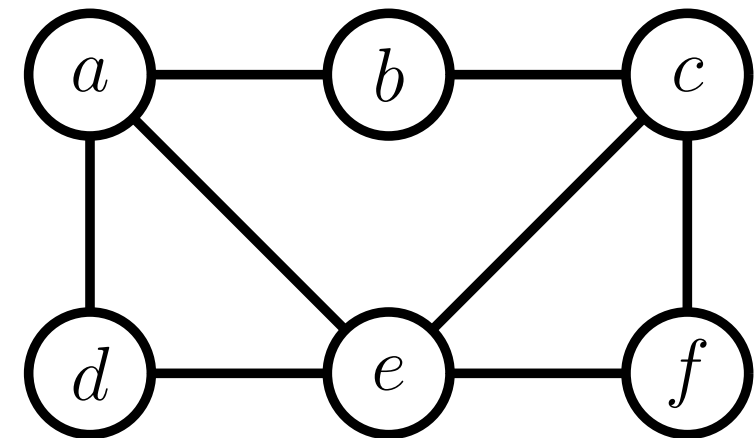
Sea $G = (V, E)$ un grafo simple con $V = \{v_1, \dots, v_n\}$.

La *matriz de adyacencia* de G es la matriz $A = (a_{ij}) \in \{0, 1\}^{n \times n}$ definida por

$$a_{ij} = \begin{cases} 1 & \text{si } v_i v_j \in E, \\ 0 & \text{si } v_i v_j \notin E. \end{cases}$$

- **Ej:** ¿Matriz de adyacencia en el orden a, b, c, d, e, f ?

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$



Matriz de adyacencia

- ¿Cómo almacenar un grafo de manera eficiente?

Def.

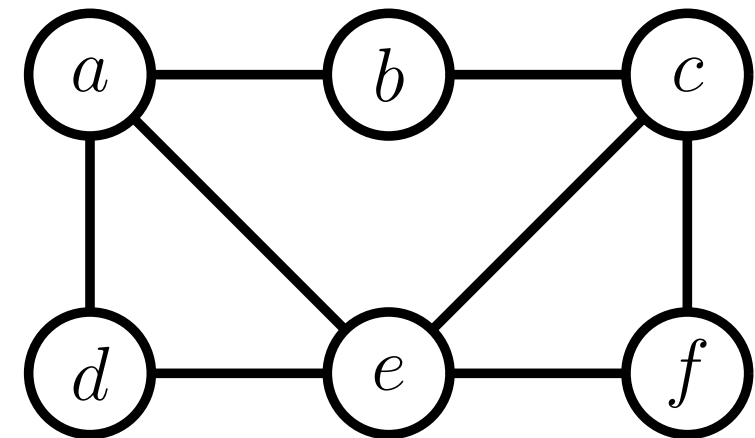
Sea $G = (V, E)$ un grafo simple con $V = \{v_1, \dots, v_n\}$.

La *matriz de adyacencia* de G es la matriz $A = (a_{ij}) \in \{0, 1\}^{n \times n}$ definida por

$$a_{ij} = \begin{cases} 1 & \text{si } v_i v_j \in E, \\ 0 & \text{si } v_i v_j \notin E. \end{cases}$$

- **Ej:** ¿Matriz de adyacencia en el orden a, b, c, d, e, f ?

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$



Matriz de adyacencia

- ¿Cómo almacenar un grafo de manera eficiente?

Def.

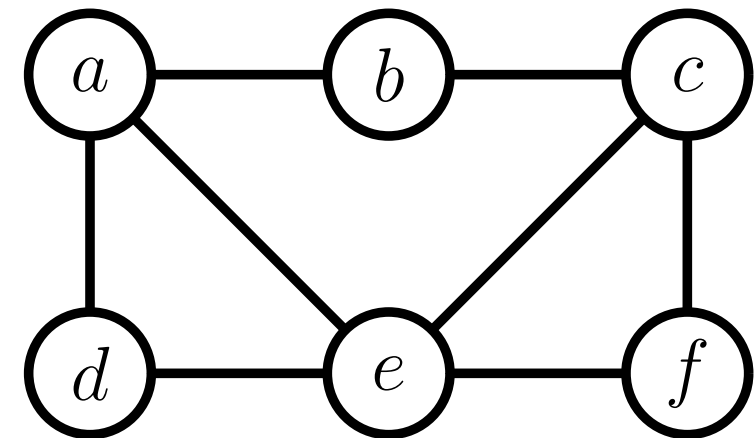
Sea $G = (V, E)$ un grafo simple con $V = \{v_1, \dots, v_n\}$.

La *matriz de adyacencia* de G es la matriz $A = (a_{ij}) \in \{0, 1\}^{n \times n}$ definida por

$$a_{ij} = \begin{cases} 1 & \text{si } v_i v_j \in E, \\ 0 & \text{si } v_i v_j \notin E. \end{cases}$$

- **Ej:** ¿Matriz de adyacencia en el orden a, b, c, d, e, f ?

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}.$$



Matriz de adyacencia

- ¿Cómo almacenar un grafo de manera eficiente?

Def.

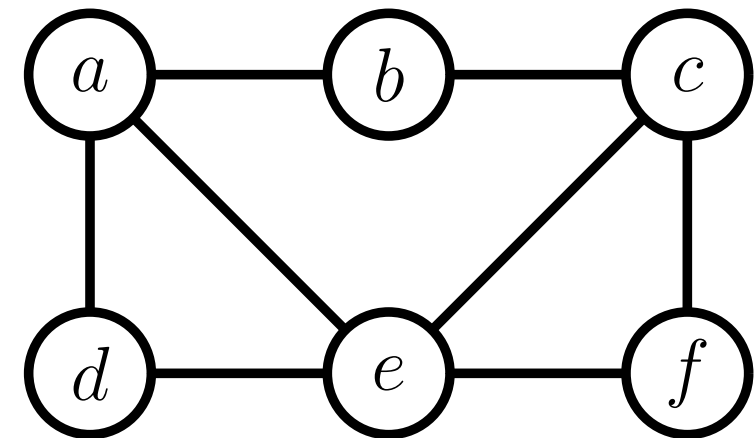
Sea $G = (V, E)$ un grafo simple con $V = \{v_1, \dots, v_n\}$.

La *matriz de adyacencia* de G es la matriz $A = (a_{ij}) \in \{0, 1\}^{n \times n}$ definida por

$$a_{ij} = \begin{cases} 1 & \text{si } v_i v_j \in E, \\ 0 & \text{si } v_i v_j \notin E. \end{cases}$$

- **Ej:** ¿Matriz de adyacencia en el orden a, b, c, d, e, f ?

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}.$$



Matriz de adyacencia

Prop.

Sea G un grafo simple y sea A su matriz de adyacencia.

Matriz de adyacencia

Prop.

Sea G un grafo simple y sea A su matriz de adyacencia.

Entonces, A es simétrica y su diagonal sólo contiene ceros.

Matriz de adyacencia

Prop.

Sea G un grafo simple y sea A su matriz de adyacencia.

Entonces, A es simétrica y su diagonal sólo contiene ceros.

- **Dem:**

Matriz de adyacencia

Prop.

Sea G un grafo simple y sea A su matriz de adyacencia.

Entonces, A es simétrica y su diagonal sólo contiene ceros.

- **Dem:**
- La matriz nos da información *adicional* ✓

Matriz de adyacencia

Prop.

Sea G un grafo simple y sea A su matriz de adyacencia.

Entonces, A es simétrica y su diagonal sólo contiene ceros.

- **Dem:**
- La matriz nos da información *adicional* ✓

Prop.

Sea G un grafo simple con vértices $v_1, \dots, v_n$ y matriz de adyacencia $A = (a_{ij})$.

Matriz de adyacencia

Prop.

Sea G un grafo simple y sea A su matriz de adyacencia.
Entonces, A es simétrica y su diagonal sólo contiene ceros.

- **Dem:**
- La matriz nos da información *adicional* ✓

Prop.

Sea G un grafo simple con vértices $v_1, \dots, v_n$ y matriz de adyacencia $A = (a_{ij})$.

1. Si $i \neq j$, $(A^2)_{ij}$ es el número de vecinos comunes de v_i y v_j .
2. Para todo $i \in \{1, \dots, n\}$, $(A^2)_{ii} = d(v_i)$.

Matriz de adyacencia

Prop.

Sea G un grafo simple y sea A su matriz de adyacencia.
Entonces, A es simétrica y su diagonal sólo contiene ceros.

- **Dem:**
- La matriz nos da información *adicional* ✓

Prop.

Sea G un grafo simple con vértices $v_1, \dots, v_n$ y matriz de adyacencia $A = (a_{ij})$.

1. Si $i \neq j$, $(A^2)_{ij}$ es el número de vecinos comunes de v_i y v_j .
2. Para todo $i \in \{1, \dots, n\}$, $(A^2)_{ii} = d(v_i)$.

- **Dem:**

Listas de adyacencia

- **Otra forma de guardar adyacencias:** listas de vecinos.

Listas de adyacencia

- **Otra forma de guardar adyacencias:** listas de vecinos.

Def.

Sea $G = (V, E)$ un grafo simple.

Listas de adyacencia

- **Otra forma de guardar adyacencias:** listas de vecinos.

Def.

Sea $G = (V, E)$ un grafo simple.

Una *lista de adyacencia* de G es una representación que, para cada vértice $v \in V$, almacena la lista de vértices adyacentes a v .

Listas de adyacencia

- **Otra forma de guardar adyacencias:** listas de vecinos.

Def.

