

# Teoría de grafos

Prof. Enrique Vílchez Quesada

Universidad Nacional de Costa Rica

# Introducción

La teoría de grafos propone un conjunto de definiciones y propiedades que presentan diversas aplicaciones en la computación. Su tarea principal, reside en simplificar muchos tipos de problemas que al ser representados por medio de un grafo, se resuelven con mayor facilidad. Además, la naturaleza misma de ciertas situaciones deriva en la formulación de un grafo, tal es el caso del análisis de redes de transporte o de flujo. Por ejemplo, en un mapa, la gráfica obtenida al relacionar usando un segmento, las distintas regiones involucradas de acuerdo con una forma disponible de transporte de un lugar a otro, construye un grafo. El paquete **VilCretas** integra en este sentido, el comando `CDFAgenteViajeroGrupos` cuya salida es un grafo con datos reales, que relaciona distintos países de una región geográfica en particular, elegida por el usuario.

# Análisis del comando CDFAgenteViajeroGrupos

Al correr CDFAgenteViajeroGrupos [] y seleccionar a Centroamérica, se despliega lo mostrado en la figura 1. En el “combo”, donde aparece CentralAmerica, se facilita la selección de otros grupos regionales de países.



**Descargue un archivo**

<https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-94.zip>

# Análisis del comando CDFAgenteViajeroGrupos

La animación también se encuentra disponible como un documento con un formato computable a través de la siguiente descarga.



**Descargue un archivo**

```
https://www.esconf.una.ac.cr/discretas/Archivos/CDFs/  
Circuito\_de\_Hamilton\_Paises.cdf.zip
```

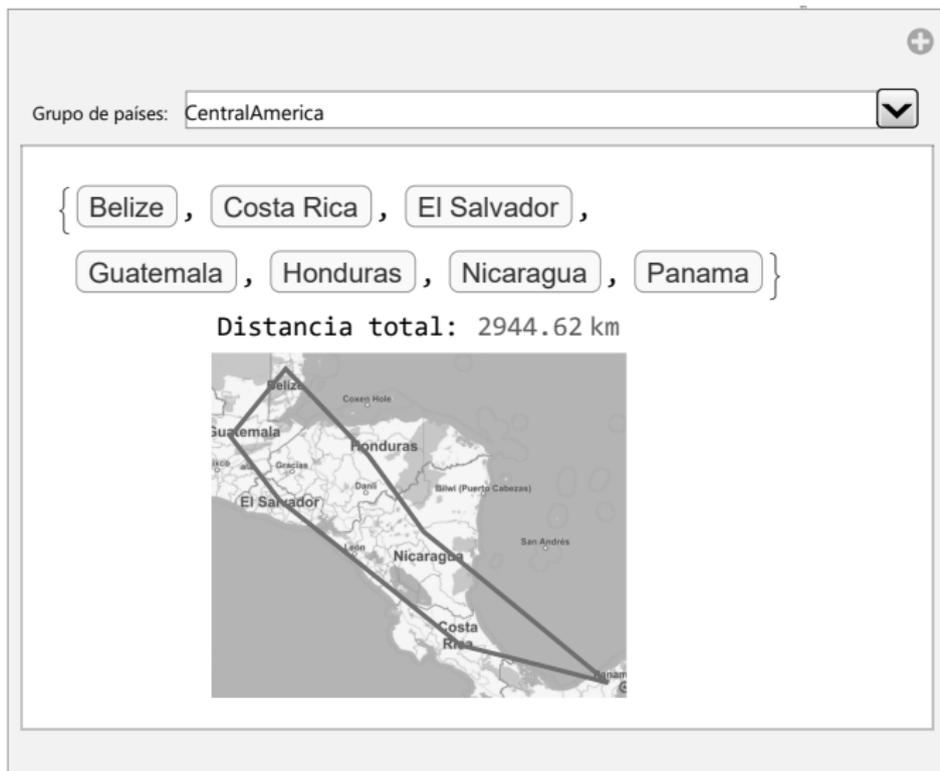


Figura: Salida de CDFAgenteViajeroGrupos para Centroamérica

## Ejemplo adicional

Otro ejemplo de un grafo, se aprecia cuando se piensa en la manera como están interconectadas una serie de computadoras en una *intranet*. Al tomar una curva o arista que las une, tal figura, genera gráficamente un grafo. La estructura de las tuberías en un sistema de acueductos concuerda con un ejemplo adicional en esta dirección.

Cada una de estas representaciones, como veremos en la definición que prosigue, constituye un grafo. Los casos citados proporcionan pequeños ejemplos que sugieren las múltiples aplicaciones que tienen los grafos en muchos campos de estudio, algunas de las cuales, serán abordadas en el presente capítulo.

# Definición 5.1

## Definition (5.1)

Sean  $V$  y  $E$  dos conjuntos no vacíos.  $V$  un conjunto de elementos llamados nodos o vértices y  $E$  un conjunto de aristas o lados. Un grafo  $G = (V, E)$  es una relación binaria  $R$  sobre el conjunto  $V$ , tal que un nodo “ $a$ ” está relacionado con otro vértice “ $b$ ”, si existe una arista  $e \in E$  cuyos extremos son “ $a$ ” y “ $b$ ”. Si las aristas de  $G$  son dirigidas, es decir, tienen un orden en la dirección que relaciona los vértices de  $V$ , entonces a  $G$  se le llama grafo dirigido o digrafo, en caso contrario, se dice que  $G$  es un grafo no dirigido. Si  $G$  es dirigido y  $aRb$  al lado “ $e$ ” de  $G$  que relaciona “ $a$ ” y “ $b$ ”, se le denota  $e = (a, b)$ . Si  $G$  es un grafo no dirigido y  $aRb$ , la arista “ $e$ ” que los une, se representa como  $e = [a, b] = [b, a]$ . Dos vértices relacionados por  $R$  se llaman adyacentes.

## Continuación de la definición 5.1

Se dice además, que una arista “ $e$ ” incide o es incidente sobre dos nodos “ $a$ ” y “ $b$ ”, si “ $e$ ” los conecta entre sí. Una arista “ $e$ ” que une a un nodo “ $a$ ” consigo mismo, se le denomina lazo o ciclo de  $G$ . Se dice que dos nodos “ $a$ ” y “ $b$ ” poseen lados múltiples o paralelos en un grafo  $G$ , si existe más de una arista entre “ $a$ ” y “ $b$ ”. Un grafo  $G$  sin lazos y sin lados múltiples recibe el nombre de grafo simple.

# Comentario sobre la definición 1

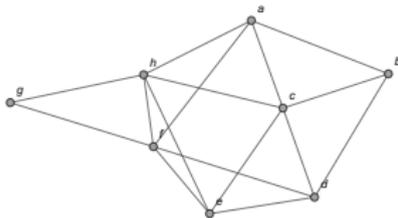
Un grafo de acuerdo con la definición 1 es una relación binaria  $R$  sobre un conjunto de vértices  $V$  donde dos nodos están relacionados si existe una arista  $e$  que los une,  $e$  proveniente de un conjunto de lados  $E$ . En general, todo grafo es una relación binaria que determina en el plano una figura donde los nodos se dibujan a través de puntos primitivos y su relación, por medio de una serie de curvas o aristas que los unen. También, cabe destacar en la definición 1, los conceptos de adyacencia e incidencia. Ambos resultan ser equivalentes, sin embargo, la adyacencia se aplica a los vértices en un grafo y la incidencia a sus aristas. Así, se dice que dos nodos  $a$  y  $b$  son adyacentes si existe un lado  $e$  del grafo con extremos  $a$  y  $b$ , y recíprocamente, al ocurrir esta adyacencia, la arista  $e$  es incidente sobre los nodos  $a$  y  $b$ . Más adelante, se vincularán estos términos con dos tipos de arreglos bidimensionales que permiten representar cualquier grafo: la matriz de adyacencia y la matriz de incidencia.

La figura 2 muestra una serie de ejemplos de grafos. El grafo 2a corresponde a un grafo simple, es decir, sin lazos y sin aristas múltiples. El grafo 2b no es simple al tener los nodos  $e$  y  $j$  varias aristas que los unen (hay lados paralelos). El grafo 2c es un digrafo simple. Y finalmente, el grafo 2d corresponde a un grafo dirigido con un lazo en el vértice  $a$  y dos aristas del vértice  $f$  al nodo  $h$ , por lo que se considera un digrafo no simple. En estos grafos las relaciones binarias que los definen son:

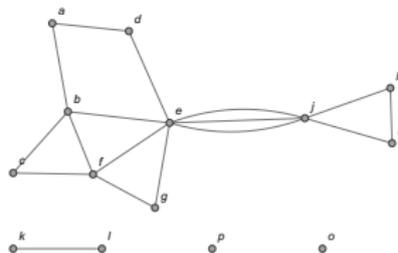
- Grafo 2a:  $R = \{(a, b), (a, c), (a, f), (a, h), (b, a), (b, c), (b, d), (c, a), (c, b), (c, d), (c, e), (c, h), (d, b), (d, c), (d, e), (d, f), (e, c), (e, d), (e, f), (e, h), (f, a), (f, d), (f, e), (f, g), (f, h), (g, f), (g, h), (h, a), (h, c), (h, e), (h, f), (h, g)\}$  sobre  $V = \{a, b, c, d, e, f, g, h\}$ .  $R$  es una relación simétrica y siempre ocurre esto, cuando el grafo es no dirigido.

- Grafo 2b:  $R = \{(a, b), (a, d), (b, a), (b, c), (b, e), (b, f), (c, b), (c, f), (d, a), (d, e), (e, b), (e, d), (e, f), (e, g), (e, j), (e, j), (e, j), (f, e), (f, b), (f, c), (f, g), (g, e), (g, f), (h, i), (h, j), (i, h), (i, j), (j, e), (j, e), (j, e), (j, h), (j, i), (k, l), (l, k)\}$  sobre  $V = \{a, b, c, d, e, f, g, h, i, j, k, l, o, p\}$ .  $R$  también es simétrica al representar un grafo no dirigido. En teoría de conjuntos, si un conjunto tiene elementos repetidos como sucede en este caso, esos elementos se pueden eliminar, pese a ello, aquí se conservan en  $R$  para indicar la existencia de aristas paralelas.
- Grafo 2c:  $R = \{(a, d), (a, e), (a, f), (a, g), (a, h), (a, i), (b, e), (b, f), (b, i), (c, d), (c, e), (d, e), (d, g), (e, f), (e, i), (g, e), (g, h), (f, i)\}$  sobre  $V = \{a, b, c, d, e, f, g, h, i\}$ .  $R$  no es una relación binaria simétrica pues el grafo es dirigido.

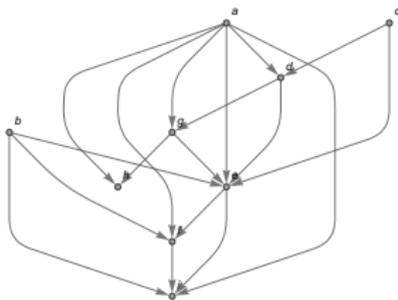
- Grafo 2d:  $R = \{(a, a), (a, b), (a, d), (a, e), (a, f), (a, j), (b, c), (b, d), (b, i), (b, j), (c, d), (c, g), (c, f), (c, h), (d, a), (d, e), (d, f), (e, j), (f, g), (f, h), (f, h), (h, i)\}$  sobre  $V = \{a, b, c, d, e, f, g, h, i, j\}$ . No se ha eliminado de  $R$  el elemento repetido  $(f, h)$  para indicar la existencia de un lado múltiple. Del mismo modo,  $R$  no es simétrica al representar un digrafo.



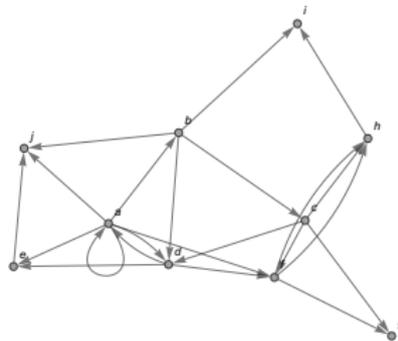
(a) Grafo simple



(b) Grafo con lados múltiples



(c) Digrafo simple



(d) Digrafo no simple

Figura: Ejemplos de grafos

# Uso del comando Grafo

La librería **VilCretas** cuenta con una colección de comandos fundamentales para resolver distintos tipos de problemas asociados con la teoría de grafos, además de considerar, los algoritmos clásicos sobre este importante tema.

Para iniciar es esencial explicar el funcionamiento de la instrucción `Grafo`. Un grafo se puede crear en *Wolfram Mathematica* por medio de la sentencia `Grafo` del paquete **VilCretas**. El comando recibe como argumento el conjunto aristas del grafo a construir.

# Uso del comando Grafo

Por ejemplo, los grafos mostrados en la figura 2 se crean en *Mathematica* así:

**In[ ] :=**

```
Grafo[{{a, b}, {a, c}, {a, f}, {a, h}, {b, c}, {b, d},
{c, d}, {c, e}, {c, h}, {d, e}, {d, f}, {e, f}, {e, h},
{f, g}, {f, h}, {g, h}}]
```

```
Grafo[{{a, b}, {a, d}, {b, c}, {b, e}, {b, f}, {c, f},
{e, d}, {e, f}, {e, g}, {e, j}, {e, j}, {e, j}, {f, g},
{h, i}, {i, j}, {j, h}, {l, k}}, vertices -> {o, p}]
```

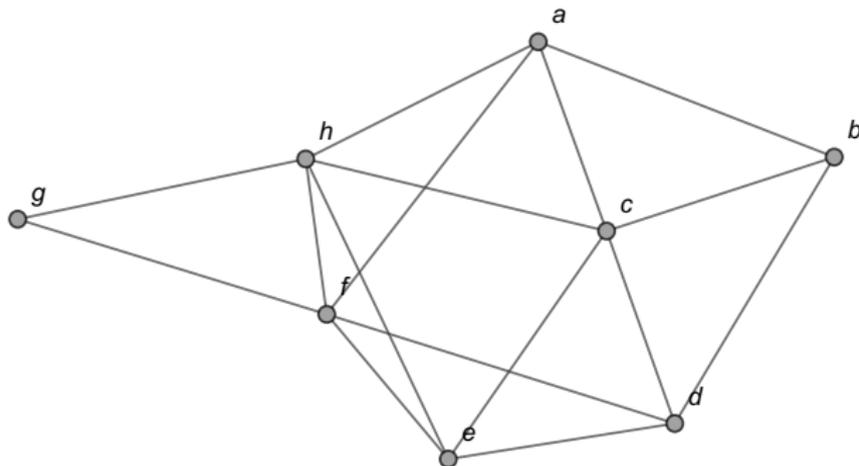
# Uso del comando Grafo

```
Grafo[{{a, d}, {a, e}, {a, f}, {a, g}, {a, h}, {a, i},  
{b, e}, {b, f}, {b, i}, {c, d}, {c, e}, {d, e}, {d, g},  
{e, f}, {e, i}, {g, e}, {g, h}, {f, i}}, dirigido -> True]  
Grafo[{{a, a}, {a, b}, {a, d}, {a, e}, {a, f}, {a, j},  
{b, c}, {b, d}, {b, i}, {b, j}, {c, d}, {c, g}, {c, f},  
{c, h}, {d, a}, {d, e}, {d, f}, {e, j}, {f, g}, {f, h},  
{f, h}, {h, i}}, dirigido -> True]
```

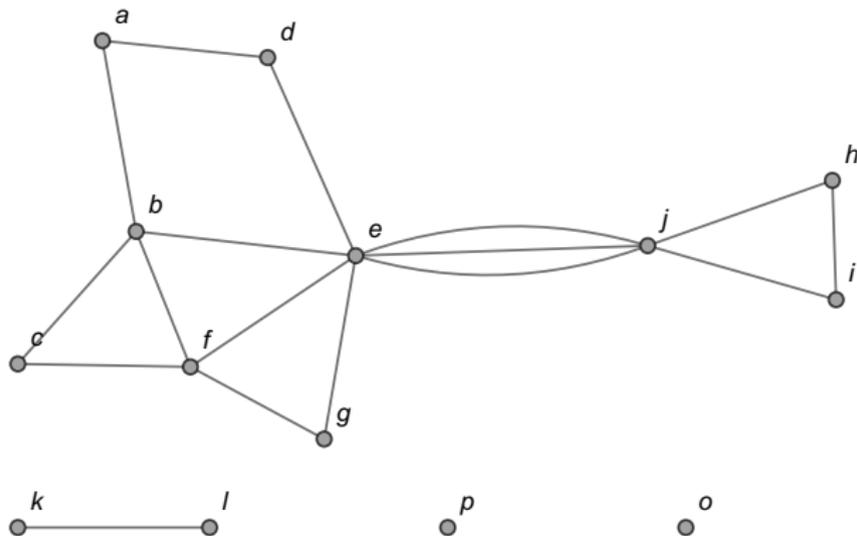
# Uso del comando Grafo

Se obtienen las siguientes salidas:

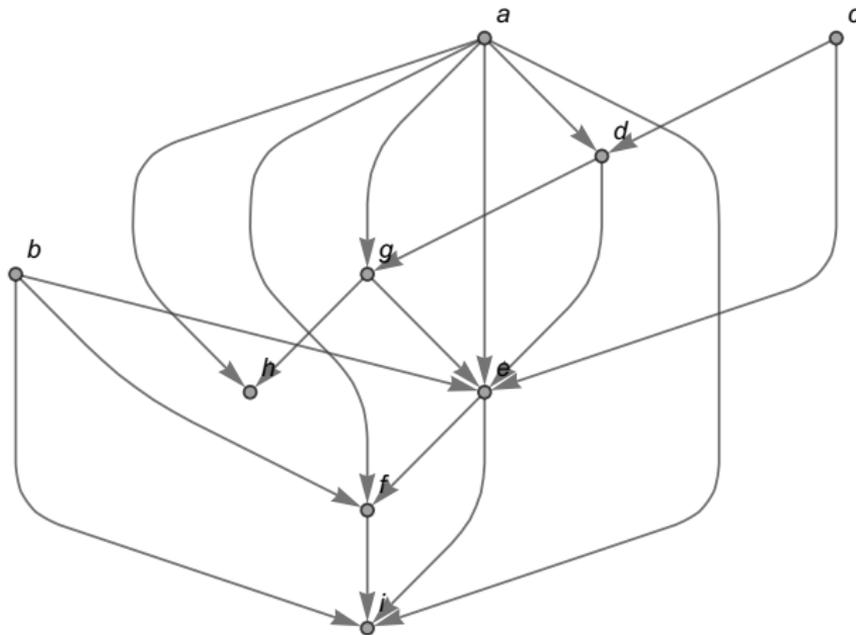
**Out[ ] =**



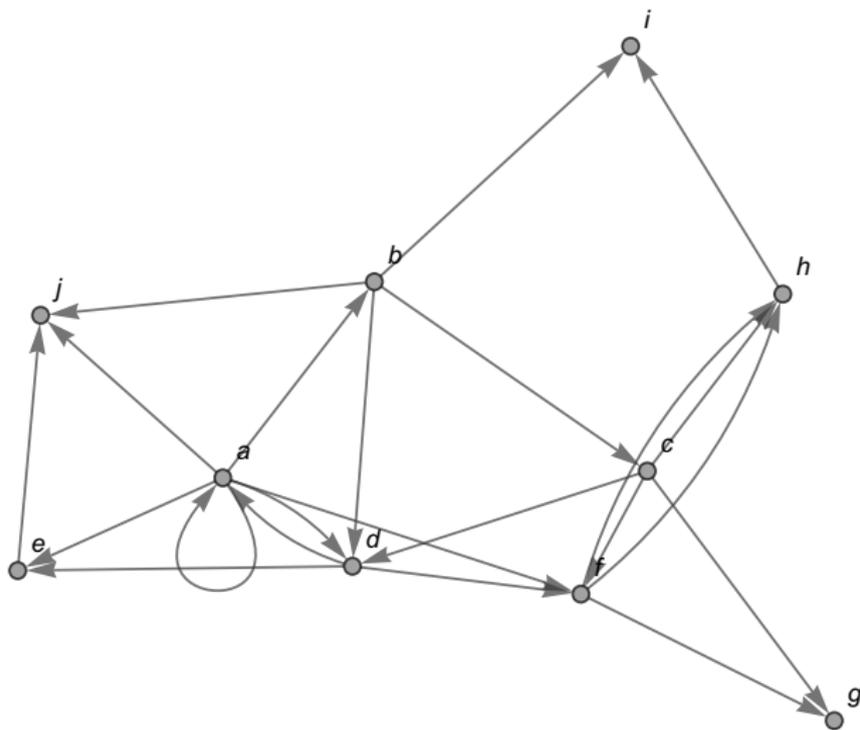
# Uso del comando Grafo



# Uso del comando Grafo



# Uso del comando Grafo



# Uso del comando Grafo

## Nota

La opción `vertices -> {o, p}` en el segundo grafo, facilita añadir los puntos aislados `o` y `p`. Se llaman aislados pues ninguna arista los incide. Además, `dirigido -> True` especifica en los grafos tres y cuatro su condición de digrafos, es decir, los lados transcritos en el conjunto de aristas tienen un orden. Esto quiere decir, que una arista de la forma  $\{x, y\}$  en el conjunto de lados, denota la presencia de una flecha dirigida iniciando en  $x$  y finalizando en  $y$ .

# Uso del comando Grafo

La instrucción Grafo presenta otras opciones. Entre ellas, `shape -> True` y `dimensions3d -> True` para colocar los vértices del grafo dentro de un círculo o una esfera (en  $3d$ ) y crear el grafo en tercera dimensión, respectivamente. Un grafo en tercera dimensión se caracteriza por poderse rotar colocando el cursor del *mouse* sobre la figura y ejerciendo un arrastre encima ella.

# Uso del comando Grafo

A continuación, se comparte la salida de Grafo utilizando estas opciones sobre el grafo 2a:

**In[ ] :=**

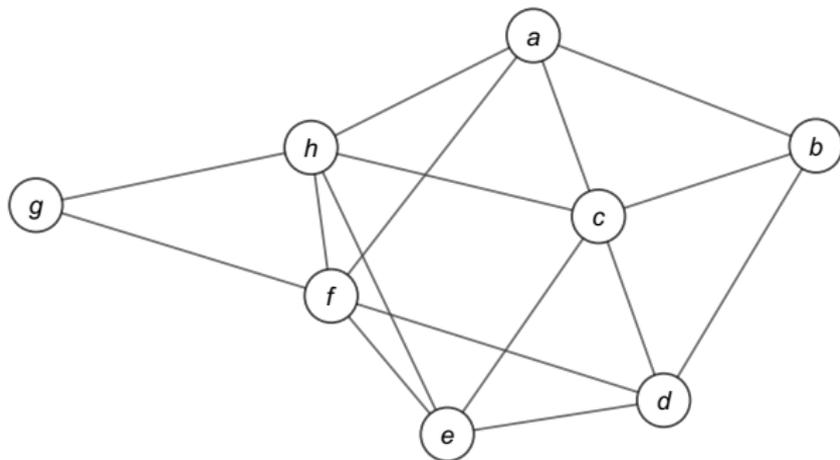
```
Grafo[{{a, b}, {a, c}, {a, f}, {a, h}, {b, c}, {b, d},  
{c, d}, {c, e}, {c, h}, {d, e}, {d, f}, {e, f}, {e, h},  
{f, g}, {f, h}, {g, h}}, shape -> True]
```

```
Grafo[{{a, b}, {a, c}, {a, f}, {a, h}, {b, c}, {b, d},  
{c, d}, {c, e}, {c, h}, {d, e}, {d, f}, {e, f}, {e, h},  
{f, g}, {f, h}, {g, h}}, dimensions3d -> True]
```

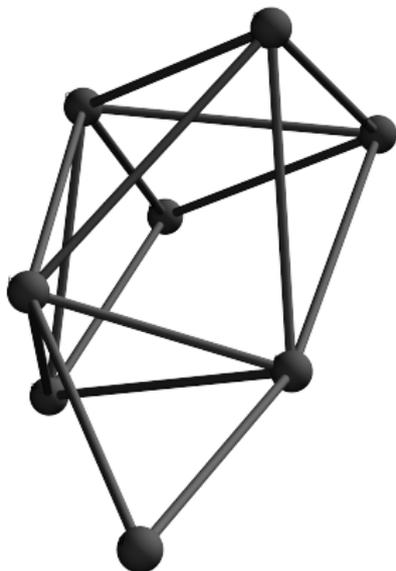
# Uso del comando Grafo

Se obtienen la siguientes salidas:

**Out[ ] =**



# Uso del comando Grafo



# Atributos adicionales del comando Grafo

Otros atributos de utilidad residen en las opciones pesos y mostrarpesos del comando Grafo. La propiedad pesos permite asignar pesos o valores a las aristas de un grafo y la opción mostrarpesos como su nombre lo indica, muestra los pesos sobre los lados del grafo. Un grafo con pesos asignados en sus aristas recibe el nombre de grafo ponderado. Veamos el concepto.

## Definición 5.2

### Definition (5.2)

Si los lados de un grafo tienen asociado un valor, usualmente numérico, al grafo se le llama grafo ponderado.

## Atributos adicionales del comando Grafo

Como ya se señaló, pesos y mostrarpesos facultan la construcción de un grafo ponderado o con pesos, usando la sentencia Grafo. Por ejemplo, el siguiente código reconstruye el grafo 2a, donde a las aristas se les está asignando un peso pseudoaleatorio correspondiente a un número entero en el rango de 1 a 20 :

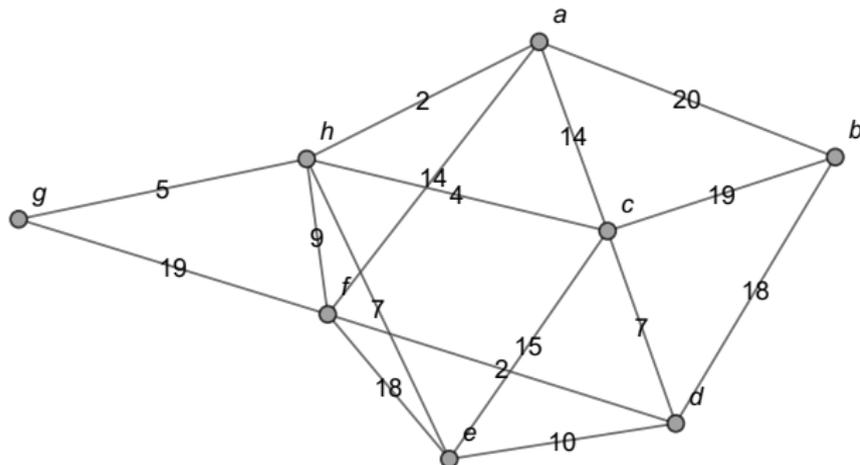
```
In[ ] :=
```

```
Grafo[{{a, b}, {a, c}, {a, f}, {a, h}, {b, c}, {b, d},  
{c, d}, {c, e}, {c, h}, {d, e}, {d, f}, {e, f}, {e, h},  
{f, g}, {f, h}, {g, h}}, pesos -> RandomInteger[{1, 20},  
16], mostrarpesos -> True]
```

# Atributos adicionales del comando Grafo

Se obtiene la siguiente salida:

**Out[ ] =**



## Atributos adicionales del comando Grafo

Al ser la salida pseudoaleatoria, cada vez que se ejecuta el `In[ ]`, se asignan pesos distintos a los lados del grafo `2a`. Si se omite `mostrarpesos -> True` se crea el grafo con pesos en *Mathematica* pero no se muestran las ponderaciones sobre sus aristas.

Otros comandos interesantes que retornan la lista de vertices, la cantidad de nodos, la lista de aristas y el número de lados de un grafo, corresponden a: `ListaVertices`, `CantNodos`, `ListaAristas` y `CantAristas`, respectivamente.

# Atributos adicionales del comando Grafo

Si para el grafo 2a se corren estas sentencias en *Wolfram*, se obtiene lo que sigue:

**In[ ] :=**

```
grafo = Grafo[{{a, b}, {a, c}, {a, f}, {a, h}, {b, c},  
{b, d}, {c, d}, {c, e}, {c, h}, {d, e}, {d, f}, {e, f},  
{e, h}, {f, g}, {f, h}, {g, h}}];
```

```
ListaVertices[grafo]
```

```
CantNodos[grafo]
```

```
ListaAristas[grafo]
```

```
CantAristas[grafo]
```

# Atributos adicionales del comando Grafo

Se obtienen las siguientes salidas:

**Out[ ] =**

{a, b, c, d, e, f, g, h}

8

{{a, b}, {a, c}, {a, f}, {a, h}, {b, c}, {b, d}, {c, d}, {c, e}, {c, h},  
{d, e}, {d, f}, {e, f}, {e, h}, {f, g}, {f, h}, {g, h}}

16

Concluyéndose que este grafo contiene ocho nodos y dieciseis lados.



## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-95.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-95.zip)



## Explicación en video

<https://youtu.be/yeK5r8FQRuE>

## Definición 5.3

### Definition (5.3)

Una ruta, camino o trayectoria entre dos nodos  $a$  y  $b$  de un grafo  $G = (V, E)$  es una sucesión de vértices de  $V$  que conecta el nodo  $a$  con el vértice  $b$ , por medio de las aristas de  $E$  sobre  $G$ , sin repetición de lados. La longitud de una ruta se define como la cantidad de aristas de  $G$  utilizadas para conectar  $a$  con  $b$ . Si el grafo es ponderado esta longitud se calcula como la suma de los pesos de los lados involucrados en la trayectoria. En general, en un grafo sin pesos, si se usan  $n$  nodos en una trayectoria, su longitud corresponde a  $n - 1$ . Un grafo  $G$  se dice que es conexo, cuando entre cada par de nodos cualesquiera siempre existe una ruta que los une. En caso contrario, se dice que el grafo es desconexo.

## Comentario sobre la definición 3

El alumno no debe olvidar en el concepto 3, que en una ruta, camino o trayectoria entre dos vértices de un grafo, nunca se pueden repetir aristas.

# Rutas y caminos

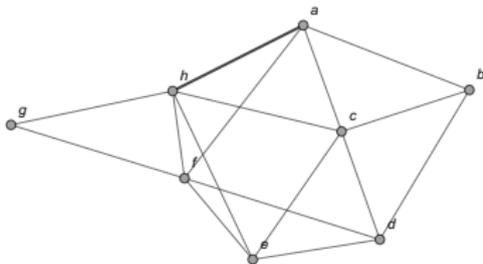
Los grafos **2a**, **2c** y **2d** son ejemplos de grafos conexos y **2b** es un ejemplo de un grafo desconexo.

Además, en la figura **3** se comparten cuatro rutas o caminos distintos del vértice  $a$  al nodo  $h$  sobre el grafo **2a**. Estas trayectorias se denotan mediante una sucesión de vértices que las definen, a saber:

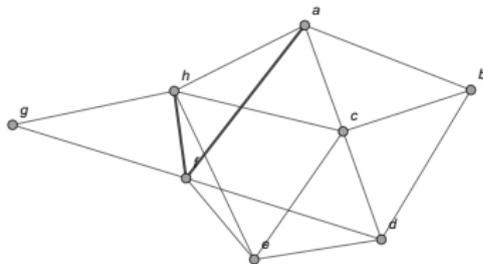
- Ruta **3a**:  $a \rightarrow h$ . La cantidad de nodos presentes es  $n = 2$ , por lo cual, la longitud de la ruta corresponde a  $n - 1 = 2 - 1 = 1$ . Si se recurre al grafo ponderado de la página 29, al tener  $a \rightarrow h$  peso 2, lo que se representa así:  $a \xrightarrow{2} h$ , se infiere que la longitud de la ruta es 2.
- Camino **3b**:  $a \rightarrow f \rightarrow h$ . Aquí,  $n = 3$ , la longitud de la trayectoria responde a la fórmula  $n - 1 = 3 - 1 = 2$ . Si usa el grafo con pesos de la página 29, se observa:  $a \xrightarrow{14} f \xrightarrow{9} h$ , con lo cual la longitud de la ruta sería  $14 + 9 = 23$ .

# Rutas y caminos

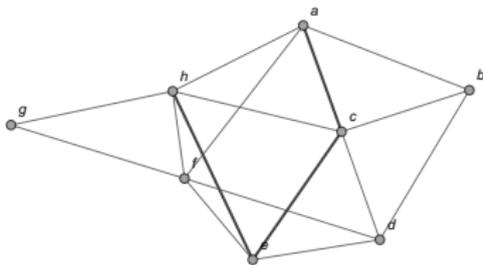
- Trayectoria **3c**:  $a \rightarrow c \rightarrow e \rightarrow h$ . En este camino hay 4 vértices,  $n = 4$ , por lo que la longitud de la ruta es  $n - 1 = 4 - 1 = 3$ . Empleando el grafo ponderado de la página 29, se aprecia:  
 $a \xrightarrow{14} c \xrightarrow{15} e \xrightarrow{7} h$ , de donde se interpreta la longitud del camino como  $14 + 15 + 7 = 36$ .
- Camino **3d**:  $a \rightarrow f \rightarrow d \rightarrow c \rightarrow h$ . Por consiguiente, la longitud de la ruta es  $n - 1 = 5 - 1 = 4$ . Usando el grafo con pesos de la página 29:  $a \xrightarrow{14} f \xrightarrow{2} d \xrightarrow{7} c \xrightarrow{4} h$ , por lo tanto, la longitud de esta trayectoria es  $14 + 2 + 7 + 4 = 27$ .