Sea $G = (V, E)$ un grafo simple.

Una *lista de adyacencia* de G es una representación que, para cada vértice $v \in V$, almacena la lista de vértices adyacentes a v .

- Ej:

Listas de adyacencia

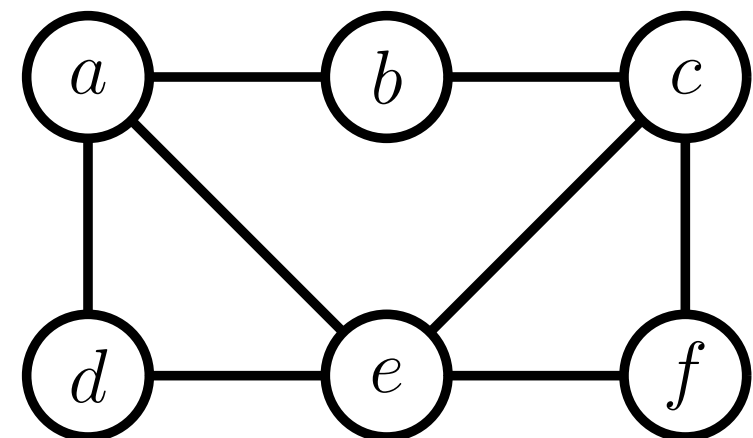
- Otra forma de guardar adyacencias: listas de vecinos.

Def.

Sea $G = (V, E)$ un grafo simple.

Una *lista de adyacencia* de G es una representación que, para cada vértice $v \in V$, almacena la lista de vértices adyacentes a v .

- Ej:
 - ¿Listas de adyacencia?



Listas de adyacencia

- Otra forma de guardar adyacencias: listas de vecinos.

Def.

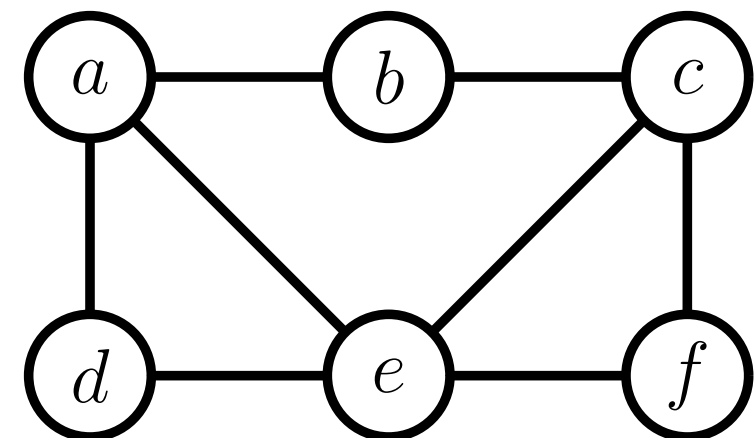
Sea $G = (V, E)$ un grafo simple.

Una *lista de adyacencia* de G es una representación que, para cada vértice $v \in V$, almacena la lista de vértices adyacentes a v .

- Ej:

- ¿Listas de adyacencia?

$a : b, d, e$



Listas de adyacencia

- Otra forma de guardar adyacencias: listas de vecinos.

Def.

Sea $G = (V, E)$ un grafo simple.

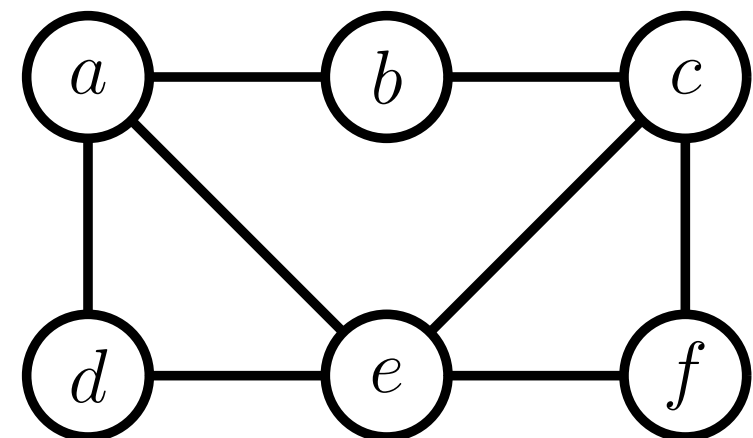
Una *lista de adyacencia* de G es una representación que, para cada vértice $v \in V$, almacena la lista de vértices adyacentes a v .

- Ej:

- ¿Listas de adyacencia?

$a : b, d, e$

$b : a, c$



Listas de adyacencia

- Otra forma de guardar adyacencias: listas de vecinos.

Def.

Sea $G = (V, E)$ un grafo simple.

Una *lista de adyacencia* de G es una representación que, para cada vértice $v \in V$, almacena la lista de vértices adyacentes a v .

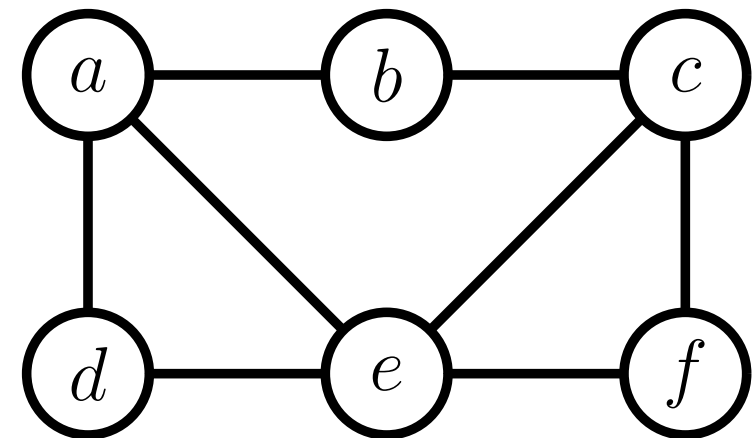
- Ej:

- ¿Listas de adyacencia?

$a : b, d, e$

$b : a, c$

$c : b, e, f$



Listas de adyacencia

- Otra forma de guardar adyacencias: listas de vecinos.

Def.

Sea $G = (V, E)$ un grafo simple.

Una *lista de adyacencia* de G es una representación que, para cada vértice $v \in V$, almacena la lista de vértices adyacentes a v .

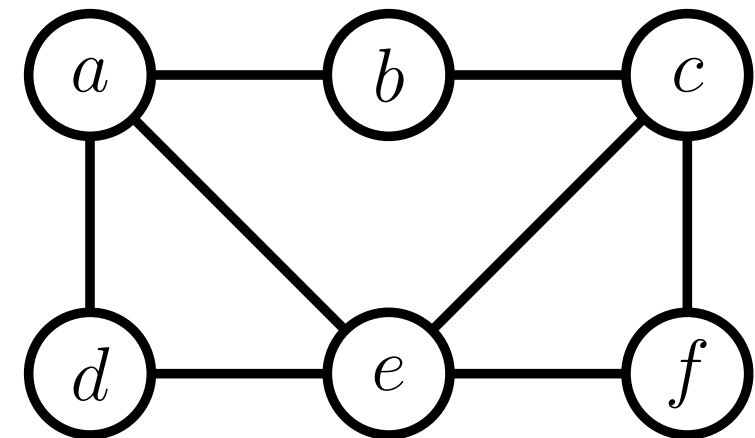
- Ej:

- ¿Listas de adyacencia?

$a : b, d, e$ $d : a, e$

$b : a, c$

$c : b, e, f$



Listas de adyacencia

- Otra forma de guardar adyacencias: listas de vecinos.

Def.

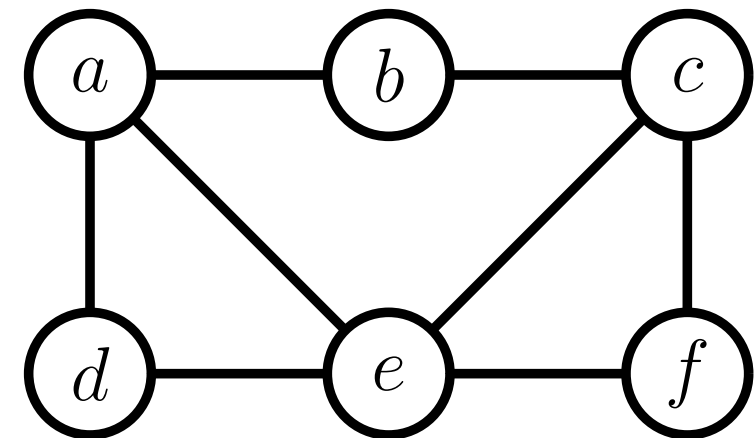
Sea $G = (V, E)$ un grafo simple.

Una *lista de adyacencia* de G es una representación que, para cada vértice $v \in V$, almacena la lista de vértices adyacentes a v .

- Ej:

- ¿Listas de adyacencia?

$a : b, d, e$	$d : a, e$
$b : a, c$	$e : a, c, d, f$
$c : b, e, f$	



Listas de adyacencia

- Otra forma de guardar adyacencias: listas de vecinos.

Def.

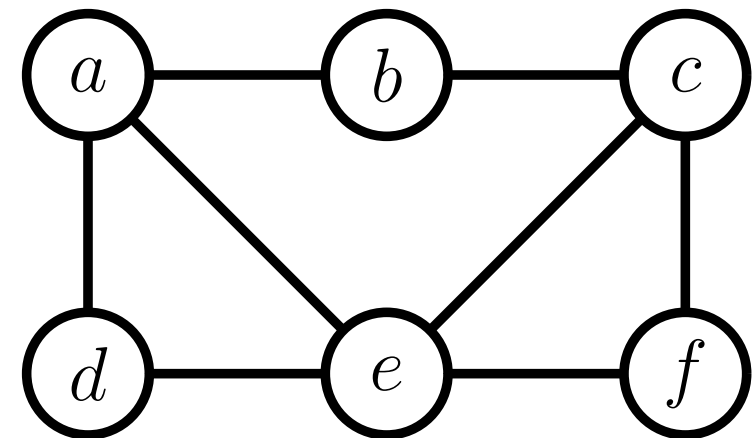
Sea $G = (V, E)$ un grafo simple.

Una *lista de adyacencia* de G es una representación que, para cada vértice $v \in V$, almacena la lista de vértices adyacentes a v .

- Ej:

- ¿Listas de adyacencia?

$a : b, d, e$	$d : a, e$
$b : a, c$	$e : a, c, d, f$
$c : b, e, f$	$f : c, e$



Matriz de incidencia

- Podemos también guardar *incidencias*.

Matriz de incidencia

- Podemos también guardar *incidencias*.

Def.

Sea $G = (V, E)$ un grafo con $V = \{v_1, \dots, v_n\}$ y $E = \{e_1, \dots, e_m\}$.

Matriz de incidencia

- Podemos también guardar *incidencias*.

Def.

Sea $G = (V, E)$ un grafo con $V = \{v_1, \dots, v_n\}$ y $E = \{e_1, \dots, e_m\}$.

La *matriz de incidencia* de G es la matriz $B = (b_{ij}) \in \{0, 1\}^{n \times m}$ definida por

Matriz de incidencia

- Podemos también guardar *incidencias*.

Def.

Sea $G = (V, E)$ un grafo con $V = \{v_1, \dots, v_n\}$ y $E = \{e_1, \dots, e_m\}$.

La *matriz de incidencia* de G es la matriz $B = (b_{ij}) \in \{0, 1\}^{n \times m}$ definida por

$$b_{ij} = \begin{cases} 1 & \text{si } e_j \text{ incide en } v_i, \\ 0 & \text{en otro caso.} \end{cases}$$

Matriz de incidencia

- Podemos también guardar *incidencias*.

Def.

Sea $G = (V, E)$ un grafo con $V = \{v_1, \dots, v_n\}$ y $E = \{e_1, \dots, e_m\}$.

La *matriz de incidencia* de G es la matriz $B = (b_{ij}) \in \{0, 1\}^{n \times m}$ definida por

$$b_{ij} = \begin{cases} 1 & \text{si } e_j \text{ incide en } v_i, \\ 0 & \text{en otro caso.} \end{cases}$$

- Ej:

Matriz de incidencia

- Podemos también guardar *incidencias*.

Def.

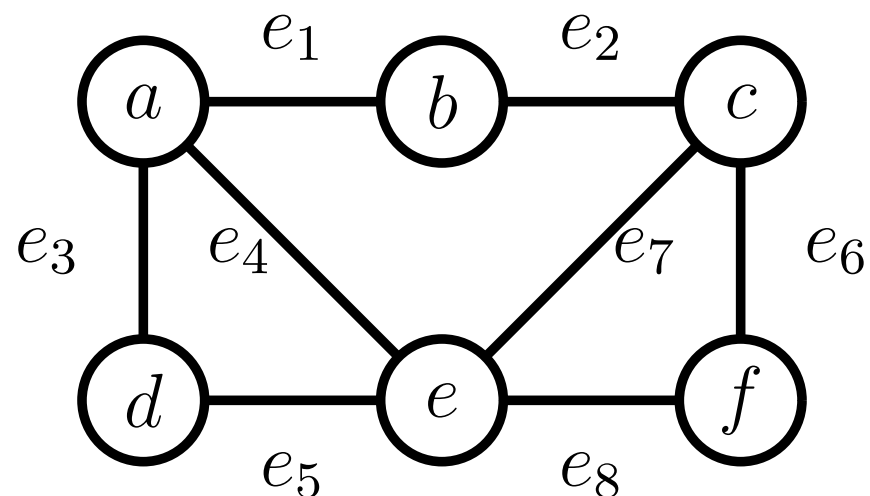
Sea $G = (V, E)$ un grafo con $V = \{v_1, \dots, v_n\}$ y $E = \{e_1, \dots, e_m\}$.

La *matriz de incidencia* de G es la matriz $B = (b_{ij}) \in \{0, 1\}^{n \times m}$ definida por

$$b_{ij} = \begin{cases} 1 & \text{si } e_j \text{ incide en } v_i, \\ 0 & \text{en otro caso.} \end{cases}$$

Ej:

- Filas: a, b, c, d, e, f . Columnas: $e_1, \dots, e_8$.



Matriz de incidencia

- Podemos también guardar *incidencias*.

Def.

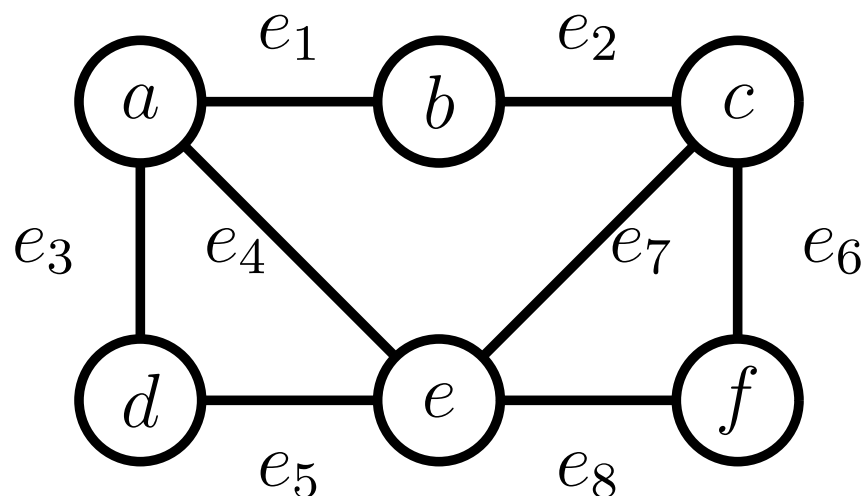
Sea $G = (V, E)$ un grafo con $V = \{v_1, \dots, v_n\}$ y $E = \{e_1, \dots, e_m\}$.

La *matriz de incidencia* de G es la matriz $B = (b_{ij}) \in \{0, 1\}^{n \times m}$ definida por

$$b_{ij} = \begin{cases} 1 & \text{si } e_j \text{ incide en } v_i, \\ 0 & \text{en otro caso.} \end{cases}$$

- Ej:**

○ Filas: a, b, c, d, e, f . Columnas: $e_1, \dots, e_8$.



$$B = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Matriz de incidencia

- Podemos también guardar *incidencias*.

Def.

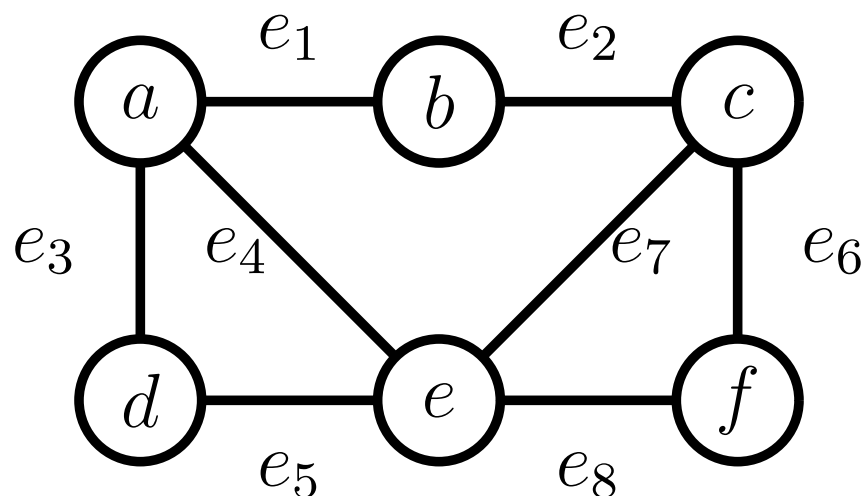
Sea $G = (V, E)$ un grafo con $V = \{v_1, \dots, v_n\}$ y $E = \{e_1, \dots, e_m\}$.

La *matriz de incidencia* de G es la matriz $B = (b_{ij}) \in \{0, 1\}^{n \times m}$ definida por

$$b_{ij} = \begin{cases} 1 & \text{si } e_j \text{ incide en } v_i, \\ 0 & \text{en otro caso.} \end{cases}$$

- Ej:**

○ Filas: a, b, c, d, e, f . Columnas: $e_1, \dots, e_8$.



$$B = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Matriz de incidencia

- Podemos también guardar *incidencias*.

Def.