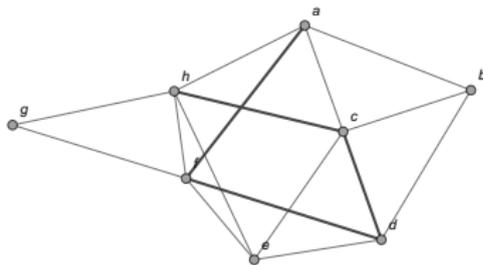
(a) Ruta de longitud uno



(b) Ruta de longitud dos



(c) Ruta de longitud tres



(d) Ruta de longitud cuatro

Figura: Ejemplos de rutas sobre el grafo 2a

# Usos de los comandos GeneraRutas y CantRutas

El paquete **VilCretas** integra las instrucciones GeneraRutas y CantRutas, donde la primera, retorna un máximo de  $n$  trayectorias de diferente longitud entre dos nodos distintos de un grafo pasados como parámetro y la segunda, devuelve sobre un grafo simple, el número total de caminos existentes sin repetición de vértices, entre dos nodos diferentes especificados por el usuario. CantRutas, adicionalmente, posee la opción `rutas -> True` capaz de mostrar la cantidad total de trayectorias y todos los caminos posibles.

# Usos de los comandos GeneraRutas y CantRutas

Al utilizar estos dos comandos sobre el grafo **2a** del vértice  $a$  al nodo  $h$ , se obtiene:

**In[ ]** :=

```
grafo = Grafo[{{a, b}, {a, c}, {a, f}, {a, h}, {b, c},  
{b, d}, {c, d}, {c, e}, {c, h}, {d, e}, {d, f}, {e, f},  
{e, h}, {f, g}, {f, h}, {g, h}}];  
GeneraRutas[grafo, a, h, 5]  
CantRutas[grafo, a, h, rutas -> True]
```

# Usos de los comandos GeneraRutas y CantRutas

Se obtienen la siguientes salidas:

**Out[ ] =**

$$\{ \{ \{a, h\} \}, \{ \{a, f\}, \{f, h\} \}, \{ \{a, c\}, \{c, e\}, \{e, h\} \}, \{ \{a, b\}, \{b, c\}, \{c, h\} \} \}$$

$$\{ 55, \{ \{ \{a, h\} \}, \{ \{a, f\}, \{f, h\} \}, \{ \{a, c\}, \{c, h\} \}, \dots, \{ \{a, b\}, \{b, c\}, \{c, d\}, \{d, e\}, \{e, f\}, \{f, g\}, \{g, h\} \} \} \}$$

# Usos de los comandos GeneraRutas y CantRutas

## Nota

Se aprecia que la instrucción `GeneraRutas` no logró retornar las cinco trayectorias del nodo  $a$  al vértice  $h$ , solicitadas en su argumento 5. La sentencia solo consiguió encontrar cuatro de ellas. También, se aclara al lector que en el **Out[ ]** aparecen puntos suspensivos por la imposibilidad de presentar la salida completa, al devolver `CantRutas` cincuenta y cinco caminos distintos del vértice  $a$  al nodo  $h$ .

# Usos de los comandos GeneraRutas y CantRutas



Descargue un archivo

<https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-96.zip>



Explicación en video

<https://youtu.be/qmPVwFLXmxg>



Explicación en video

<https://youtu.be/TGmx370ssH4>

# Uso del comando PesoRuta

Además, la librería **VilCretas** cuenta con el comando `PesoRuta` encargado de hallar la longitud de un camino ingresado como parámetro. Si el grafo en cuestión es ponderado, `PesoRuta` determina la longitud a través de los pesos asignados a las aristas de la trayectoria, tal y como lo establece la definición 3. Al emplear el camino 3d del grafo 2a en la sentencia `PesoRuta`, se tiene:

`In[ ] :=`

```
grafo = Grafo[{{a, b}, {a, c}, {a, f}, {a, h}, {b, c},  
{b, d}, {c, d}, {c, e}, {c, h}, {d, e}, {d, f}, {e, f},  
{e, h}, {f, g}, {f, h}, {g, h}}];  
L = {{a, f}, {f, d}, {d, c}, {c, h}};  
PesoRuta[grafo, L]
```

# Uso del comando `PesoRuta`

Se obtiene la siguiente salida:

**Out[ ] =**

4

`PesoRuta` presenta la opción `ruta -> True` que dibuja el camino sobre el grafo, además de compartir la longitud de la trayectoria correspondiente.

Al agregar este atributo en la instrucción `PesoRuta` del **In[ ]** anterior, *Mathematica* retorna:

**In[ ] :=**

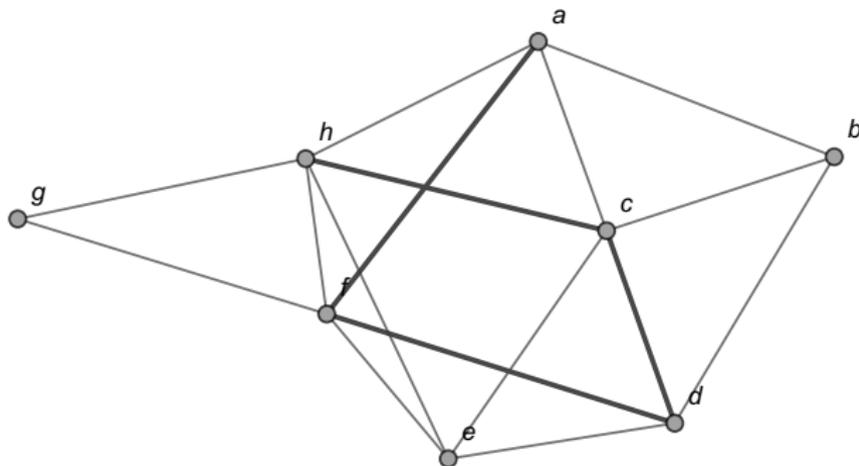
```
PesoRuta[grafo, L, ruta -> True]
```

# Uso del comando `PesoRuta`

Muestra la siguiente salida:

**Out[ ] =**

4



## Nota

Al crear un grafo en *Wolfram Mathematica* con la instrucción `Grafo`, la imagen devuelta por el software es reconocida en sí misma como un grafo por parte del programa. Esto quiere decir, que si se copia esa figura y se pega como argumento de cualquier comando que reciba un grafo como parámetro, la instrucción correrá con toda normalidad.

# Uso del comando Grafo

Por ejemplo, en *Mathematica* el grafo ponderado 29, se puede obtener como sigue:

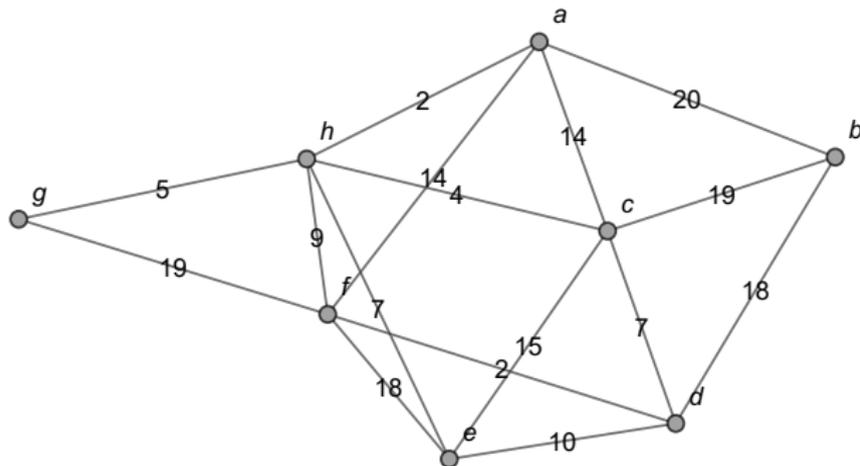
**In[ ] :=**

```
Grafo[{{a, b}, {a, c}, {a, f}, {a, h}, {b, c}, {b, d},  
{c, d}, {c, e}, {c, h}, {d, e}, {d, f}, {e, f}, {e, h},  
{f, g}, {f, h}, {g, h}}, pesos -> {20, 14, 14, 2, 19, 18,  
7, 15, 4, 10, 2, 18, 7, 19, 9, 5}, mostrarpesos -> True]
```

# Uso del comando Grafo

Se obtiene la siguiente salida:

**Out[ ] =**

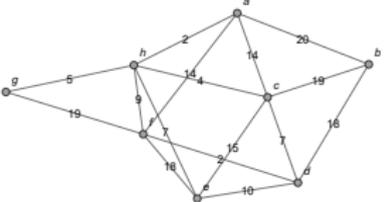


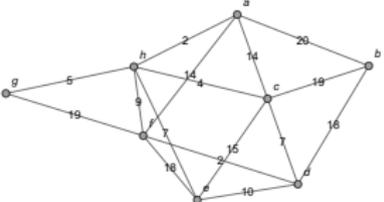
# Uso del comando `PesoRuta`

Si ahora se copia en *Wolfram* esta imagen y se pega dentro de la instrucción `PesoRuta`, se resolverá sin problemas el cálculo de la longitud de  $L = \{\{a, f\}, \{f, d\}, \{d, c\}, \{c, h\}\}$ , en este caso, recurriendo a los pesos de las aristas involucradas. Veamos:

`In[ ] :=`

`L = {{a, f}, {f, d}, {d, c}, {c, h}};`

`PesoRuta[`  `, L]`

`PesoRuta[`  `, L, ruta -> True]`

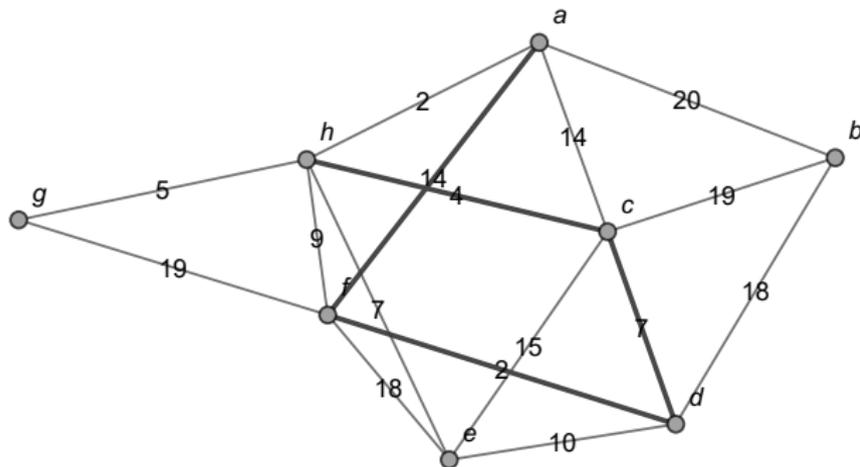
# Uso del comando `PesoRuta`

Se obtienen las siguientes salidas:

**Out[ ] =**

27

27



# Uso del comando PesoRuta



## Descargue un archivo

<https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-97.zip>



## Explicación en video

<https://youtu.be/vJbf61RkEJM>

# Tipos de grafos

Prof. Enrique Vílchez Quesada

Universidad Nacional de Costa Rica

En la literatura existen muchos tipos de grafos. Por lo pronto, se mencionarán tres tipos de grafos de mucha importancia, los grafos: vacío, completo y bipartito completo.

- 1 **Grafo vacío:** cumple con la condición de no contener ninguna arista. Se le representa como  $D_n$ , siendo  $n$  el número de vértices que integra. La instrucción `GrafoVacio` del paquete **VilCretas** facilita su trazo. Por ejemplo, un grafo vacío de orden 8 se construye así:

```
In[ ] :=
```

```
GrafoVacio[8]
```

```
Out[ ] =
```

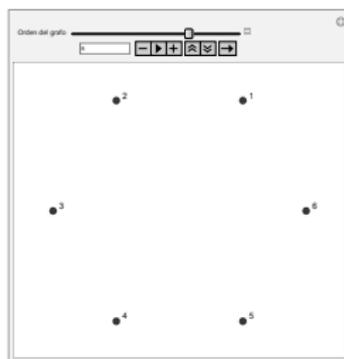


GrafoVacio goza de la opción analizar  $\rightarrow$  True que genera una animación con distintos grafos vacíos comenzando en el orden 1 y hasta un orden  $n$ ,  $n \in \mathbb{N}$ , indicado por el usuario. Por ejemplo:

In[ ] :=

GrafoVacio[8, analizar  $\rightarrow$  True]

Out[ ] =



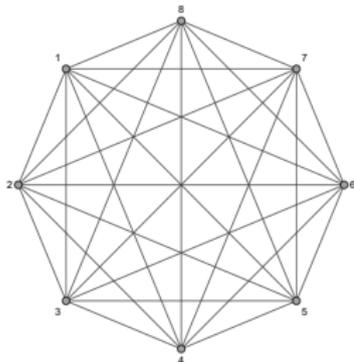
Al utilizar el deslizador Orden del grafo van apareciendo distintos grafos vacíos hasta el orden 8.

- 2 **Grafo completo:** un grafo completo se entiende como un grafo simple, donde para cualquier par de vértices siempre existe una arista que los une (exceptuando un nodo con él mismo). Se suele denotar como  $K_n$ , siendo  $n$  la cantidad de nodos que contiene. La librería **VilCretas** posee el comando `GrafoCompleto` que permite su construcción. Por ejemplo, la siguiente sentencia forma el grafo completo  $K_8$ :

```
In[ ] :=
```

```
GrafoCompleto[8]
```

```
Out[ ] =
```

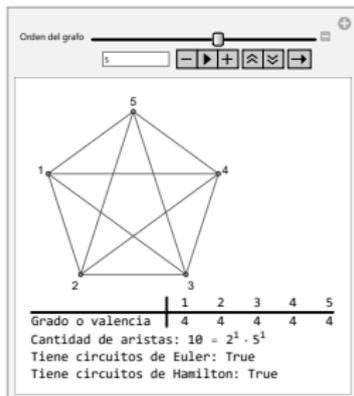


También, GrafoCompleto incluye la opción analizar  $\rightarrow$  True. En este comando, analizar además de mostrar una animación con distintos grafos completos, provee información adicional referida a las valencias de los vértices, a la cantidad de lados del grafo y a la existencia de circuitos de *Euler* y de *Hamilton*, temas que serán desarrollados más adelante. Así:

**In[ ] :=**

GrafoCompleto[8, analizar  $\rightarrow$  True]

**Out[ ] =**



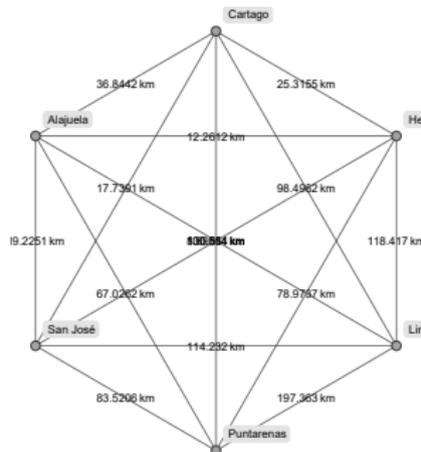
Los grafos completos tienen interesantes aplicaciones prácticas. Por ejemplo, el comando `GrafoCountryRegions` de **VilCretas**, siempre ofrece como salida un grafo completo ponderado que relaciona las principales regiones de un país, siendo los pesos de las aristas la distancia real en kilómetros entre cada provincia. A continuación, se muestra la salida de `GrafoCountryRegions` sobre el país Costa Rica:

```
In[ ] :=
```

```
GrafoCountryRegions["Costa Rica"]
```

Se obtiene la siguiente salida:

**Out[ ] =**



La lista de países admisibles por la instrucción `GrafoCountryRegions` se genera en *Mathematica* ejecutando la sentencia `ListaCountries[]` de **VilCretas**. El alumno debe observar que siempre el nombre del país se pasa a `GrafoCountryRegions` entre comillas, es decir, como un “string”.