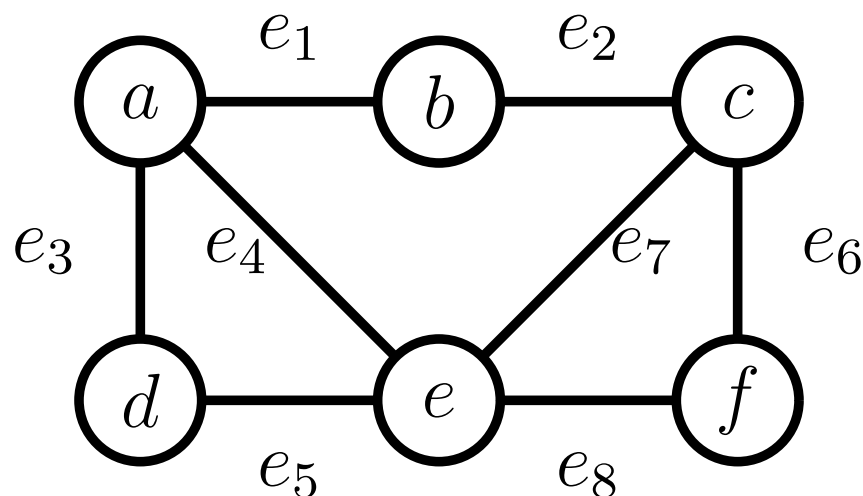
Sea $G = (V, E)$ un grafo con $V = \{v_1, \dots, v_n\}$ y $E = \{e_1, \dots, e_m\}$.

La *matriz de incidencia* de G es la matriz $B = (b_{ij}) \in \{0, 1\}^{n \times m}$ definida por

$$b_{ij} = \begin{cases} 1 & \text{si } e_j \text{ incide en } v_i, \\ 0 & \text{en otro caso.} \end{cases}$$

- Ej:**

○ Filas: a, b, c, d, e, f . Columnas: $e_1, \dots, e_8$.



$$B = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

Matriz de incidencia

- Podemos también guardar *incidencias*.

Def.

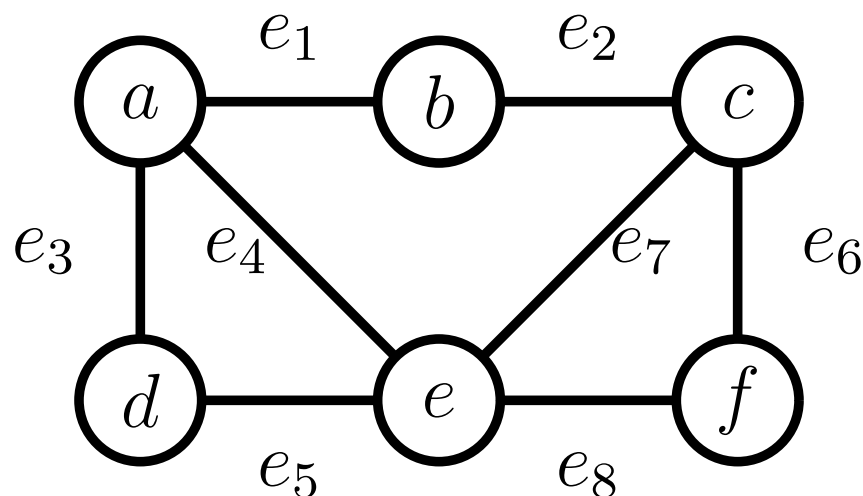
Sea $G = (V, E)$ un grafo con $V = \{v_1, \dots, v_n\}$ y $E = \{e_1, \dots, e_m\}$.

La *matriz de incidencia* de G es la matriz $B = (b_{ij}) \in \{0, 1\}^{n \times m}$ definida por

$$b_{ij} = \begin{cases} 1 & \text{si } e_j \text{ incide en } v_i, \\ 0 & \text{en otro caso.} \end{cases}$$

- Ej:**

○ Filas: a, b, c, d, e, f . Columnas: $e_1, \dots, e_8$.



$$B = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

Matriz de incidencia

- Podemos también guardar *incidencias*.

Def.

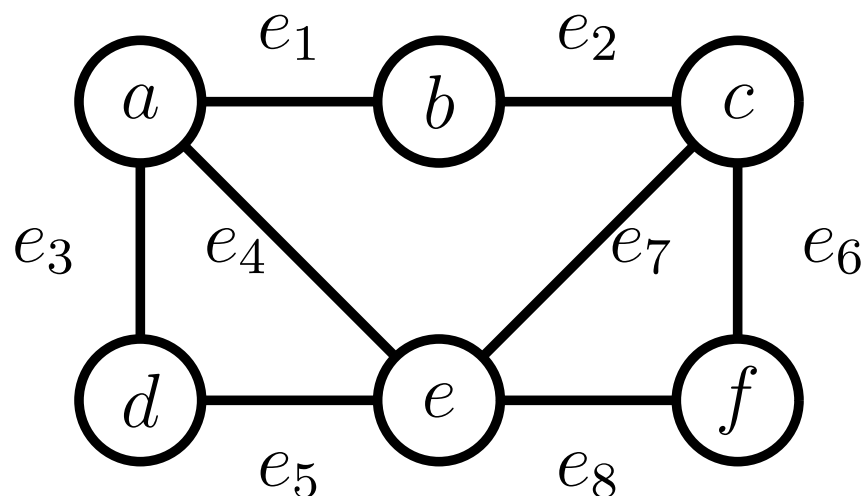
Sea $G = (V, E)$ un grafo con $V = \{v_1, \dots, v_n\}$ y $E = \{e_1, \dots, e_m\}$.

La *matriz de incidencia* de G es la matriz $B = (b_{ij}) \in \{0, 1\}^{n \times m}$ definida por

$$b_{ij} = \begin{cases} 1 & \text{si } e_j \text{ incide en } v_i, \\ 0 & \text{en otro caso.} \end{cases}$$

Ej:

- Filas: a, b, c, d, e, f . Columnas: $e_1, \dots, e_8$.



$$B = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

Matriz de incidencia

- Podemos también guardar *incidencias*.

Def.

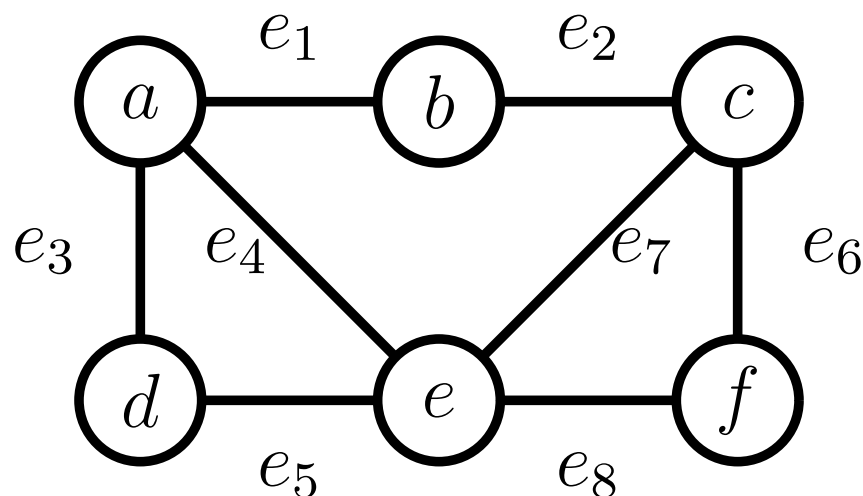
Sea $G = (V, E)$ un grafo con $V = \{v_1, \dots, v_n\}$ y $E = \{e_1, \dots, e_m\}$.

La *matriz de incidencia* de G es la matriz $B = (b_{ij}) \in \{0, 1\}^{n \times m}$ definida por

$$b_{ij} = \begin{cases} 1 & \text{si } e_j \text{ incide en } v_i, \\ 0 & \text{en otro caso.} \end{cases}$$

- Ej:**

○ Filas: a, b, c, d, e, f . Columnas: $e_1, \dots, e_8$.



$$B = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Comparando representaciones

- Para un grafo simple con n vértices y m aristas:

Comparando representaciones

- Para un grafo simple con n vértices y m aristas:
 - ¿Cuánto espacio ocupa cada representación?

Comparando representaciones

- Para un grafo simple con n vértices y m aristas:
 - ¿Cuánto espacio ocupa cada representación?
 - ¿Cómo consultamos si $uv \in E$?

Comparando representaciones

- Para un grafo simple con n vértices y m aristas:
 - ¿Cuánto espacio ocupa cada representación?
 - ¿Cómo consultamos si $uv \in E$?
 - ¿Cómo recorremos los vecinos de un vértice?

Comparando representaciones

- Para un grafo simple con n vértices y m aristas:
 - ¿Cuánto espacio ocupa cada representación?
 - ¿Cómo consultamos si $uv \in E$?
 - ¿Cómo recorremos los vecinos de un vértice?
- ¿Qué cambia con aristas paralelas o loops?

Comparando representaciones

- Para un grafo simple con n vértices y m aristas:
 - ¿Cuánto espacio ocupa cada representación?
 - ¿Cómo consultamos si $uv \in E$?
 - ¿Cómo recorremos los vecinos de un vértice?
- ¿Qué cambia con aristas paralelas o loops?
- ¿Y si las aristas tienen dirección?

Comparando representaciones

- Para un grafo simple con n vértices y m aristas:
 - ¿Cuánto espacio ocupa cada representación?
 - ¿Cómo consultamos si $uv \in E$?
 - ¿Cómo recorremos los vecinos de un vértice?
- ¿Qué cambia con aristas paralelas o loops?
- ¿Y si las aristas tienen dirección?
 - Adyacencia: $a_{ij} = 1$ si existe el arco $v_i \rightarrow v_j$.

Comparando representaciones

- Para un grafo simple con n vértices y m aristas:
 - ¿Cuánto espacio ocupa cada representación?
 - ¿Cómo consultamos si $uv \in E$?
 - ¿Cómo recorremos los vecinos de un vértice?
- ¿Qué cambia con aristas paralelas o loops?
- ¿Y si las aristas tienen dirección?
 - Adyacencia: $a_{ij} = 1$ si existe el arco $v_i \rightarrow v_j$.
 - Listas: podemos guardar sólo los vecinos de salida.

Comparando representaciones

- Para un grafo simple con n vértices y m aristas:
 - ¿Cuánto espacio ocupa cada representación?
 - ¿Cómo consultamos si $uv \in E$?
 - ¿Cómo recorremos los vecinos de un vértice?
- ¿Qué cambia con aristas paralelas o loops?
- ¿Y si las aristas tienen dirección?
 - Adyacencia: $a_{ij} = 1$ si existe el arco $v_i \rightarrow v_j$.
 - Listas: podemos guardar sólo los vecinos de salida.
 - Incidencia: distinguimos el extremo inicial y el final.

Isomorfismos e invariantes

Isomorfismos

- Nos interesan los grafos, **no** sus etiquetas.

Isomorfismos

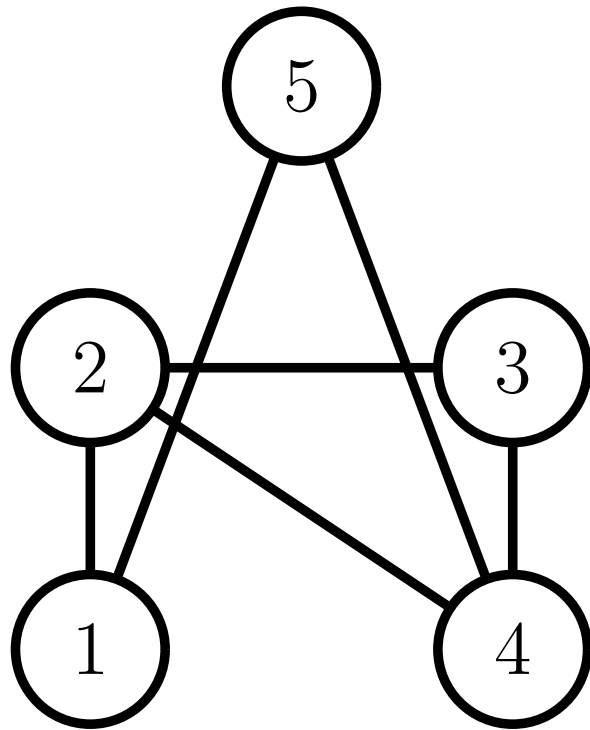
- Nos interesan los grafos, **no** sus etiquetas.
- Ej:

Isomorfismos

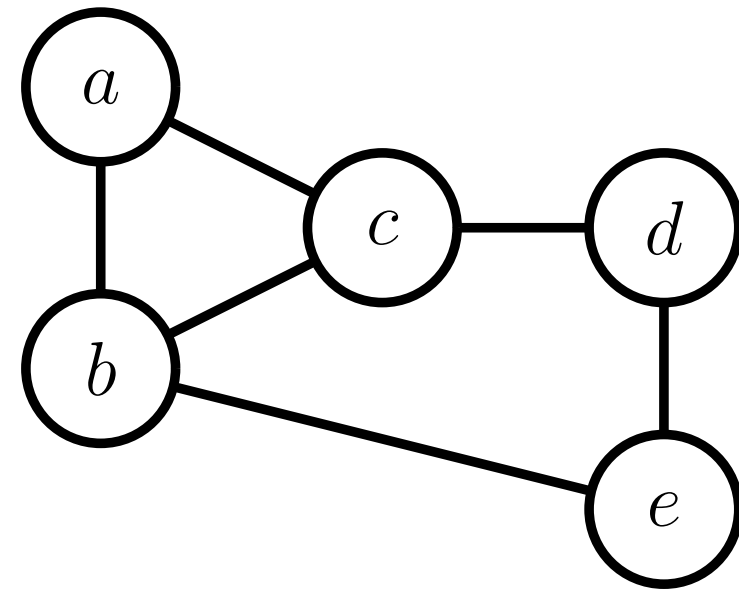
- Nos interesan los grafos, **no** sus etiquetas.

- Ej:

- ¿Misma o diferente estructura?



G



H

Isomorfismos

Def.

Sean $G = (V, E)$ y $H = (V', E')$ dos grafos simples.

Isomorfismos

Def.

Sean $G = (V, E)$ y $H = (V', E')$ dos grafos simples.

Diremos que G y H son *isomorfos* ($G \cong H$) si existe una biyección $f: V \rightarrow V'$ tal que, para todo $u, v \in V$,

Isomorfismos

Def.

Sean $G = (V, E)$ y $H = (V', E')$ dos grafos simples.

Diremos que G y H son *isomorfos* ($G \cong H$) si existe una biyección $f: V \rightarrow V'$ tal que, para todo $u, v \in V$,

$$uv \in E \iff f(u)f(v) \in E'.$$

Isomorfismos

Def.

Sean $G = (V, E)$ y $H = (V', E')$ dos grafos simples.

Diremos que G y H son *isomorfos* ($G \cong H$) si existe una biyección $f: V \rightarrow V'$ tal que, para todo $u, v \in V$,

$$uv \in E \iff f(u)f(v) \in E'.$$

En ese caso, f es un *isomorfismo* entre G y H .

Isomorfismos

Def.

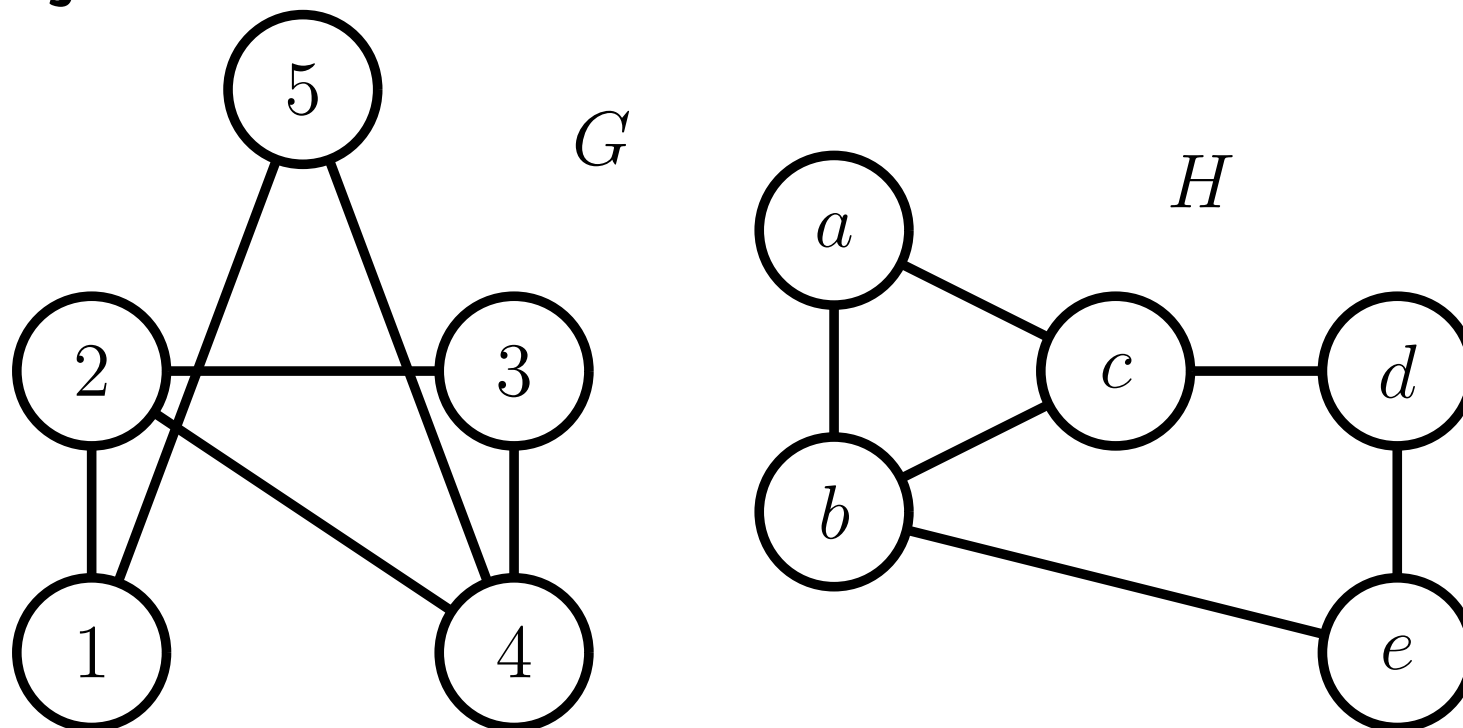
Sean $G = (V, E)$ y $H = (V', E')$ dos grafos simples.

Diremos que G y H son *isomorfos* ($G \cong H$) si existe una biyección $f: V \rightarrow V'$ tal que, para todo $u, v \in V$,

$$uv \in E \iff f(u)f(v) \in E'.$$

En ese caso, f es un *isomorfismo* entre G y H .

• Ej:



Isomorfismos

Def.