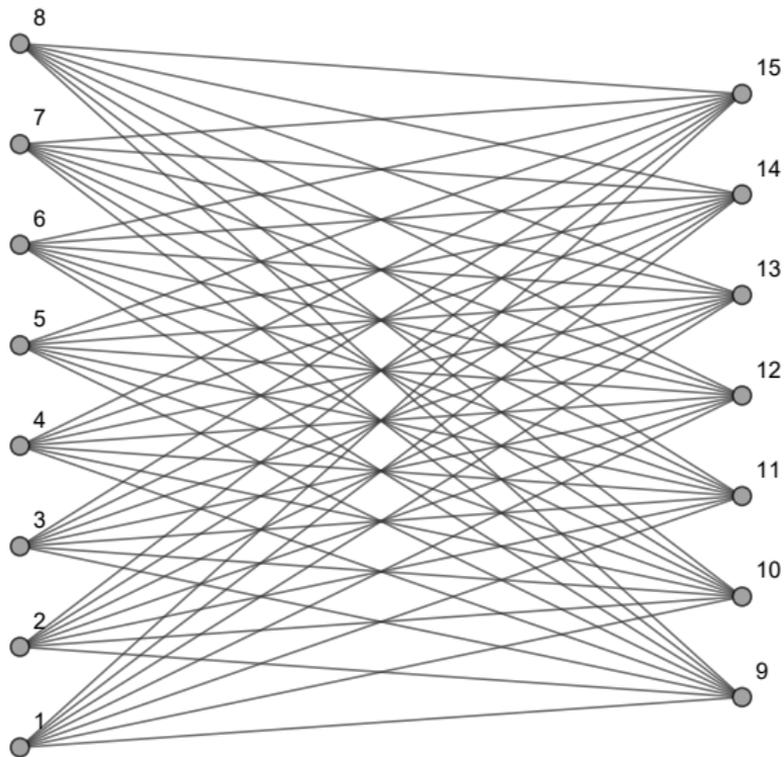
- 8 **Grafo bipartito completo:** este tipo de grafo simple es aquel formado por dos conjuntos disjuntos de vértices (su intersección es vacía) y todas las posibles aristas que los unen entre sí, considerando que los nodos de un mismo conjunto no están vinculados por una arista. El grafo bipartito con  $n$  y  $m$  nodos disjuntos, se representa por  $K_{n,m}$ . A continuación, se muestra el grafo bipartito completo  $K_{8,7}$ , elaborado mediante el comando `GrafoBipartitoCompleto` de la librería **VilCretas**:

```
In[ ] :=
```

```
GrafoBipartitoCompleto[8, 7]
```

Se obtiene la siguiente salida:

**Out[ ] =**

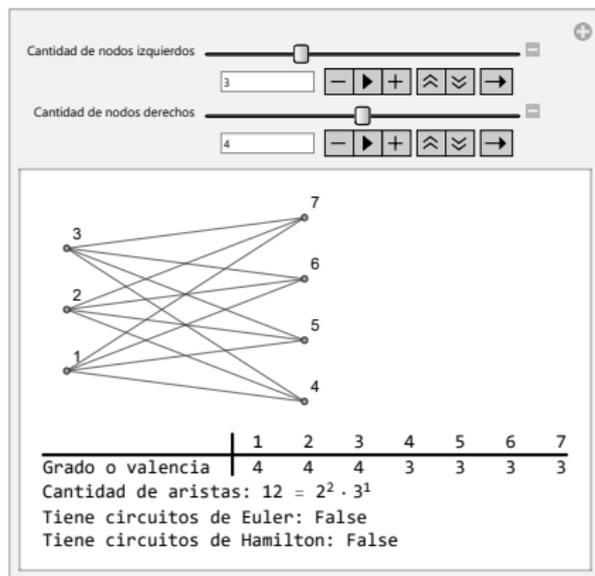


De manera análoga, GrafoBipartitoCompleto posee la opción analizar  $\rightarrow$  True. Veamos:

**In[ ] :=**

GrafoBipartitoCompleto[8, 7, analizar  $\rightarrow$  True]

**Out[ ] =**



Todo grafo bipartito completo presenta entonces dos órdenes  $n$  y  $m$ , uno para indicar la cantidad de vértices distintos que aparecen a la izquierda ( $n$ ) y el otro para señalar el número de nodos diferentes que se tienen a la derecha ( $m$ ). Nunca los vértices de un mismo lado en un grafo bipartito estarán relacionados por una arista.



Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-98.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-98.zip)

# Uso del comando GrafoVacio



Explicación en video

<https://youtu.be/ylhaSKe3-4k>

# Empleo de la sentencia GrafoCompleto



Explicación en video

<https://youtu.be/-4RlqpEeasg>

# Utilización de la instrucción GrafoBipartitoCompleto



Explicación en video

<https://youtu.be/hEqBCbwiRAg>

## Definición 5.4

Otro concepto primordial en teoría de grafos estriba en la definición de valencia o grado.

### Definition (5.4)

Sea  $G = (V, E)$  un grafo. Si  $G$  es no dirigido y  $a \in V$ , se llama valencia o grado del nodo  $a$ , al número de lados de  $E$  que son incidentes sobre  $a$ , la valencia se representa como  $\theta(a)$ . Si en  $a$  ocurre un lazo, el ciclo suma 2 unidades a su grado. Por otra parte, si  $G$  es dirigido, se llama valencia o grado interno de  $a$ , denotado  $\theta_*(a)$ , al número de aristas de  $G$  que llegan al vértice  $a$ . La cantidad de lados de  $G$  que salen del nodo  $a$ , recibe el nombre de valencia o grado externo de  $a$  y se representa  $\theta^*(a)$ . Un nodo  $a$  tal que  $\theta(a) = 0$ , o bien,  $\theta_*(a) = \theta^*(a) = 0$  se denomina vértice aislado (sin aristas incidentes).

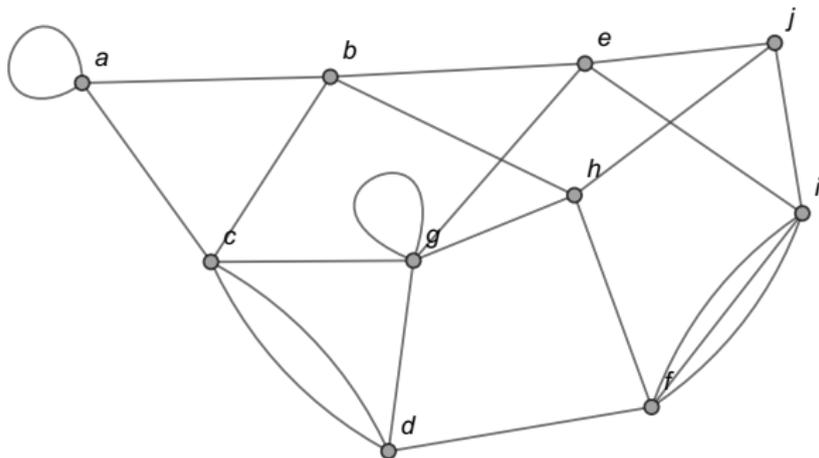
## Comentario sobre la definición 4

De acuerdo con la definición 4 todos los vértices de un grafo dirigido o no dirigido tienen un número asociado llamado grado o valencia. Este valor se determina observando la cantidad de lados del grafo que tienen como extremo al vértice de interés. También, cabe recalcar, que si un nodo de un grafo no dirigido posee lazos o ciclos (recuerde: un ciclo es una arista que relaciona un vértice con él mismo), cada lazo suma 2 unidades a la valencia, según lo propuesto por la definición 4.

Consideremos algunos ejemplos.

### Example (5.1)

Determine las valencias de los vértices del grafo:



## Solución del ejemplo 5.1

En el grafo del enunciado el conjunto de nodos  $V$  es tal que  $V = \{a, b, c, d, e, f, g, h, i, j\}$ . Por cada vértice de  $V$  existe un grado o una valencia vinculada.  $\theta(a) = 4$  pues hay 2 aristas incidentes a  $a$  y un lazo que suma 2 unidades más a la valencia.  $\theta(b) = 4$  pues existen 4 lados del grafo con extremo  $b$ .  $\theta(c) = 5$  dado que hay 5 aristas con extremo  $c$ .  $\theta(d) = 4$  al existir 4 lados incidentes a  $d$ .  $\theta(e) = 4$  pues el grafo posee 4 aristas con extremo  $e$ .  $\theta(f) = 5$  a razón de visualizar 5 lados incidentes a  $f$ .  $\theta(g) = 6$  al existir 4 lados incidentes a  $g$  y un ciclo que suma 2 unidades más al grado.  $\theta(h) = 4$  al existir 4 lados con extremo igual a  $h$ .  $\theta(i) = 5$  al existir 5 aristas incidentes a  $i$ . Y finalmente,  $\theta(j) = 3$  pues el grafo contiene 3 lados con extremo  $j$ .

## Solución del ejemplo 5.1

Estos resultados se pueden resumir en una tabla de valencias o grados:

$V$	$a$	$b$	$c$	$d$	$e$	$f$	$g$	$h$	$i$	$j$
Grado o valencia	4	4	5	4	4	5	6	4	5	3

La librería **VilCretas** suministra un comando que calcula las valencias o los grados de los vértices de un grafo. Esta instrucción se denomina *Valencias* y recibe como parámetro el grafo respectivo. En este ejercicio, al emplear la sentencia *Valencias*, se obtiene en *Mathematica*:

**In[ ] :=**

```
grafo = Grafo[{{a, a}, {a, b}, {a, c}, {b, c}, {b, e},
{b, h}, {c, d}, {c, d}, {c, g}, {d, f}, {d, g}, {e, g},
{e, i}, {e, j}, {f, h}, {f, i}, {f, i}, {f, i}, {g, g},
{g, h}, {h, j}, {i, j}}];
Valencias[grafo]
```

## Solución del ejemplo 5.1

Se obtiene la siguiente salida:

**Out[ ]** =

	a	b	c	e	h	d	g	f	i	j
Grado o valencia	4	4	5	4	4	4	6	5	5	3

En el **Out[ ]** los vértices no aparecen en orden alfabético. El orden mostrado es elegido por el software en función de cómo fueron incluidas las aristas en el comando `Grafo`.



## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-99.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-99.zip)

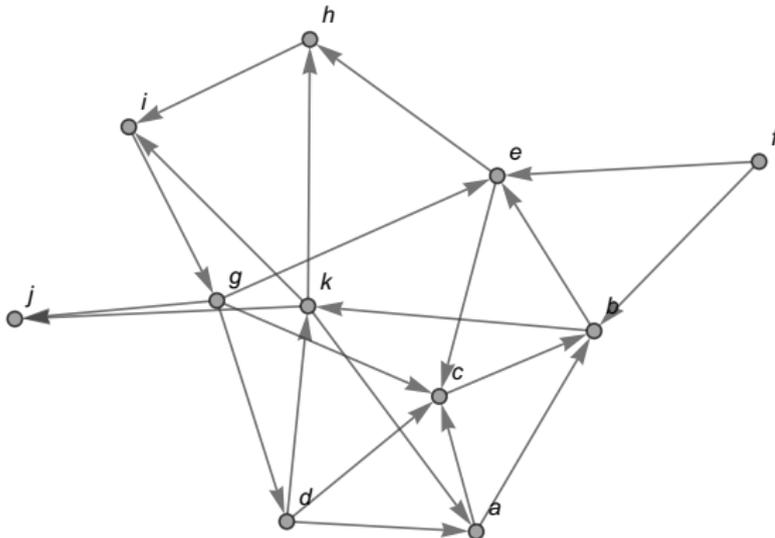


## Explicación en video

<https://youtu.be/qrmKR7Yx7vs>

## Example (5.2)

Halle los grados de los nodos del grafo:



## Solución del ejemplo 5.2

El grafo del enunciado es dirigido. Según la definición 4 existen dos tipos de grados, las valencias internas ( $\theta_*$ ) que corresponden al número de lados entrantes al nodo y las valencias externas ( $\theta^*$ ) referidas a la cantidad de aristas salientes del vértice. Además, en el grafo de este ejemplo  $V = \{a, b, c, d, e, f, g, h, i, j, k\}$ . Luego, la valencia interna de  $a$  es igual a 2 pues hay 2 flechas que entran a  $a$  y su grado externo también es 2 al existir dos flechas que salen de ese nodo. En  $b$ ,  $\theta_*(b) = 3$  dado que hay 3 flechas entrantes a  $b$  y  $\theta^*(b) = 2$  al poseer el grafo 2 lados salientes de  $b$ .  $\theta_*(c) = 4$  pues existen 4 aristas entrantes a  $c$  y  $\theta^*(c) = 1$  al aparecer un lado saliente de  $c$ .  $\theta_*(d) = 1$  a razón de haber un lado entrante a  $d$  y  $\theta^*(d) = 3$  al existir en el grafo 3 flechas salientes de  $d$ .  $\theta_*(e) = 3$  pues existen 3 flechas que entran a  $e$  y  $\theta^*(e) = 2$  al observarse en el grafo 2 flechas que salen del vértice  $e$ .  $\theta_*(f) = 0$  dado que no hay ninguna arista que entre al nodo  $f$  y  $\theta^*(f) = 2$  pues existen 2 flechas que salen del vértice  $f$ .

## Solución del ejemplo 5.2

Aunque, el grado interno de  $f$  es igual a 0, no se considera a  $f$  un nodo aislado pues su valencia externa no es igual a 0.  $\theta_*(g) = 1$  pues entra a  $g$  una arista y  $\theta^*(g) = 4$  al existir 4 lados que salen de ese nodo.  $\theta_*(h) = 2$  a razón de haber dos flechas entrantes a  $h$  y  $\theta^*(h) = 1$  al contener el grafo un lado saliente de  $h$ .  $\theta_*(i) = 2$ , hay 2 flechas que entran al vértice  $i$  y  $\theta^*(i) = 1$ , hay un lado que sale del nodo  $i$ .  $\theta_*(j) = 2$  dado que sobre  $j$  hay 2 lados entrantes y  $\theta^*(j) = 0$  pues no hay ninguna arista saliente de  $j$ . Aquí, el vértice  $j$  tampoco es aislado en el grafo, el grado interno de  $j$  no es igual a 0.  $\theta_*(k) = 2$  a razón de existir dos aristas entrantes a  $k$  y  $\theta^*(k) = 4$  al contener el grafo 4 flechas salientes de  $k$ .

## Solución del ejemplo 5.2

En resumen:

$V$	$a$	$b$	$c$	$d$	$e$	$f$	$g$	$h$	$i$	$j$	$k$
Grado o valencia interna	2	3	4	1	3	0	1	2	2	2	2
Grado o valencia externa	2	2	1	3	2	2	4	1	1	0	4

La instrucción `Valencias` resuelve además, el cálculo de los grados en un grafo dirigido. El comando detecta de manera automática si los lados del grafo pasado como parámetro son dirigidos. En este ejemplo:

```
In[ ] :=
```

```
grafo = Grafo[{{a, b}, {a, c}, {b, k}, {b, e}, {c, b},
{d, a}, {d, c}, {d, k}, {e, c}, {e, h}, {f, b}, {f, e},
{g, c}, {g, d}, {g, e}, {g, j}, {h, i}, {i, g}, {k, j},
{k, a}, {k, h}, {k, i}}, dirigido -> True];
Valencias[grafo]
```

## Solución del ejemplo 5.2

Se obtiene la siguiente salida:

**Out[ ] =**

	a	b	c	k	e	d	h	f	g	j	i
Grado o valencia interna	2	3	4	2	3	1	2	0	1	2	2
Grado o valencia externa	2	2	1	4	2	3	1	2	4	0	1

El **Out[ ]** no muestra un orden alfabético. Como ya se mencionó en la solución del ejemplo anterior, *Mathematica* selecciona el orden de los vértices en la tabla, de acuerdo con la forma en cómo aparecen las aristas del grafo en el comando Grafo.



## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-100.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-100.zip)

### Example (5.3)

Encuentre los grados de los vértices en los grafos completos  $K_3$ ,  $K_4$  y  $K_5$ .  
¿Existe una fórmula que calcule las valencias de los nodos en  $K_n$ ,  $\forall n$ ,  
 $n \in \mathbb{N}$ ?

### Solución del ejemplo 5.3

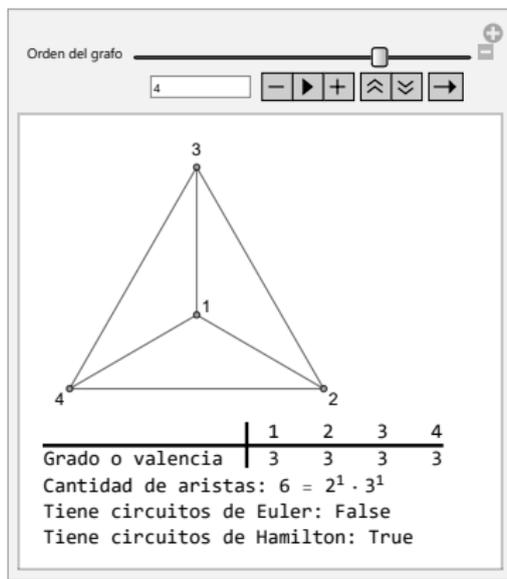
Las valencias de los vértices en los grafos completos  $K_3$ ,  $K_4$  y  $K_5$ , se pueden obtener empleando la instrucción `GrafoCompleto` y su atributo `analizar` -> `True`:

```
In[ ] :=
```

```
GrafoCompleto[5, analizar -> True]
```

## Solución del ejemplo 5.3

Se obtiene la siguiente salida:

**Out[ ] =**

### Solución del ejemplo 5.3

Al jugar con el deslizador de la animación se concluye que en  $K_3$  todos los nodos tienen la misma valencia igual a 2, en  $K_4$  todos los vértices tienen el mismo grado igual a 3 y en  $K_5$  todos los nodos tienen la misma valencia igual a 4. En general, al observar estos casos particulares es fácil conjeturar que en el grafo  $K_n$ ,  $\forall n, n \in \mathbb{N}$ , la valencia de cualquier vértice  $v_i$ ,  $1 \leq i \leq n, i \in \mathbb{N}$ , corresponde a  $n - 1$ , es decir,  $\theta(v_i) = n - 1$ . Una prueba formal de este resultado se efectuaría utilizando la técnica de demostración por inducción matemática, sin embargo, ésta será omitida invitando al estudiante a realizarla por su propia cuenta.



## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-101.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-101.zip)

### Example (5.4)

Determine las valencias de los vértices en los grafos bipartitos completos  $K_{3,4}$ ,  $K_{4,5}$  y  $K_{5,6}$ . ¿Existe una fórmula que calcule los grados de los nodos en  $K_{n,m}$ ,  $\forall n, m \in \mathbb{N}$ ?

## Solución del ejemplo 5.4

De forma análoga, las valencias de los vértices en los grafos bipartitos completos  $K_{3,4}$ ,  $K_{4,5}$  y  $K_{5,6}$ , se hallan recurriendo al comando `GrafoBipartitoCompleto` y su opción `analizar -> True`:

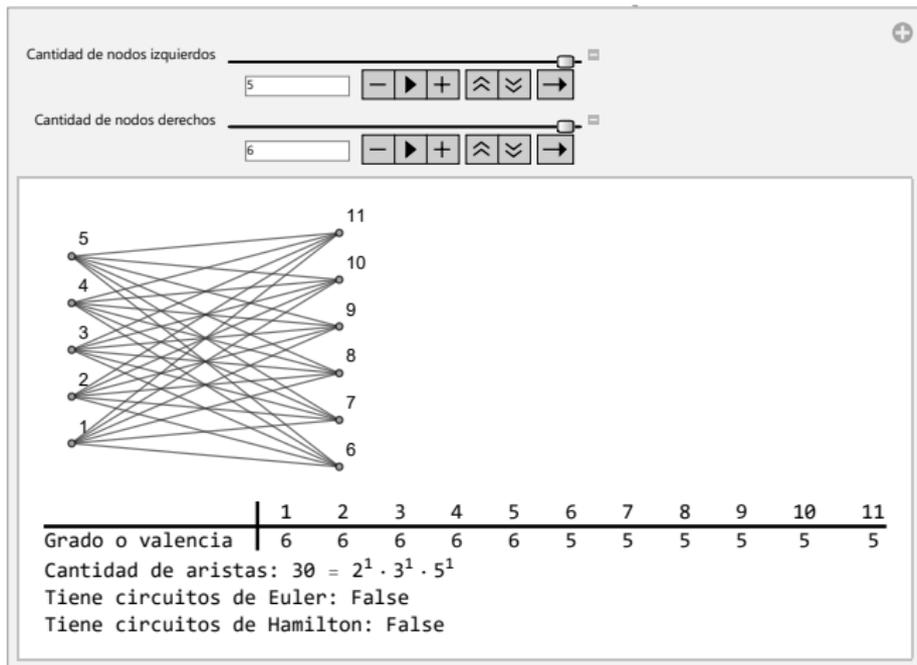
```
In[ ] :=
```

```
GrafoBipartitoCompleto[5, 6, analizar -> True]
```

Se obtiene la siguiente salida:

## Solución del ejemplo 5.4

Out[ ] =



## Solución del ejemplo 5.4

De donde, se aprecia en la animación que  $K_{3,4}$  tiene dos tipos de valencias, los 3 vértices que se encuentran a la izquierda tienen grado igual a 4 y los 4 nodos ubicados a la derecha tienen valencia igual a 3. En el grafo  $K_{4,5}$ , los 4 vértices izquierdos tienen el mismo grado, igual a 5 y los 5 nodos derechos poseen la misma valencia, igual a 4. Finalmente, en  $K_{5,6}$  los 5 vértices ubicados al lado izquierdo tienen valencia igual a 6 y los 6 nodos del lado derecho poseen grado igual a 5. En general, por el comportamiento observado, se concluye que en un grafo bipartito completo  $K_{n,m}$ ,  $\forall n, m \in \mathbb{N}$ , hay dos tipos de valencias: el grado de los  $n$  vértices ubicados al lado izquierdo que corresponde a  $m$  y la valencia de los  $m$  nodos ubicados al lado derecho que es igual a  $n$ . En conclusión, simbólicamente, si  $v_i^I$  y  $v_j^D$  representan un vértice de  $K_{n,m}$  izquierdo y derecho, respectivamente,  $1 \leq i \leq n$ ,  $1 \leq j \leq m$ ,  $i, j \in \mathbb{N}$ , entonces  $\theta(v_i^I) = m$  y  $\theta(v_j^D) = n$ .



## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-102.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-102.zip)

## Teorema 5.1

Las valencias o grados de los vértices de un grafo son muy importantes pues tienen diversas aplicaciones en el contexto de esta teoría. En un grafo no dirigido, la suma de las valencias da como resultado el doble de la cantidad de lados que contiene el grafo. El siguiente teorema garantiza esta propiedad.

### Theorem (5.1)

Sea  $G = (V, E)$  un grafo no dirigido con  $V = \{v_1, v_2, \dots, v_n\}$  y  $p$  el número de aristas diferentes de  $G$  entonces:

$$\sum_{i=1}^n \theta(v_i) = 2p$$

## Comentario sobre el teorema 9

El teorema 9 plantea un procedimiento formal para encontrar la cantidad de lados  $p$  distintos de un grafo no dirigido. En algunos casos, ese valor es fácilmente deducible con solo observar el grafo y contar manualmente sus aristas, sin embargo, en otras ocasiones si el grafo contiene una cantidad significativa de lados, podría ocurrir que una visualización de la figura no permita determinar el valor de  $p$ . El alumno debe notar el procedimiento expuesto en el teorema 9,  $p$  corresponde a la suma de todas las valencias de los vértices del grafo dividida entre 2. Lo anterior, naturalmente está implicando de manera implícita, que al sumar los grados de todos los nodos de un grafo no dirigido, el resultado siempre corresponderá a un número par.

### Example (5.5)

Encuentre la cantidad de aristas diferentes del grafo del ejemplo 5, usando el teorema 9.

## Solución del ejemplo 5.5

Por lo resuelto en el ejemplo 5:

$V$	$a$	$b$	$c$	$d$	$e$	$f$	$g$	$h$	$i$	$j$
Grado o valencia	4	4	5	4	4	5	6	4	5	3

Si se desea hallar la cantidad de lados  $p$  del grafo en cuestión, se debe sumar la fila de valencias de la tabla:

$$4 + 4 + 5 + 4 + 4 + 5 + 6 + 4 + 5 + 3 = 44$$

En consecuencia:

$$p = 44 \div 2 = 22$$

## Solución del ejemplo 5.5

El estudiante podrá corroborar sin dificultad, realizando un conteo manual, que efectivamente el grafo del ejemplo 5 posee 22 lados distintos.

El procedimiento ya expuesto basado en el teorema 9, se puede automatizar en *Wolfram*, así:

**In[ ] :=**

```
grafo = Grafo[{{a, a}, {a, b}, {a, c}, {b, c}, {b, e},  
{b, h}, {c, d}, {c, d}, {c, g}, {d, f}, {d, g}, {e, g},  
{e, i}, {e, j}, {f, h}, {f, i}, {f, i}, {f, i}, {g, g},  
{g, h}, {h, j}, {i, j}}];
```

```
Total[Valencias[grafo][[1,1]]]/2
```

**Out[ ] =**

22

## Solución del ejemplo 5.5

La expresión `[[1,1]]` tiene una función similar a lo explicado en el capítulo de relaciones, donde se había utilizado `[[1]]` para extraer de un arreglo rectangular una matriz. En este ejercicio, `[[1,1]]` extrae de la tabla devuelta por `Valencias` un vector cuyas componentes son los grados de cada uno de los nodos del grafo. El comando `Total` empleado en el `In[ ]` es una sentencia de *Wolfram Mathematica* que suma las componentes de un vector y en este código en particular, resuelve la suma de todas las valencias. Al dividir esa suma entre 2, se calcula el número de aristas del grafo, de acuerdo con el teorema 9.



## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-103.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-103.zip)

### Example (5.6)

La instrucción `GrafoVilCretas []` del paquete **VilCretas** genera un grafo pseudoaleatorio que siempre muestra la palabra “VilCretas”. Determine la cantidad de lados diferentes que posee este grafo.

## Solución del ejemplo 5.6

En *Mathematica* al correr el comando `GrafoVilCretas[]`, se obtiene:

```
In[ ] :=
```

```
GrafoVilCretas[]
```

```
Out[ ] =
```



## Solución del ejemplo 5.6

Este grafo es pseudoaleatorio. Si se vuelve a correr la sentencia `GrafoVilCretas`, el grafo resultante variará muy posiblemente la cantidad de aristas presentes pero siempre mostrando la palabra “VilCretas”. El estudiante preverá que encontrar sobre este grafo el número total de aristas no es una tarea factible a través de un conteo visual. Por consiguiente, se usará el teorema 9, veamos:

```
In[ ] :=
```

```
grafo = GrafoVilCretas[];  
Total[Valencias[grafo][[1, 1]]]/2
```

```
Out[ ] =
```

```
14628
```

## Solución del ejemplo 5.6

Finalmente, el número de aristas buscado corresponde a 14628. Este número variará de una ejecución a otra, al ser GrafoVilCretas un grafo pseudoaleatorio.



**Descargue un archivo**

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-104.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-104.zip)

### Example (5.7)

¿Se puede construir un grafo donde sus vértices  $V = \{a, b, c, d, e, f, g, h\}$  tienen como valencias  $\theta(a) = 5$ ,  $\theta(b) = 4$ ,  $\theta(c) = 0$ ,  $\theta(d) = 2$ ,  $\theta(e) = 3$ ,  $\theta(f) = 2$ ,  $\theta(g) = 6$  y  $\theta(h) = 1$ ?

## Solución del ejemplo 5.7

Al sumar las valencias brindadas en el ejemplo, se tiene:

$$\begin{aligned}\theta(a) + \theta(b) + \theta(c) + \theta(d) + \theta(e) + \theta(f) + \theta(g) + \theta(h) \\ = 5 + 4 + 0 + 2 + 3 + 2 + 6 + 1 = 23\end{aligned}$$

Como se mencionó en el comentario de la página 91, la suma de los grados de todos los vértices de un grafo siempre debe dar un número par. Como consecuencia de ello, no es posible construir un grafo con las valencias especificadas en el enunciado, pues el resultado de la suma derivó en un número impar.

### Example (5.8)

Encuentre una fórmula que calcule la cantidad de lados de un grafo completo  $K_n$ ,  $\forall n, n \in \mathbb{N}$ .

## Solución del ejemplo 5.8

En el ejemplo 7 se comprobó que  $\theta(v_i) = n - 1$ , siendo  $v_i$  un vértice de  $K_n$ ,  $\forall n, n \in \mathbb{N}, i \in \{1, \dots, n\}$ . Por el teorema 9, sabemos que el número  $p$  de aristas distintas de  $K_n$ , viene dado por:

$$2p = \sum_{i=1}^n \theta(v_i) \Rightarrow 2p = \sum_{i=1}^n (n-1) \Rightarrow 2p = n(n-1) \Rightarrow p = \frac{n(n-1)}{2}$$

La fórmula  $p = \frac{n(n-1)}{2}$  proporciona la cantidad de aristas diferentes en el grafo completo  $K_n$ ,  $\forall n, n \in \mathbb{N}$ .

### Example (5.9)

Determine una fórmula para el número de aristas de un grafo bipartito completo  $K_{n,m}$ ,  $\forall n, m \in \mathbb{N}$ .

## Solución del ejemplo 5.9

En el ejemplo 8 se concluyó que en un grafo bipartito completo  $K_{n,m}$ ,  $\forall n, m \in \mathbb{N}$ , existen únicamente dos tipos de valencias. Los  $n$  nodos izquierdos ( $v_i^I$ ,  $1 \leq i \leq n$ ,  $i \in \mathbb{N}$ ) tienen como grado el valor de  $m$  y los  $m$  vértices derechos ( $v_j^D$ ,  $1 \leq j \leq m$ ,  $j \in \mathbb{N}$ ) poseen como valencia el número  $n$ . Luego, empleando el teorema 9, se tiene:

$$2p = \sum_{i=1}^n \theta(v_i^I) + \sum_{j=1}^m \theta(v_j^D) \Rightarrow 2p = \sum_{i=1}^n m + \sum_{j=1}^m n$$

$$\Rightarrow 2p = n \cdot m + m \cdot n \Rightarrow p = \frac{2 \cdot n \cdot m}{2} = n \cdot m$$

La fórmula  $p = n \cdot m$  calcula la cantidad de lados distintos en el grafo bipartito completo  $K_{n,m}$ ,  $\forall n, m \in \mathbb{N}$ .

Cuando un grafo es dirigido existe un resultado similar al expuesto en el teorema 9. En un digrafo la suma de los grados internos siempre será igual a la suma de las valencias externas de todos los vértices y ese valor es precisamente el número de lados distintos contenidos en el grafo. No se enunciará esta característica como un teorema, sin embargo, el estudiante debe tener la propiedad muy presente.

Veamos un ejemplo de ello.

### Example (5.10)

Halle utilizando propiedades, el número de aristas del grafo del ejemplo 6.

## Solución del ejemplo 5.10

El grafo del ejemplo 6 es dirigido. Por lo comentado con anterioridad, el número de aristas distintas del digrafo se calcula al sumar los grados internos, o bien, los grados externos, de todos sus nodos. Ya sabemos, por lo expuesto en la solución del ejemplo 6:

$V$	$a$	$b$	$c$	$d$	$e$	$f$	$g$	$h$	$i$	$j$	$k$
Grado o valencia interna	2	3	4	1	3	0	1	2	2	2	2
Grado o valencia externa	2	2	1	3	2	2	4	1	1	0	4

(1)

Al sumar las valencias internas:

$$2 + 3 + 4 + 1 + 3 + 0 + 1 + 2 + 2 + 2 + 2 = 22$$

## Solución del ejemplo 5.10

Por lo que, el digrafo contiene 22 lados distintos. Este valor, también se puede encontrar, al sumar los grados externos de la tabla 1:

$$2 + 2 + 1 + 3 + 2 + 2 + 4 + 1 + 1 + 0 + 4 = 22$$

En *Wolfram Mathematica* el procedimiento descrito es automatizable, en analogía a lo compartido en la solución del ejemplo 10:

**In[ ] :=**

```
grafo = Grafo[{{a, b}, {a, c}, {b, k}, {b, e}, {c, b},
{d, a}, {d, c}, {d, k}, {e, c}, {e, h}, {f, b}, {f, e},
{g, c}, {g, d}, {g, e}, {g, j}, {h, i}, {i, g}, {k, j},
{k, a}, {k, h}, {k, i}}, dirigido -> True];
Total[Valencias[grafo][[1,1]]]
```

**Out[ ] =**

22

## Solución del ejemplo 5.10

El `[[1,1]]` en el código del `ln[ ]` señala que se están empleando los grados internos de los nodos para determinar el número total de lados del grafo. Si se reemplaza `[[1,1]]` por `[[1,2]]` en el `ln[ ]`, se estaría calculando la cantidad de aristas del digrafo pero en función de las valencias externas. Desde luego, la salida continuará siendo 22.

## Nota

El paquete **VilCretas** brinda la posibilidad de determinar el número de aristas y de vértices distintos de un grafo utilizando los comandos `CantAristas` y `CantNodos`, respectivamente. Estas dos instrucciones reciben como parámetro el grafo de interés, sea éste dirigido o no dirigido. Se insta al lector a explorar su uso.



## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-105.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-105.zip)

- Como se recordará, toda relación binaria tiene una forma de representación matricial, los grafos no son la excepción. La sección que sigue trata este importante tema.

# Representaciones para un grafo

Prof. Enrique Vílchez Quesada

Universidad Nacional de Costa Rica

## Definición 5.5

La definición 4.3 del capítulo de relaciones, introdujo un arreglo bidimensional que representa una relación binaria  $R$ . Si en particular  $R$  es un grafo  $G$ , a su matriz de representación se le llama matriz de adyacencia de  $G$  y se le denota como  $M_A$ . La definición 16 lo formaliza.

### Definition (5.5)

Sea  $G = (V, E)$  un grafo con  $V = \{v_1, v_2, \dots, v_n\}$  y  $E = \{e_1, e_2, \dots, e_p\}$ . Se llama matriz de adyacencia de  $G$  en el orden de los elementos de  $V$ , a la matriz denotada  $M_A$ ,  $M_A = (a_{ij})$  de tamaño  $n$  por  $n$ , tal que  $a_{ij} = k$ , siendo  $k$  el número de aristas que unen los vértices  $v_i$  y  $v_j$ ,  $k \geq 0$ ,  $k \in \mathbb{N} \cup \{0\}$ .