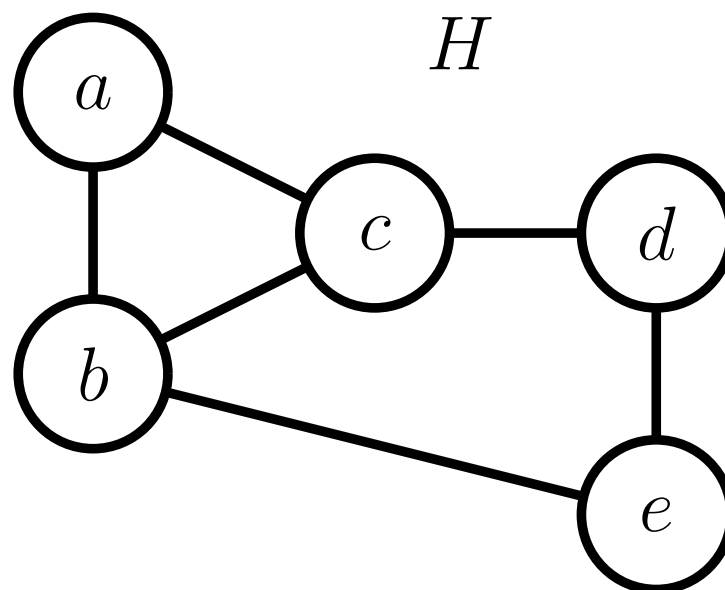
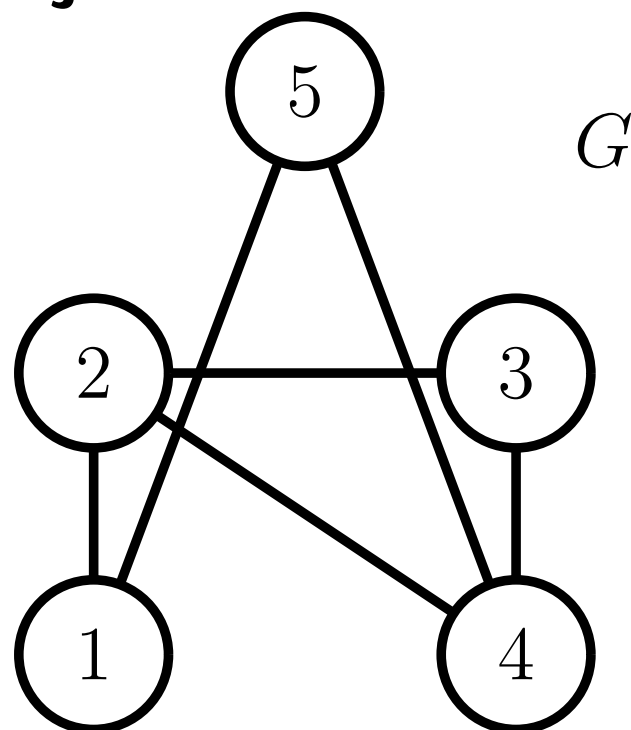
Sean $G = (V, E)$ y $H = (V', E')$ dos grafos simples.

Diremos que G y H son *isomorfos* ($G \cong H$) si existe una biyección $f: V \rightarrow V'$ tal que, para todo $u, v \in V$,

$$uv \in E \iff f(u)f(v) \in E'.$$

En ese caso, f es un *isomorfismo* entre G y H .

• Ej:



$$\begin{aligned} f(1) &= e, \\ f(2) &= b, \\ f(3) &= a, \\ f(4) &= c, \\ f(5) &= d. \end{aligned}$$

Dos descripciones del cubo

- **Isomorfismo:** misma estructura.

Dos descripciones del cubo

- **Isomorfismo:** misma estructura.
- Sea $r \geq 1$ un entero. Consideremos los siguientes grafos:

Dos descripciones del cubo

- **Isomorfismo:** misma estructura.
- Sea $r \geq 1$ un entero. Consideremos los siguientes grafos:
 - Q_r : strings binarios de largo r ; dos son adyacentes si difieren en exactamente una coordenada.

Dos descripciones del cubo

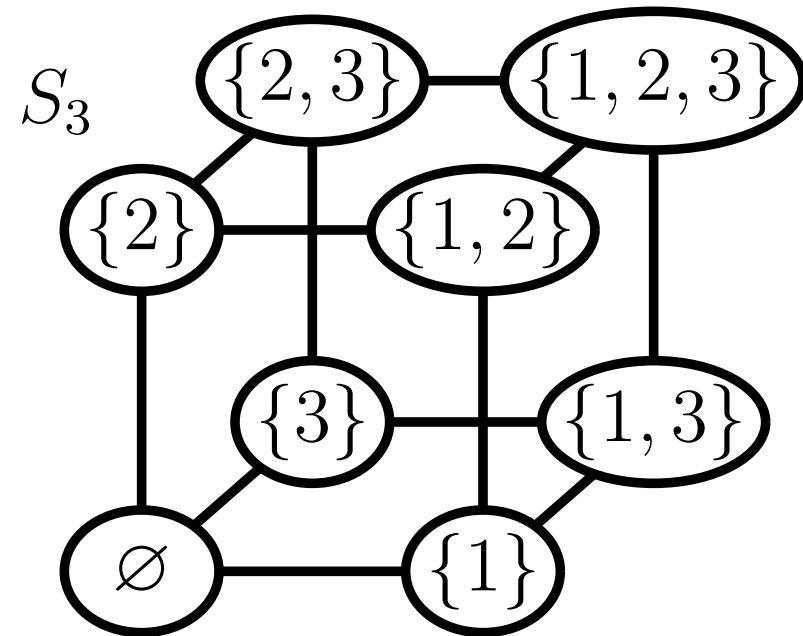
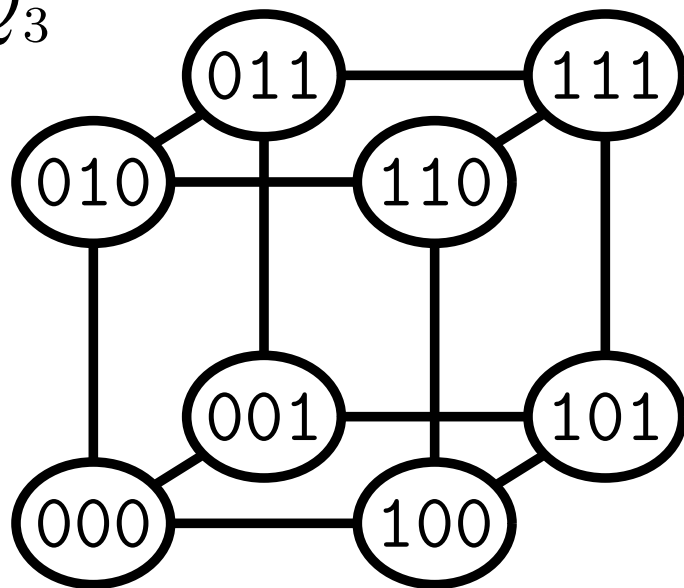
- **Isomorfismo:** misma estructura.
- Sea $r \geq 1$ un entero. Consideremos los siguientes grafos:
 - Q_r : strings binarios de largo r ; dos son adyacentes si difieren en exactamente una coordenada.
 - S_r : subconjuntos de $\{1, \dots, r\}$; dos son adyacentes si uno se obtiene del otro agregando un elemento.

Dos descripciones del cubo

- **Isomorfismo:** misma estructura.
- Sea $r \geq 1$ un entero. Consideremos los siguientes grafos:
 - Q_r : strings binarios de largo r ; dos son adyacentes si difieren en exactamente una coordenada.
 - S_r : subconjuntos de $\{1, \dots, r\}$; dos son adyacentes si uno se obtiene del otro agregando un elemento.

- **Ej:**

Q_3

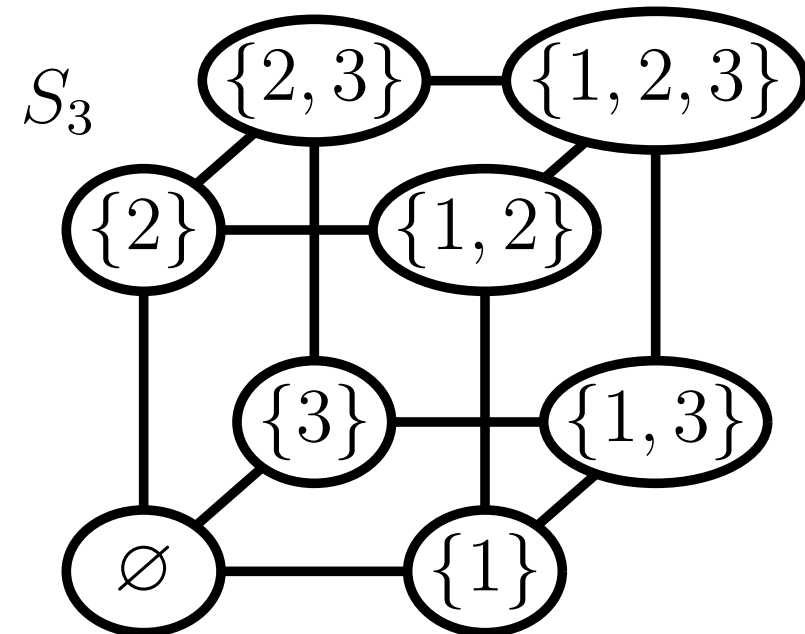
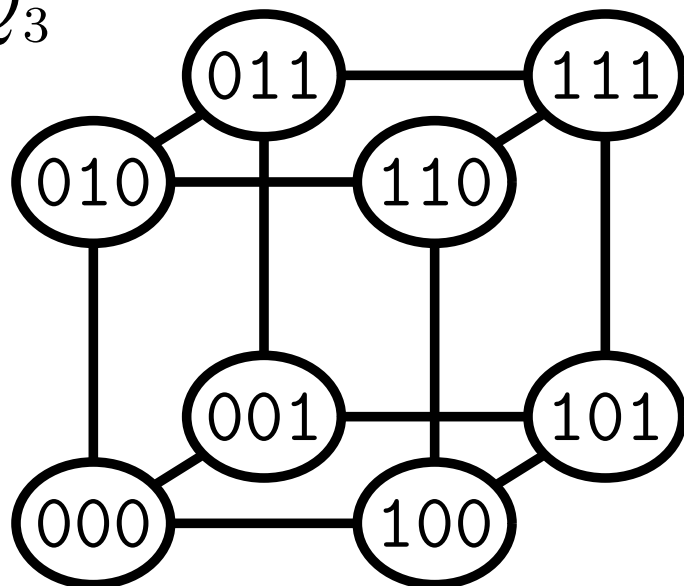


Dos descripciones del cubo

- **Isomorfismo:** misma estructura.
- Sea $r \geq 1$ un entero. Consideremos los siguientes grafos:
 - Q_r : strings binarios de largo r ; dos son adyacentes si difieren en exactamente una coordenada.
 - S_r : subconjuntos de $\{1, \dots, r\}$; dos son adyacentes si uno se obtiene del otro agregando un elemento.

- **Ej:**

Q_3



- **Ej:**

Invariantes

- Nos interesan propiedades de la **estructura**, no las etiquetas.

Invariantes

- Nos interesan propiedades de la **estructura**, no las etiquetas.

Def.

Consideremos una propiedad de grafos.

Invariantes

- Nos interesan propiedades de la **estructura**, no las etiquetas.

Def.

Consideremos una propiedad de grafos.

La propiedad es un *invariante (bajo isomorfismo)* si asigna el mismo valor a grafos isomorfos.

Invariantes

- Nos interesan propiedades de la **estructura**, no las etiquetas.

Def.

Consideremos una propiedad de grafos.

La propiedad es un *invariante (bajo isomorfismo)* si asigna el mismo valor a grafos isomorfos.

- Ej:

Invariantes

- Nos interesan propiedades de la **estructura**, no las etiquetas.

Def.

Consideremos una propiedad de grafos.

La propiedad es un *invariante (bajo isomorfismo)* si asigna el mismo valor a grafos isomorfos.

- Ej:

Prop.

El número de vértices, el número de aristas y el número de vértices de grado d son invariantes.

Invariantes

- Nos interesan propiedades de la **estructura**, no las etiquetas.

Def.

Consideremos una propiedad de grafos.

La propiedad es un *invariante (bajo isomorfismo)* si asigna el mismo valor a grafos isomorfos.

- Ej:

Prop.

El número de vértices, el número de aristas y el número de vértices de grado d son invariantes.

- Dem:

Invariantes

- Nos interesan propiedades de la **estructura**, no las etiquetas.

Def.

Consideremos una propiedad de grafos.

La propiedad es un *invariante (bajo isomorfismo)* si asigna el mismo valor a grafos isomorfos.

- Ej:

Prop.

El número de vértices, el número de aristas y el número de vértices de grado d son invariantes.

- Dem:

- Nos permite razonar cuando dos grafos **no** son isomorfos.

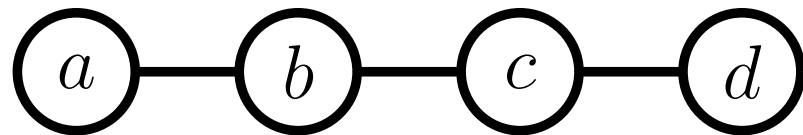
Invariantes

- Ej:

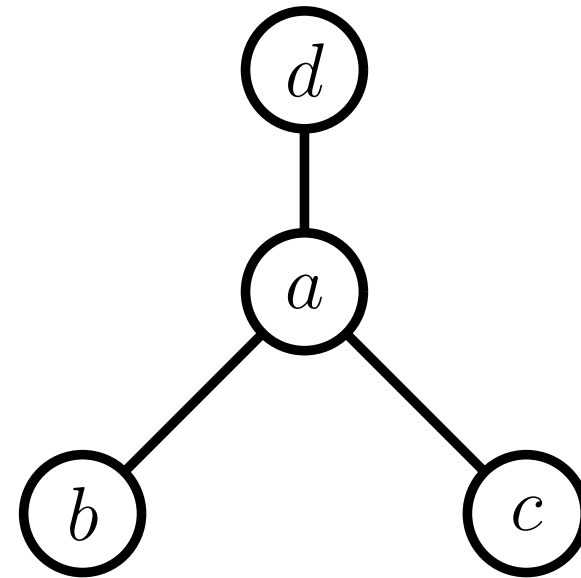
Invariantes

- Ej:

- ¿Son isomorfos?



G

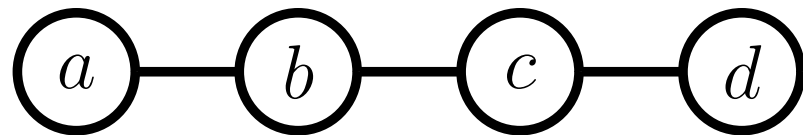


H

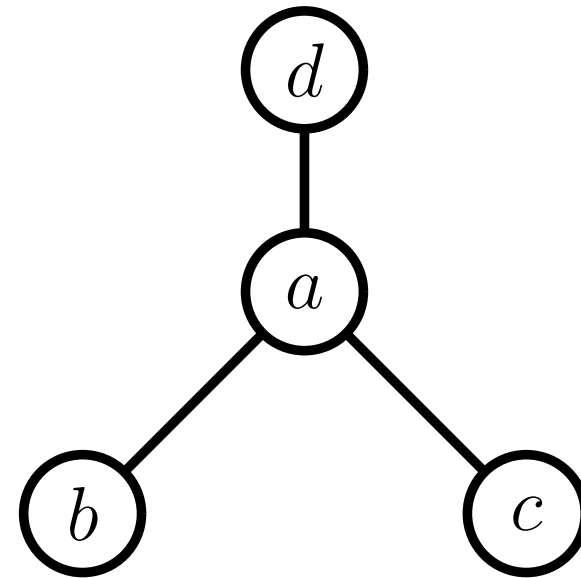
Invariantes

- Ej:

- ¿Son isomorfos?



G



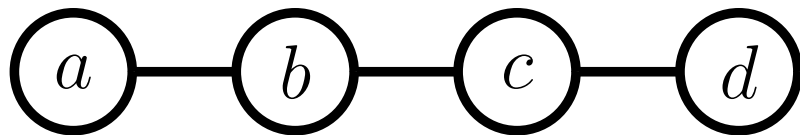
H

- Ej:

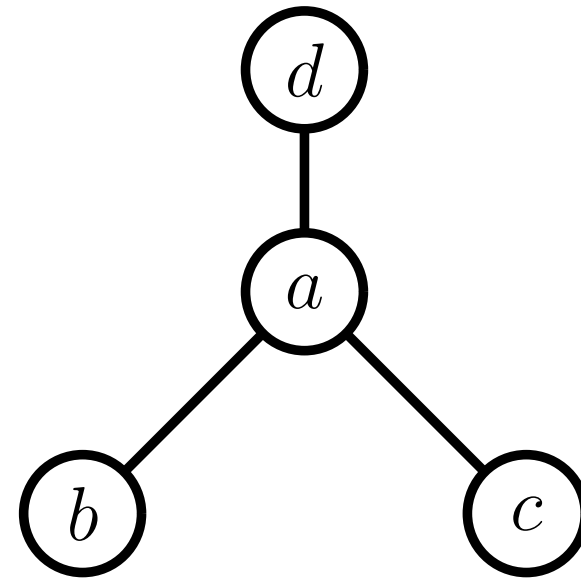
Invariantes

- Ej:

- ¿Son isomorfos?



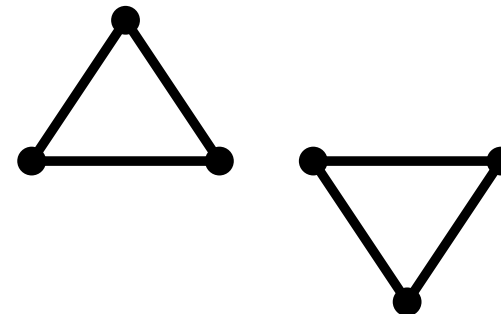
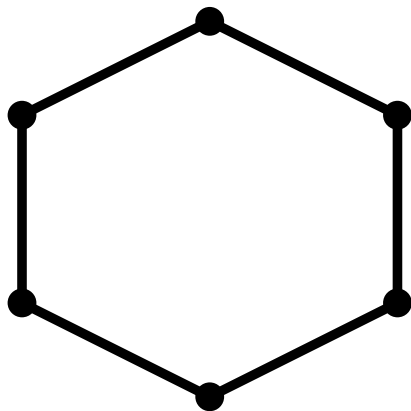
G



H

- Ej:

- ¿Son isomorfos?



Copias de subgrafos

Def.

Sean F y H dos grafos simples y sea $k \geq 0$ un entero.

Copias de subgrafos

Def.

Sean F y H dos grafos simples y sea $k \geq 0$ un entero.

Una *copia* de F en H es un subgrafo de H isomorfo a F .

Copias de subgrafos

Def.

Sean F y H dos grafos simples y sea $k \geq 0$ un entero.

Una *copia* de F en H es un subgrafo de H isomorfo a F .

Dos copias son distintas si difieren en su conjunto de vértices o en su conjunto de aristas.

Copias de subgrafos

Def.

Sean F y H dos grafos simples y sea $k \geq 0$ un entero.

Una *copia* de F en H es un subgrafo de H isomorfo a F .

Dos copias son distintas si difieren en su conjunto de vértices o en su conjunto de aristas.

Decir que H contiene k copias de F significará que contiene al menos k copias distintas.