## Comentario sobre la definición 16

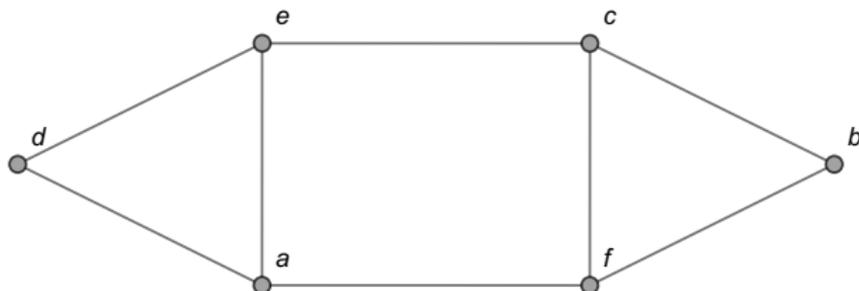
Un grafo  $G = (V, E)$  puede poseer varias matrices de adyacencia, dependiendo del orden tomado para los elementos del conjunto  $V$ . Todas las matrices resultantes, eso sí, representan el mismo grafo. Para efectos del presente libro, el orden de los elementos de  $V$  casi siempre se asume, por razones de simplicidad, en correspondencia con el orden del abecedario. A esta matriz se le llama matriz de adyacencia pues estudia la adyacencia de un vértice cualesquiera del grafo con respecto a todos los demás, incluyéndose a él mismo.

Por otra parte, la definición 16 en comparación con la 4.3, propone que la matriz de representación de una relación binaria  $R$  correspondiente a un grafo  $G$ , puede no ser una matriz booleana (sus entradas no son necesariamente ceros o unos). Si un grafo  $G$  contiene aristas múltiples cualquier matriz de adyacencia que represente a  $G$  no será una matriz booleana.

Se desarrollarán algunos ejemplos de construcción de una matriz de adyacencia para un grafo.

### Example (5.11)

Determine una matriz de adyacencia del grafo:



## Solución del ejemplo 5.11

En el grafo del enunciado el conjunto de vértices  $V$  es tal que  $V = \{a, b, c, d, e, f\}$ , por lo que una matriz de adyacencia que lo representa, se construye al tomar  $V$  (puede ser en ese orden) tanto en las filas como en las columnas de la matriz:

$$\begin{array}{c} a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{pmatrix} & a & b & c & d & e & f \\ \square & \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square & \square \end{pmatrix}$$

## Solución del ejemplo 5.11

Ahora, cada una de las entradas en la matriz de adyacencia se completa analizando si los nodos están relacionados y en caso afirmativo, contando el número de lados que los unen. Así, comenzado con el vértice  $a$ , se observa su relación con  $d$ ,  $e$  y  $f$ :

$$\begin{array}{c} a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{pmatrix} & a & b & c & d & e & f \\ a & 0 & 0 & 0 & 1 & 1 & 1 \\ b & \square & \square & \square & \square & \square & \square \\ c & \square & \square & \square & \square & \square & \square \\ d & \square & \square & \square & \square & \square & \square \\ e & \square & \square & \square & \square & \square & \square \\ f & \square & \square & \square & \square & \square & \square \end{pmatrix}$$

## Solución del ejemplo 5.11

El 1 indica que en el grafo existe únicamente una arista que une el nodo  $a$  con  $d$ ,  $e$  y  $f$ , respectivamente. Al ubicarnos en la segunda fila de la matriz, tomando el vértice  $b$ , se aprecia en el grafo la existencia de una única arista que relaciona  $b$  con los nodos  $c$  y  $f$ , luego:

$$\begin{array}{c}
 \\
 a \\
 b \\
 c \\
 d \\
 e \\
 f
 \end{array}
 \begin{pmatrix}
 & a & b & c & d & e & f \\
 0 & 0 & 0 & 1 & 1 & 1 \\
 0 & 0 & 1 & 0 & 0 & 1 \\
 \square & \square & \square & \square & \square & \square \\
 \square & \square & \square & \square & \square & \square \\
 \square & \square & \square & \square & \square & \square \\
 \square & \square & \square & \square & \square & \square
 \end{pmatrix}$$

## Solución del ejemplo 5.11

Ahora en el vértice  $c$ , hay una única arista que une  $c$  con  $b$ ,  $e$  y  $f$  :

$$\begin{array}{c} a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{pmatrix} a & b & c & d & e & f \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \end{pmatrix}$$

## Solución del ejemplo 5.11

El nodo  $d$  es adyacente a los vértices  $a$  y  $e$  :

$$\begin{array}{c} a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{pmatrix} & a & b & c & d & e & f \\ a & 0 & 0 & 0 & 1 & 1 & 1 \\ b & 0 & 0 & 1 & 0 & 0 & 1 \\ c & 0 & 1 & 0 & 0 & 1 & 1 \\ d & 1 & 0 & 0 & 0 & 1 & 0 \\ e & \square & \square & \square & \square & \square & \square \\ f & \square & \square & \square & \square & \square & \square \end{pmatrix}$$

## Solución del ejemplo 5.11

El vértice  $e$  está relacionado mediante un lado único, con los nodos  $a$ ,  $c$  y  $d$ :

$$\begin{array}{c} a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{pmatrix} a & b & c & d & e & f \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ \square & \square & \square & \square & \square & \square \end{pmatrix}$$

## Solución del ejemplo 5.11

Finalmente, en el nodo  $f$  existe adyacencia con respecto a los vértices  $a$ ,  $b$  y  $c$ :

$$\begin{array}{c} a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{pmatrix} a & b & c & d & e & f \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

## Solución del ejemplo 5.11

Se concluye que una matriz de adyacencia  $M_A$  usando el orden  $V = \{a, b, c, d, e, f\}$  para el grafo dado es:

$$M_A = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix} \quad (2)$$

## Nota

Una matriz de adyacencia sobre un grafo  $G$  siempre se construye por filas y no por columnas. También, cabe destacar en el presente ejemplo, que  $M_A$  es una matriz booleana pues el grafo no contiene lados paralelos.

## Solución del ejemplo 5.11

La librería **VilCretas** proporciona el comando `MGrafo` que al recibir un grafo y un orden específico para sus vértices, retorna una matriz de adyacencia que lo representa. La instrucción cuenta con el atributo `table -> True` encargada de etiquetar las filas y las columnas de la matriz, permitiendo observar el orden de los nodos usado en su elaboración. En este ejemplo, la matriz **2** se genera mediante `MGrafo`, así:

**In[ ]** :=

```
grafo = Grafo[{{a, d}, {a, e}, {a, f}, {b, c}, {b, f},  
{c, e}, {c, f}, {d, e}}];  
MGrafo[grafo, {a, b, c, d, e, f}]  
MGrafo[grafo, {a, b, c, d, e, f}, table -> True]
```

## Solución del ejemplo 5.11

Se obtiene la siguiente salida:

**Out[ ] =**

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

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

## Solución del ejemplo 5.11

La segunda salida obedece al uso de la opción `table`  $\rightarrow$  True de `MGrafo`.

Si se cambia el orden de los elementos del conjunto pasado a `MGrafo`, se evidencia que un grafo puede tener distintas matrices de adyacencia. A manera de ejemplo, si se utiliza el orden  $\{b, a, e, f, c, d\}$  en `MGrafo`, se retorna una matriz de adyacencia distinta a la matriz 2:

**In[ ] :=**

`MGrafo[grafo, {b, a, e, f, c, d}]`

**Out[ ] =**

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$



## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-106.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-106.zip)

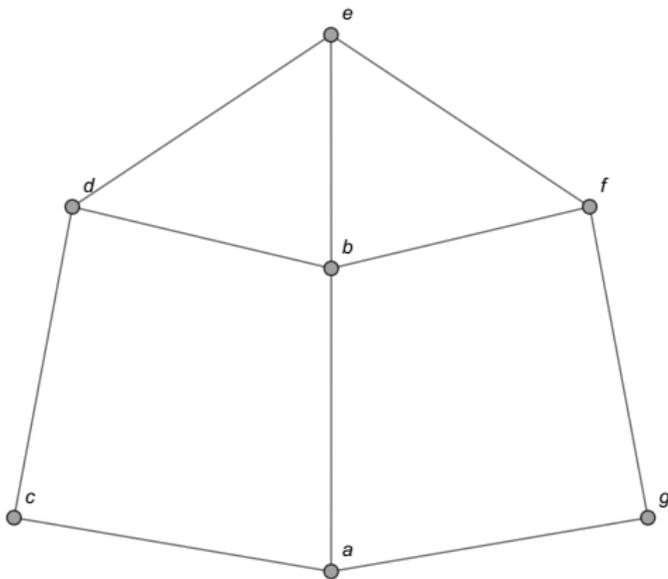


## Explicación en video

<https://youtu.be/j197roRuyXc>

## Example (5.12)

Halle una matriz de adyacencia para:



## Solución del ejemplo 5.12

Si se usa el orden  $V = \{a, b, c, d, e, f, g\}$  y la adyacencia de los vértices del grafo compartido, se obtiene:

$$\begin{array}{c} a \\ b \\ c \\ d \\ e \\ f \\ g \end{array} \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

## Solución del ejemplo 5.12

Es decir:

$$M_A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

## Solución del ejemplo 5.12

Verificando el resultado a través de la sentencia MGrafo:

**In[ ] :=**

```
grafo = Grafo[{{a, b}, {a, g}, {b, d}, {b, e}, {b, f},  
{c, a}, {c, d}, {d, e}, {e, f}, {f, g}}];  
MGrafo[grafo, {a, b, c, d, e, f, g}]  
MGrafo[grafo, {a, b, c, d, e, f, g}, table -> True]
```

Se obtienen la siguientes salidas:

## Solución del ejemplo 5.12

Out[ ] =

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

	a	b	c	d	e	f	g
a	0	1	1	0	0	0	1
b	1	0	0	1	1	1	0
c	1	0	0	1	0	0	0
d	0	1	1	0	1	0	0
e	0	1	0	1	0	1	0
f	0	1	0	0	1	0	1
g	1	0	0	0	0	1	0

## Solución del ejemplo 5.12

Aquí, nuevamente la matriz de adyacencia hallada es booleana. Esto ocurre pues el grafo del ejercicio no contiene aristas múltiples.

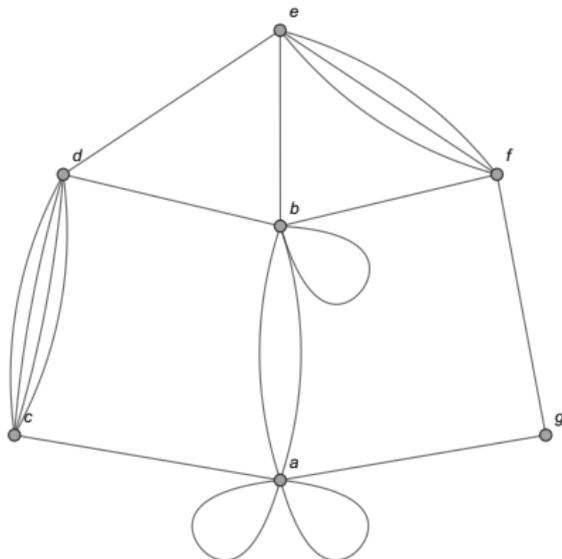


**Descargue un archivo**

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-107.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-107.zip)

## Example (5.13)

Encuentre una matriz de adyacencia que represente el grafo:



## Solución del ejemplo 5.13

En este ejemplo, el grafo del enunciado posee lados paralelos. Esto provoca que cualquier matriz de adyacencia que lo represente, no podrá ser una matriz booleana. Al recurrir a la instrucción `MGrafo` de **VilCretas**, se tiene:

**In[ ]** :=

```
grafo = Grafo[{{a, a}, {a, a}, {a, b}, {a, b}, {a, g},  
{b, b}, {b, d}, {b, e}, {b, f}, {c, a}, {c, d}, {c, d},  
{c, d}, {c, d}, {d, e}, {e, f}, {e, f}, {e, f}, {f, g}}];  
MGrafo[grafo, {a, b, c, d, e, f, g}, table -> True]
```

Muestra la siguiente salida:

## Solución del ejemplo 5.13

Out[ ] =

	a	b	c	d	e	f	g
a	2	2	1	0	0	0	1
b	2	1	0	1	1	1	0
c	1	0	0	4	0	0	0
d	0	1	4	0	1	0	0
e	0	1	0	1	0	3	0
f	0	1	0	0	3	0	1
g	1	0	0	0	0	1	0

Suponiendo  $M_A = (a_{ij})$ , se aprecia que  $a_{11} = 2$  por el doble lazo que se encuentra sobre el vértice  $a$ ,  $a_{12} = a_{21} = 2$  pues entre el vértice  $a$  y el nodo  $b$  hay 2 lados que los unen,  $a_{34} = a_{43} = 4$  dado que hay 4 aristas que relacionan los vértices  $c$  y  $d$  y,  $a_{56} = a_{65} = 3$  al existir 3 lados del grafo que unen los nodos  $e$  y  $f$ . El lazo que ocurre en  $b$  queda representado en  $M_A$  por medio de la entrada  $a_{22} = 1$ , ubicada en la diagonal principal de  $M_A$ .



## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-108.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-108.zip)

Al analizar cada una de las matrices de adyacencia  $M_A$  halladas en los ejemplos 17, 18 y 19, se desprenden una serie de propiedades fundamentales para este tipo de matriz cuando el grafo representado es no dirigido, a saber:

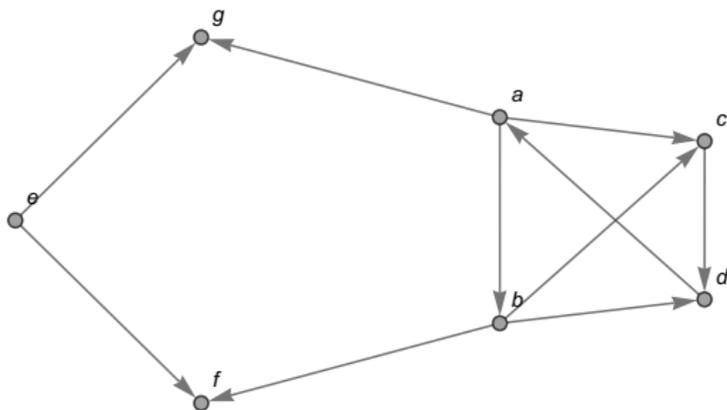
- $M_A$  siempre es una matriz simétrica, es decir,  $(M_A)^t = M_A$ . Esto ya se había mencionado en la página 10, pues todo grafo no dirigido es una relación binaria simétrica.
- Si la diagonal principal de  $M_A$  contiene entradas no nulas, el grafo posee lazos o ciclos.
- La suma de los elementos de cada una de las filas en  $M_A$ , da como resultado la valencia del nodo que corresponde a esa fila, siempre y cuando, en ese vértice no ocurra un ciclo.

## Nota

En un grafo dirigido, no necesariamente  $M_A$  cumplirá con la propiedad de simetría.

## Example (5.14)

Determine una matriz de adyacencia asociada al grafo:



## Solución del ejemplo 5.14

El grafo compartido es un digrafo. Una matriz de adyacencia se construye como ya se ha explicado en los ejercicios anteriores, con el detalle de considerar la adyacencia entre dos nodos  $x$  y  $y$ , estrictamente en ese orden, si existe en el grafo una flecha que inicia en el vértice  $x$  y finaliza en el nodo  $y$ . Luego:

$$\begin{array}{c} a \\ b \\ c \\ d \\ e \\ f \\ g \end{array} \begin{pmatrix} a & b & c & d & e & f & g \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

## Solución del ejemplo 5.14

Como consecuencia:

$$M_A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$M_A = (a_{ij})$  no es una matriz simétrica, por ejemplo,  $a_{12} = 1$  pues existe en el grafo una flecha dirigida de  $a$  a  $b$ , pero al no haber un lado dirigido de  $b$  a  $a$ , se tiene que  $a_{21} = 0$ . Solo este par de entradas ya comprueban que  $M_A$  no satisface la simetría.

## Solución del ejemplo 5.14

El comando MGrafo también procesa digrafos. En este ejercicio:

**In[ ] :=**

```
grafo = Grafo[{{a, b}, {a, c}, {a, g}, {b, c}, {b, d},  
{b, f}, {c, d}, {d, a}, {e, f}, {e, g}}, dirigido -> True];  
MGrafo[grafo, {a, b, c, d, e, f, g}]  
MGrafo[grafo, {a, b, c, d, e, f, g}, table -> True]
```

Se obtiene la siguiente salida:

## Solución del ejemplo 5.14

Out[ ] =

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

	a	b	c	d	e	f	g
a	0	1	1	0	0	0	1
b	0	0	1	1	0	1	0
c	0	0	0	1	0	0	0
d	1	0	0	0	0	0	0
e	0	0	0	0	0	1	1
f	0	0	0	0	0	0	0
g	0	0	0	0	0	0	0



## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-109.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-109.zip)

Al tener una matriz de adyacencia  $M_A$  de un grafo  $G$  se puede construir el grafo conociendo de previo cuáles son sus vértices. A continuación se presenta un ejemplo en esa dirección.

### Example (5.15)

Dadas las siguientes matrices de adyacencia  $A$  y  $B$  represente los grafos correspondientes, asumiendo el conjunto  $V$  de vértices como elementos consecutivos del abecedario iniciando en  $a$ .

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 2 & 4 & 1 \\ 0 & 2 & 0 & 1 & 0 \\ 1 & 4 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

## Solución del ejemplo 5.15

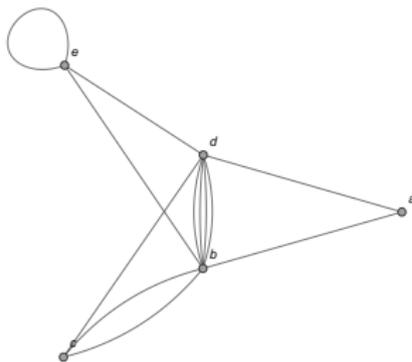
Las matrices  $A$  y  $B$  son simétricas lo que fácilmente es verificable calculando su transpuesta. Como  $A$  y  $B$  son arreglos bidimensionales simétricos esto quiere decir que los grafos  $G_A$  y  $G_B$ , representados por  $A$  y  $B$ , respectivamente, son no dirigidos. Cada entrada no nula en  $A$  y  $B$  denota la adyacencia entre dos vértices en el grafo  $G_A$  o  $G_B$  correspondiente, además, de mostrar la cantidad de lados que unen dichos nodos.

Al tomar la matriz  $A$  :

$$\begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} \begin{pmatrix} & a & b & c & d & e \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 2 & 4 & 1 \\ 0 & 2 & 0 & 1 & 0 \\ 1 & 4 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

## Solución del ejemplo 5.15

Se aprecia un lado que une el vértice  $a$  con el nodo  $b$ , una arista que une los nodos  $a$  y  $d$ , dos lados que relacionan los vértices  $b$  y  $c$ , cuatro aristas que unen los nodos  $b$  y  $d$ , un lado que relaciona los vértices  $b$  y  $e$ , una arista que une los nodos  $c$  y  $d$ , un lado del grafo que relaciona los vértices  $d$  y  $e$  y la existencia de un lazo sobre el nodo  $e$ . Por consiguiente,  $G_A$  toma la forma:



## Solución del ejemplo 5.15

La sentencia GrafoM del paquete **VilCretas** facilita la elaboración de un grafo conociendo una matriz de adyacencia, con la ventaja de poseer un algoritmo interno que normalmente, acomoda los vértices del grafo para formar una figura legible, es decir, fácil de leer para quien la visualiza. Si se utiliza GrafoM mediante la matriz  $B$  de este ejemplo, se tiene:

**In[ ]** :=

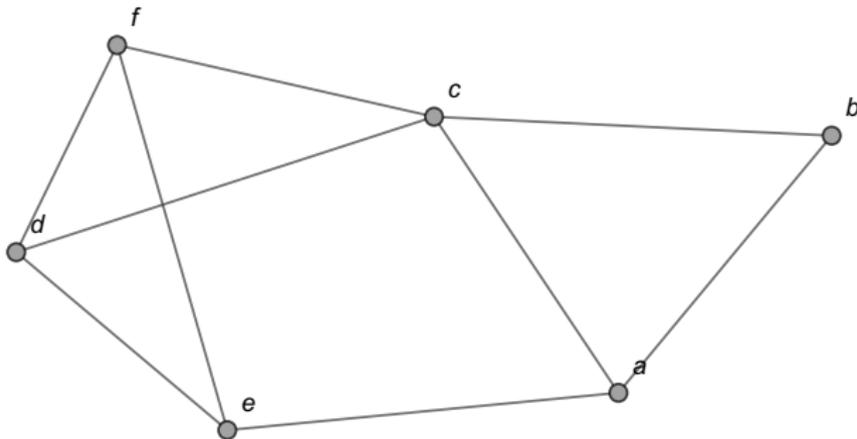
$$B = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix};$$

GrafoM[B, {a, b, c, d, e, f}]

## Solución del ejemplo 5.15

Se obtiene la siguiente salida:

**Out[ ] =**





## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-110.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-110.zip)



## Explicación en video

<https://youtu.be/cRz5VVDaXRQ>

## Teorema 5.2

La matriz de adyacencia de un grafo simple tiene otra notable aplicación. Consideremos el siguiente teorema.

### Theorem (5.2)

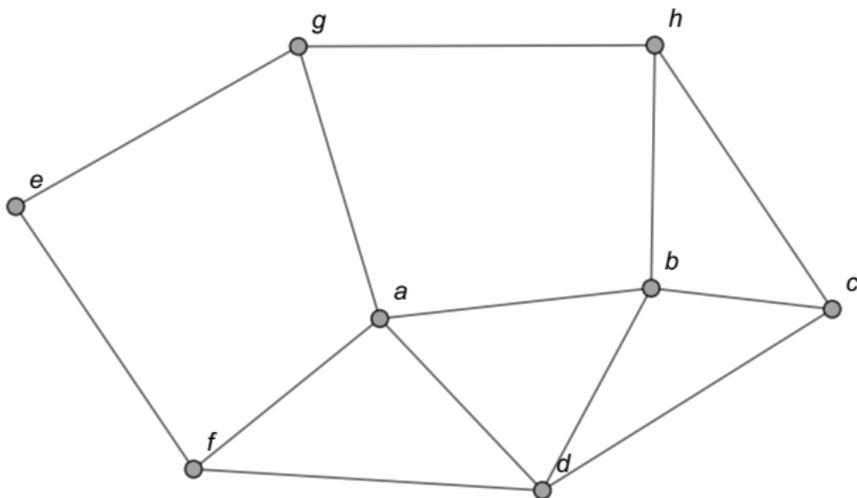
*Sea  $G = (V, E)$  un grafo simple,  $V = \{v_1, v_2, \dots, v_n\}$  y  $M_A$  una matriz de adyacencia de  $G$  en el orden del conjunto  $V$ .  $(M_A)^k = (c_{ij})$ , con  $k \in \mathbb{N}$  es una matriz donde la entrada  $c_{ij}$  representa el número de rutas con repetición de aristas de longitud  $k$ , del nodo  $v_i$  al nodo  $v_j$  en  $G$ .*

## Comentario sobre el teorema 22

El teorema 22 propone otra interesante propiedad para las matrices de adyacencia en un grafo simple (sin lazos y sin aristas múltiples). Al tomar la potencia  $k$ -ésima de una matriz de adyacencia del grafo se genera otra matriz donde sus entradas facilitan el número máximo de trayectorias de longitud  $k$  entre dos vértices cualesquiera. Se habla de una cantidad máxima de caminos pues el teorema admite repetición de aristas en las rutas, aspecto que contradice la definición 3. Por consiguiente, la potencia  $k$ -ésima de una matriz de adyacencia de un grafo simple no puede retornar el número exacto de trayectorias entre cualquier par de vértices, sino más bien, una cantidad máxima de rutas existentes de longitud  $k$ .

## Example (5.16)

¿Cuántas rutas como máximo, de longitud 4, posee el grafo compartido del nodo  $c$  al vértice  $e$ ?



## Solución del ejemplo 5.16

En este ejemplo, usando el teorema 22, se debe obtener una matriz de adyacencia del grafo simple dado y posteriormente elevar esa matriz a la 4. El comando `CantMRutas` de la librería **VilCretas** automatiza el proceso, pasando como argumento el grafo de interés. Veamos:

**In[ ]** :=

```
grafo = Grafo[{{a, b}, {a, f}, {a, g}, {b, c}, {b, d},  
{b, h}, {c, d}, {c, h}, {d, a}, {d, f}, {e, f}, {e, g},  
{g, h}}];
```

```
CantMRutas[grafo, 4]
```

Se obtiene la siguiente salida:

## Solución del ejemplo 5.16

Out[ ] =

	a	b	c	d	e	f	g	h
a	34	20	21	27	16	13	8	23
b	20	34	23	26	5	23	22	15
c	21	23	21	19	6	15	13	14
d	27	26	19	32	12	16	14	22
e	16	5	6	12	10	3	1	11
f	13	23	15	16	3	20	18	7
g	8	22	13	14	1	18	19	5
h	23	15	14	22	11	7	5	20

## Solución del ejemplo 5.16

CantMRutas construye internamente una matriz de adyacencia  $M_A$  para el grafo y posteriormente la eleva a la 4 mostrando como salida el resultado  $(M_A)^4$ , cálculo bastante tedioso si se efectuara manualmente. Las entradas  $c_{35} = c_{53} = 6$  de  $(M_A)^4 = (c_{ij})$ , retornan la cantidad máxima de caminos de longitud 4 que se pueden construir en el grafo, del nodo  $c$  al vértice  $e$ . De hecho, en este ejercicio en particular, existen exactamente 6 caminos diferentes de longitud 4 entre los vértices  $c$  y  $e$ . Empleando los comandos CantRutas y PesoRuta se pueden hallar explícitamente estas trayectorias y su representación sobre el grafo:

**In[ ] :=**

```
Lista = Select[CantRutas[grafo, c, e, rutas -> True][[2]],
Length[#] == 4 &]
Table[PesoRuta[grafo, i, ruta -> True], {i, Lista}]
```

## Solución del ejemplo 5.16

**Out[ ] =**

```

{{{c, d}, {d, a}, {a, g}, {g, e}},
 {{c, d}, {d, a}, {a, f}, {f, e}},
 {{c, b}, {b, h}, {h, g}, {g, e}},
 {{c, b}, {b, d}, {d, f}, {f, e}},
 {{c, b}, {b, a}, {a, g}, {g, e}},
 {{c, b}, {b, a}, {a, f}, {f, e}}}
:

```

En el **Out[ ]** no se muestra la salida del `Table` que corresponde a un vector de grafos, con las 6 rutas dibujadas en cada caso. La figura 4 contiene dichos caminos devueltos por la sentencia `PesoRuta`.

## Solución del ejemplo 5.16

La instrucción `CantMRutas` posee el atributo `all -> True` capaz de generar una animación con todas las potencias  $k$ -ésimas de una matriz de adyacencia de un grafo simple, comenzando con la potencia  $k$  igual a 1 y hasta un valor indicado por el usuario. En este ejemplo:

```
In[ ] :=
```

```
CantMRutas[grafo, 4, all -> True]
```

## Solución del ejemplo 5.16

Out[ ] =

The screenshot shows a software interface with a slider labeled 'k' set to the value 3. Below the slider is a table representing an adjacency matrix. The table has 8 rows and 8 columns, labeled 'a' through 'h' on both the top and left sides. The values in the cells are as follows:

	a	b	c	d	e	f	g	h
a	4	10	5	8	1	8	8	3
b	10	6	7	9	4	3	2	8
c	5	7	4	8	2	3	3	6
d	8	9	8	6	2	7	5	4
e	1	4	2	2	0	5	5	0
f	8	3	3	7	5	2	1	5
g	8	2	3	5	5	1	0	6
h	3	8	6	4	0	5	6	2

El deslizador  $k$  recorre las potencias  $k$ -ésimas de una matriz de adyacencia, con  $k = 1$  y hasta  $k = 4$ .

## Solución del ejemplo 5.16

Un documento con un formato computable con una funcionalidad similar al comando CantMRutas, se puede descargar del siguiente enlace.



**Descargue un archivo**

[https://www.escinf.una.ac.cr/discretas/Archivos/Grafos/  
File-111.zip](https://www.escinf.una.ac.cr/discretas/Archivos/Grafos/File-111.zip)

## Nota

Se insiste al estudiante que las entradas de la potencia  $k$ -ésima de una matriz de adyacencia en un grafo simple, no siempre brindarán el número exacto de trayectorias de longitud  $k$  entre un par de nodos cualesquiera. Algunas veces, el valor de las entradas en la matriz potencia  $k$ -ésima, será mayor que la cantidad real de trayectorias existentes.



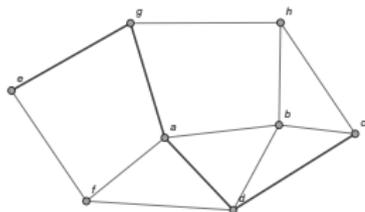
## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/CDFs/  
Matriz\\_de\\_adyacencia\\_potencias.cdf.zip](https://www.esconf.una.ac.cr/discretas/Archivos/CDFs/Matriz_de_adyacencia_potencias.cdf.zip)

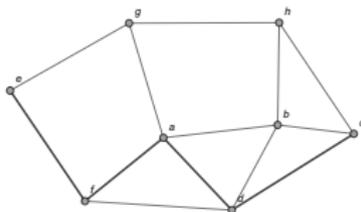


## Explicación en video

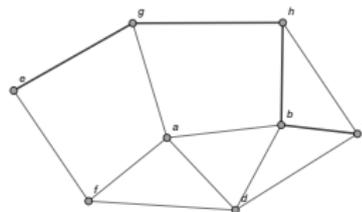
<https://youtu.be/IHBhuERIpSY>



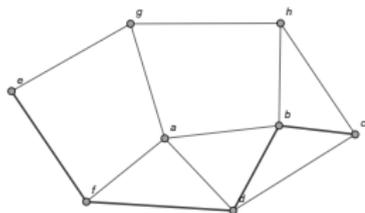
(a) Camino 1



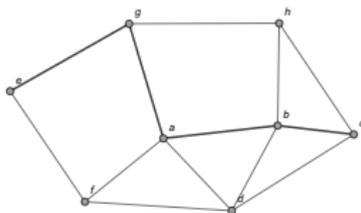
(b) Camino 2



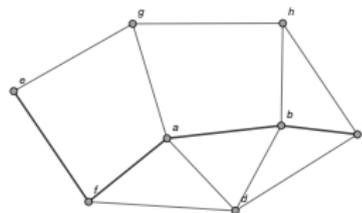
(c) Camino 3



(d) Camino 4



(e) Camino 5



(f) Camino 6

Figura: Rutas de longitud 4 del nodo c al vértice e sobre el grafo del ejemplo 23

### Example (5.17)

Sea  $A$  una matriz de adyacencia del grafo bipartito completo  $K_{10,10}$ . Verifique que todas las entradas no nulas de  $A^k$ , son iguales entre sí,  $1 \leq k \leq 10$ ,  $k \in \mathbb{N}$ . Conjeture una fórmula que calcule el valor constante de las entradas distintas de cero en  $A^k$ ,  $\forall k$ ,  $k \in \mathbb{N}$ .

## Solución del ejemplo 5.17

La primera parte de este ejercicio es resoluble recurriendo a la sentencia `CantMRutas` y su opción `all -> True`:

**In[ ] :=**

```
grafo = GrafoBipartitoCompleto[10, 10];  
CantMRutas[grafo, 10, all -> True]
```

Se obtiene la siguiente salida:

## Solución del ejemplo 5.17

Out[ ] =

[-] [▶] [◀] [↕] [↔] [→]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
3	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
4	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
5	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
6	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
7	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
8	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
9	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
10	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
12	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
13	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
14	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
15	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
16	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
17	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
18	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
19	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
20	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0

## Solución del ejemplo 5.17

Al utilizar el deslizador  $k$  de la animación, se obtiene:

Valor de $k$	Valor constante de las entradas no nulas
1	$1 = 10^0$
2	$10 = 10^1$
3	$100 = 10^2$
4	$1000 = 10^3$
5	$10000 = 10^4$
6	$100000 = 10^5$
7	$1000000 = 10^6$
8	$10000000 = 10^7$
9	$100000000 = 10^8$
10	$1000000000 = 10^9$

## Solución del ejemplo 5.17

Observando el comportamiento de las entradas no nulas, se conjetura que el valor constante distinto de cero en la matriz  $A^k$  corresponde a  $10^{k-1}$ ,  $\forall k, k \in \mathbb{N}$ .



Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-112.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-112.zip)

## Definición 5.6

Cuando un grafo es simple y ponderado tiene asociada otro tipo de matriz de adyacencia llamada “matriz de adyacencia de pesos”. Consideremos la definición que prosigue.

### Definition (5.6)

Sea  $G = (V, E)$  un grafo simple y ponderado con  $V = \{v_1, v_2, \dots, v_n\}$  y  $E = \{e_1, e_2, \dots, e_p\}$ . Se llama matriz de adyacencia de pesos de  $G$  en el orden de los elementos de  $V$ , a la matriz denotada  $M_{AP}$ ,  $M_{AP} = (a_{ij})$  de tamaño  $n$  por  $n$ , tal que  $a_{ij} = 0$  si los vértices  $v_i$  y  $v_j$  no son adyacentes en  $G$  y  $a_{ij} = k$ ,  $k \neq 0$ ,  $k \in \mathbb{R}$  si  $k$  representa el peso o la ponderación asignada a la arista que une los vértices  $v_i$  y  $v_j$ .