Copias de subgrafos

Def.

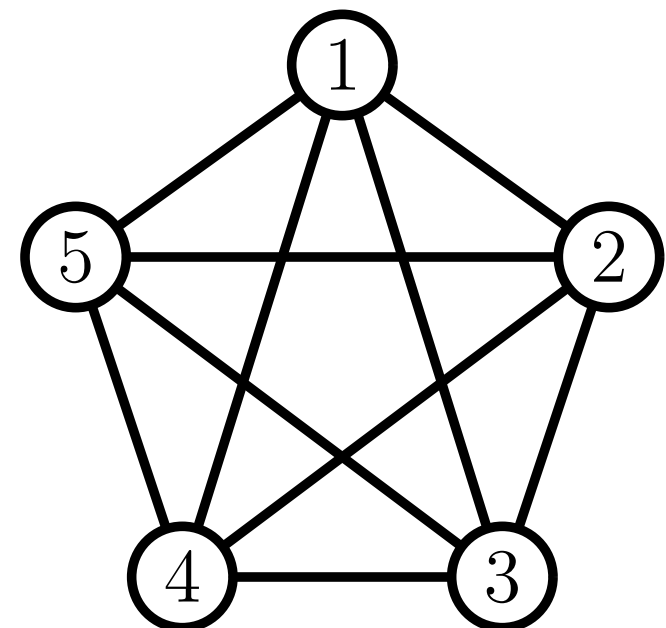
Sean F y H dos grafos simples y sea $k \geq 0$ un entero.

Una *copia* de F en H es un subgrafo de H isomorfo a F .

Dos copias son distintas si difieren en su conjunto de vértices o en su conjunto de aristas.

Decir que H contiene k copias de F significará que contiene al menos k copias distintas.

- Ej:



Copias de subgrafos

Def.

Sean F y H dos grafos simples y sea $k \geq 0$ un entero.

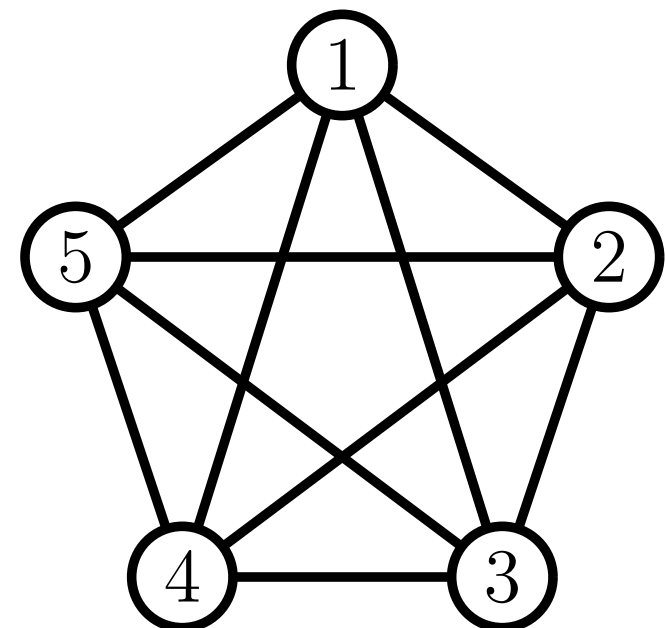
Una *copia* de F en H es un subgrafo de H isomorfo a F .

Dos copias son distintas si difieren en su conjunto de vértices o en su conjunto de aristas.

Decir que H contiene k copias de F significará que contiene al menos k copias distintas.

• Ej:

- ¿Cuántas copias de C_3 contiene K_5 ?



Invariantes y copias

Prop.

Fijemos un grafo simple H y enteros $k \geq 0$ y $\ell \geq 1$.

Invariantes y copias

Prop.

Fijemos un grafo simple H y enteros $k \geq 0$ y $\ell \geq 1$.

Las siguientes propiedades de grafos simples son invariantes:

Invariantes y copias

Prop.

Fijemos un grafo simple H y enteros $k \geq 0$ y $\ell \geq 1$.

Las siguientes propiedades de grafos simples son invariantes:

- Contener una copia de K_ℓ .

Invariantes y copias

Prop.

Fijemos un grafo simple H y enteros $k \geq 0$ y $\ell \geq 1$.

Las siguientes propiedades de grafos simples son invariantes:

- Contener una copia de K_ℓ .
- Contener una copia de P_ℓ .

Invariantes y copias

Prop.

Fijemos un grafo simple H y enteros $k \geq 0$ y $\ell \geq 1$.

Las siguientes propiedades de grafos simples son invariantes:

- Contener una copia de K_ℓ .
- Contener una copia de P_ℓ .
- Contener una copia de C_ℓ , para $\ell \geq 3$.

Invariantes y copias

Prop.

Fijemos un grafo simple H y enteros $k \geq 0$ y $\ell \geq 1$.

Las siguientes propiedades de grafos simples son invariantes:

- Contener una copia de K_ℓ .
- Contener una copia de P_ℓ .
- Contener una copia de C_ℓ , para $\ell \geq 3$.
- Contener una copia de H .

Invariantes y copias

Prop.

Fijemos un grafo simple H y enteros $k \geq 0$ y $\ell \geq 1$.

Las siguientes propiedades de grafos simples son invariantes:

- Contener una copia de K_ℓ .
- Contener una copia de P_ℓ .
- Contener una copia de C_ℓ , para $\ell \geq 3$.
- Contener una copia de H .
- Contener al menos k copias distintas de H .

Invariantes y copias

Prop.

Fijemos un grafo simple H y enteros $k \geq 0$ y $\ell \geq 1$.

Las siguientes propiedades de grafos simples son invariantes:

- Contener una copia de K_ℓ .
- Contener una copia de P_ℓ .
- Contener una copia de C_ℓ , para $\ell \geq 3$.
- Contener una copia de H .
- Contener al menos k copias distintas de H .

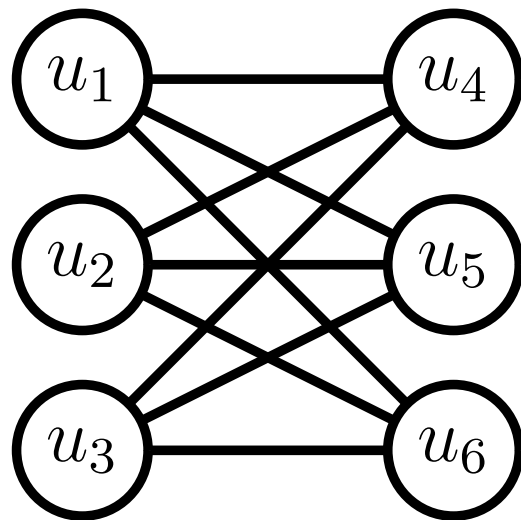
• **Dem:**

Invariantes

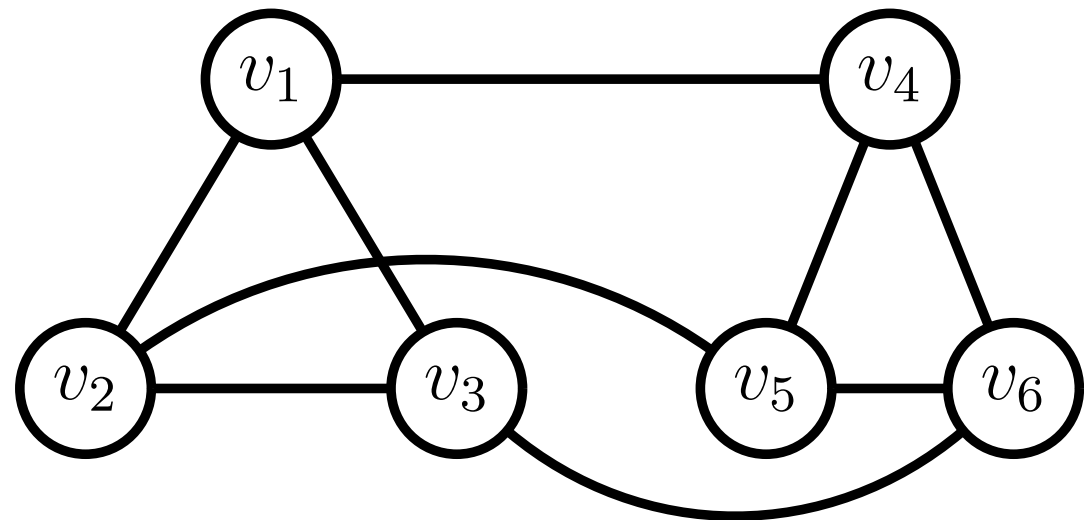
- Ej:

Invariantes

- Ej:
 - ¿Son isomorfos?



G



H

Dudas/Ejercicios

Wrap-up

- **Hoy:**

Wrap-up

- **Hoy:**

- Representaciones: matrices de ady. e incidencia y listas de adyacencia.

Wrap-up

- **Hoy:**

- Representaciones: matrices de ady. e incidencia y listas de adyacencia.
- Isomorfismos

Wrap-up

- **Hoy:**

- Representaciones: matrices de ady. e incidencia y listas de adyacencia.
- Isomorfismos
- Invariantes

Wrap-up

- **Hoy:**

- Representaciones: matrices de ady. e incidencia y listas de adyacencia.
- Isomorfismos
- Invariantes

- **Próx. clase:**

Wrap-up

- **Hoy:**

- Representaciones: matrices de ady. e incidencia y listas de adyacencia.
- Isomorfismos
- Invariantes

- **Próx. clase:**

- Conectividad y árboles.