## Comentario sobre la definición 25

Una notación bastante usual vinculada a un grafo ponderado reside en especificar una matriz de adyacencia de pesos que lo represente. En esta dirección el paquete **VilCretas** integra el comando `GrafoMP` que al recibir como argumento una matriz de adyacencia de pesos y la lista de vértices, permite construir el grafo. Al igual que la instrucción `GrafoM`, `GrafoMP` contiene un algoritmo que facilita la generación del grafo de una forma normalmente legible.

## Comentario sobre la definición 25

Por ejemplo, supóngase que la siguiente matriz es un arreglo bidimensional de adyacencia de ponderaciones de un grafo simple:

$$M_{AP} = \begin{pmatrix} 0 & 2,3 & 0 & 1 & \frac{5}{2} & 0 & 0 & 0 \\ 2,3 & 0 & -1 & 0 & \pi & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & -2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & -5 \\ \frac{5}{2} & \pi & -2 & 0 & 0 & 0 & -3 & \sqrt{2} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -3 & 0 & 0 & 0 \\ 0 & 0 & 1 & -5 & \sqrt{2} & 1 & 0 & 0 \end{pmatrix} \quad (3)$$

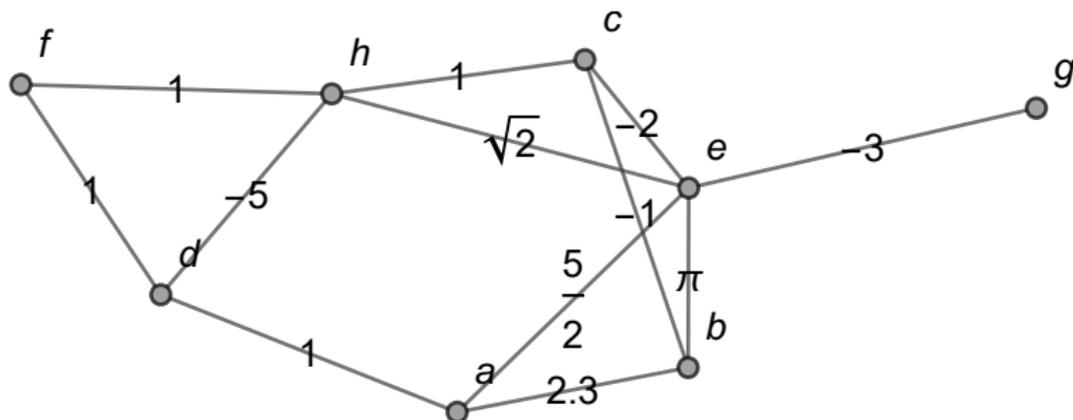
## Comentario sobre la definición 25

Luego, al usar el comando GrafoMP en *Wolfram*:

**In[ ] :=**

GrafoMP[A, {a, b, c, d, e, f, g, h}, mostrarpesos -> True]

**Out[ ] =**



## Comentario sobre la definición 25

Siendo  $A$  la matriz  $M_{AP}$  presentada en 3. Si se omite del `ln[ ]` la opción `mostrarpesos -> True`, la sentencia `GrafoMP` siempre construirá el grafo respectivo, pero sin mostrar las ponderaciones sobre sus lados.

# Comentario sobre la definición 25

## Nota

Más adelante en este libro se recurrirá a la instrucción `GrafoMP` para generar grafos ponderados, por lo que es esencial que el lector tenga muy presente su funcionalidad.

# Comentario sobre la definición 25



## Descargue un archivo

<https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-113.zip>



## Explicación en video

<https://youtu.be/RXc1y7JEHLo>

## Definición 5.7

Los grafos gozan de otro tipo de representación bidimensional denominada matriz de incidencia.

### Definition (5.7)

Sea  $G = (V, E)$  un grafo no dirigido con  $V = \{v_1, v_2, \dots, v_n\}$  y  $E = \{e_1, e_2, \dots, e_p\}$ . Se llama matriz de incidencia de  $G$  en el orden de los elementos de  $V$  y  $E$ , a la matriz denotada  $M_I$ ,  $M_I = (a_{ij})$  de tamaño  $n$  por  $p$ , tal que:

$$a_{ij} = \begin{cases} 1 & \text{si la arista } e_j \text{ es incidente sobre el v\u00e9rtice } v_i \\ 0 & \text{en caso contrario} \end{cases}$$

## Comentario sobre la definición 26

Una matriz de incidencia de un grafo no dirigido  $G = (V, E)$  estudia la incidencia de cada uno de sus lados sobre sus vértices (de allí su nombre), por este motivo en la definición 26, se aprecia que este tipo de matriz toma para su construcción, en sus filas los nodos y en sus columnas las aristas del grafo. Como veremos, una matriz de incidencia se suele elaborar por columnas y no por filas.

Cuando un grafo  $G$  tiene lados paralelos, por cada arista múltiple, se añade una columna en la matriz de incidencia de  $G$ , correspondiente. Lo anterior significa que toda matriz de incidencia será una matriz booleana (sus entradas son ceros y unos) sea el grafo simple o no.

## Comentario sobre la definición 26

En un grafo  $G$  la suma de las entradas de la fila  $i$ -ésima de una matriz de incidencia, da como resultado la valencia o el grado del vértice  $v_i$ , si en  $v_i$ , no existe un ciclo. Si el grafo tiene lados múltiples esta propiedad se continúa satisfaciendo.

Finalmente, cabe destacar que un grafo  $G$  puede poseer varias matrices de incidencia dependiendo del orden establecido en su elaboración para los conjuntos  $V$  y  $E$ .

### Example (5.18)

Halle una matriz de incidencia sobre grafo del ejemplo 17.

## Solución del ejemplo 5.18

Al elegir el orden del abecedario se partirá de  $V = \{a, b, c, d, e, f\}$  y  $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$ , con cada  $e_j$ ,  $1 \leq j \leq 8$ ,  $j \in \mathbb{N}$ , tal y como se aprecia en la figura 5. El tamaño de cualquier matriz de incidencia sobre este grafo es  $6 \times 8$  :

$$\begin{array}{c}
 a \\
 b \\
 c \\
 d \\
 e \\
 f
 \end{array}
 \begin{pmatrix}
 e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \\
 \square & \square \\
 \square & \square \\
 \square & \square \\
 \square & \square \\
 \square & \square \\
 \square & \square
 \end{pmatrix}$$

## Solución del ejemplo 5.18

Ahora, se construirá la matriz por columnas. La arista  $e_1$  es incidente sobre los vértices  $a$  y  $d$  entonces en la primera columna se coloca un 1 en la fila donde aparece el nodo  $a$  y en la fila donde se encuentra el vértice  $d$ , en las otras entradas se incorpora un 0 :

$$\begin{array}{c}
 a \\
 b \\
 c \\
 d \\
 e \\
 f
 \end{array}
 \begin{pmatrix}
 e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \\
 1 & \square \\
 0 & \square \\
 0 & \square \\
 1 & \square \\
 0 & \square \\
 0 & \square & \square & \square & \square & \square & \square & \square
 \end{pmatrix}$$

## Solución del ejemplo 5.18

El lado  $e_2$  es incidente sobre los vértices  $a$  y  $e$ , por lo que, en la segunda columna se pone un 1 en la fila de  $a$  y en la fila de  $e$ , en las demás, se coloca un 0 :

$$\begin{array}{c}
 \\
 a \\
 b \\
 c \\
 d \\
 e \\
 f
 \end{array}
 \begin{pmatrix}
 e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \\
 1 & 1 & \square & \square & \square & \square & \square & \square \\
 0 & 0 & \square & \square & \square & \square & \square & \square \\
 0 & 0 & \square & \square & \square & \square & \square & \square \\
 1 & 0 & \square & \square & \square & \square & \square & \square \\
 0 & 1 & \square & \square & \square & \square & \square & \square \\
 0 & 0 & \square & \square & \square & \square & \square & \square
 \end{pmatrix}$$

## Solución del ejemplo 5.18

La arista  $e_3$  incide a los nodos  $a$  y  $f$ , en consecuencia, se pone en esa columna un 1 en la fila donde se encuentra la  $a$  y en la fila donde aparece la  $f$  y un 0 en las otras entradas:

$$\begin{array}{c}
 \\
 a \\
 b \\
 c \\
 d \\
 e \\
 f
 \end{array}
 \begin{pmatrix}
 e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \\
 1 & 1 & 1 & \square & \square & \square & \square & \square \\
 0 & 0 & 0 & \square & \square & \square & \square & \square \\
 0 & 0 & 0 & \square & \square & \square & \square & \square \\
 1 & 0 & 0 & \square & \square & \square & \square & \square \\
 0 & 1 & 0 & \square & \square & \square & \square & \square \\
 0 & 0 & 1 & \square & \square & \square & \square & \square
 \end{pmatrix}$$

## Solución del ejemplo 5.18

El lado  $e_4$  es incidente sobre los vértices  $b$  y  $c$ , luego:

$$\begin{array}{c}
 \\
 a \\
 b \\
 c \\
 d \\
 e \\
 f
 \end{array}
 \begin{pmatrix}
 e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \\
 1 & 1 & 1 & 0 & \square & \square & \square & \square \\
 0 & 0 & 0 & 1 & \square & \square & \square & \square \\
 0 & 0 & 0 & 1 & \square & \square & \square & \square \\
 1 & 0 & 0 & 0 & \square & \square & \square & \square \\
 0 & 1 & 0 & 0 & \square & \square & \square & \square \\
 0 & 0 & 1 & 0 & \square & \square & \square & \square
 \end{pmatrix}$$

## Solución del ejemplo 5.18

La arista  $e_5$  incide a los nodos  $b$  y  $f$ , en consecuencia:

$$\begin{array}{c} a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{pmatrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \\ 1 & 1 & 1 & 0 & 0 & \square & \square & \square \\ 0 & 0 & 0 & 1 & 1 & \square & \square & \square \\ 0 & 0 & 0 & 1 & 0 & \square & \square & \square \\ 1 & 0 & 0 & 0 & 0 & \square & \square & \square \\ 0 & 1 & 0 & 0 & 0 & \square & \square & \square \\ 0 & 0 & 1 & 0 & 1 & \square & \square & \square \end{pmatrix}$$

## Solución del ejemplo 5.18

El lado  $e_6$  es incidente sobre los vértices  $c$  y  $e$ . Se debe colocar un 1 en la columna de  $e_6$ , al considerar las filas de los nodos  $c$  y  $e$ , un 0 en las otras entradas:

$$\begin{array}{c}
 \\
 a \\
 b \\
 c \\
 d \\
 e \\
 f
 \end{array}
 \begin{pmatrix}
 e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \\
 1 & 1 & 1 & 0 & 0 & 0 & \square & \square \\
 0 & 0 & 0 & 1 & 1 & 0 & \square & \square \\
 0 & 0 & 0 & 1 & 0 & 1 & \square & \square \\
 1 & 0 & 0 & 0 & 0 & 0 & \square & \square \\
 0 & 1 & 0 & 0 & 0 & 1 & \square & \square \\
 0 & 0 & 1 & 0 & 1 & 0 & \square & \square
 \end{pmatrix}$$

## Solución del ejemplo 5.18

Las arista  $e_7$  incide a los vértices  $c$  y  $f$  :

$$\begin{array}{c}
 \\
 a \\
 b \\
 c \\
 d \\
 e \\
 f
 \end{array}
 \begin{pmatrix}
 e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \\
 1 & 1 & 1 & 0 & 0 & 0 & 0 & \square \\
 0 & 0 & 0 & 1 & 1 & 0 & 0 & \square \\
 0 & 0 & 0 & 1 & 0 & 1 & 1 & \square \\
 1 & 0 & 0 & 0 & 0 & 0 & 0 & \square \\
 0 & 1 & 0 & 0 & 0 & 1 & 0 & \square \\
 0 & 0 & 1 & 0 & 1 & 0 & 1 & \square
 \end{pmatrix}$$

## Solución del ejemplo 5.18

Finalmente, el lado  $e_8$  del grafo es incidente sobre los vértices  $d$  y  $e$  :

$$\begin{array}{c}
 a \\
 b \\
 c \\
 d \\
 e \\
 f
 \end{array}
 \begin{pmatrix}
 e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \\
 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0
 \end{pmatrix}$$

## Solución del ejemplo 5.18

Se concluye que una matriz de incidencia del grafo es:

$$M_I = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

## Solución del ejemplo 5.18

El comando `MIGrafo` de la librería **VilCretas** automatiza la elaboración de este tipo de matriz, pasando como parámetros el grafo y un orden sobre sus vértices. También, la instrucción contiene las opciones `edgesgraph`  $\rightarrow$  `L` y `table`  $\rightarrow$  `True`. La primera, da la posibilidad de especificar un orden para las aristas en la construcción de la matriz de incidencia (`L` es la lista de lados en el orden requerido) y la segunda, etiqueta las filas con los nodos y las columnas con los lados del grafo, de acuerdo al orden utilizado en la elaboración de la matriz de incidencia respectiva. En este ejemplo:

`In[ ] :=`

```
grafo = Grafo[{{a, d}, {a, e}, {a, f}, {b, c}, {b, f},  
{c, e}, {c, f}, {d, e}}];  
MIGrafo[grafo, {a, b, c, d, e, f}]  
MIGrafo[grafo, {a, b, c, d, e, f}, table  $\rightarrow$  True]
```

## Solución del ejemplo 5.18

Out[ ] =

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

	$a \bullet \bullet d$	$a \bullet \bullet e$	$a \bullet \bullet f$	$b \bullet \bullet c$	$b \bullet \bullet f$	$c \bullet \bullet e$	$c \bullet \bullet f$	$d \bullet \bullet e$
a	1	1	1	0	0	0	0	0
b	0	0	0	1	1	0	0	0
c	0	0	0	1	0	1	1	0
d	1	0	0	0	0	0	0	1
e	0	1	0	0	0	1	0	1
f	0	0	1	0	1	0	1	0

## Solución del ejemplo 5.18

El orden de los lados del grafo mostrado por *Mathematica* mediante el atributo `table -> True` es definido por el software en función del orden de ingreso de las aristas en el comando `Grafo`.

Si se usa un orden distinto para los vértices, o bien, para los lados de un grafo, algunas veces, se obtendrá una matriz de incidencia diferente. Se insiste al lector que un arreglo bidimensional de incidencia no necesariamente es único. Veamos un ejemplo de ello, en este ejercicio:

**In[ ] :=**

```
MIGrafo[grafo, {b, a, e, f, c, d}]
```

Se obtiene la siguiente salida:

Solución del ejemplo 5.18

**Out[ ] =**

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

En la salida se observa una matriz de incidencia distinta para el grafo, al cambiar el orden de los nodos.



## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-114.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-114.zip)



## Explicación en video

<https://youtu.be/37jeS7P-b8g>

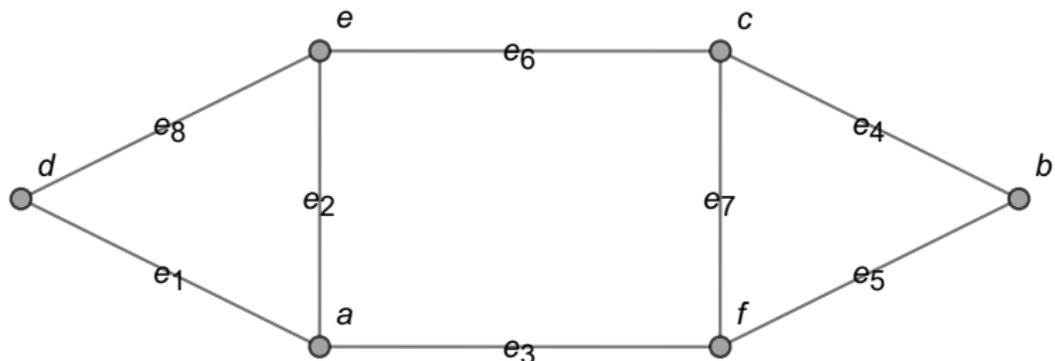


Figura: Orden de las aristas en el grafo del ejemplo 27

### Example (5.19)

En el grafo del ejemplo 18 encuentre una matriz de incidencia.

## Solución del ejemplo 5.19

Al analizar la incidencia de los lados del grafo del ejemplo 18 y recurrir al orden  $V = \{a, b, c, d, e, f, g\}$  y  $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}\}$ , con  $e_j$ ,  $1 \leq j \leq 10$ ,  $j \in \mathbb{N}$ , según lo mostrado en la figura 6, se infiere la siguiente matriz de incidencia:

$$\begin{array}{c}
 \\
 a \\
 b \\
 c \\
 d \\
 e \\
 f \\
 g
 \end{array}
 \begin{pmatrix}
 e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} \\
 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{pmatrix}$$

## Solución del ejemplo 5.19

Es decir:

$$M_I = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

## Solución del ejemplo 5.19

MIGrafo nos permite verificar la matriz anterior:

**In[ ]** :=

```
grafo = Grafo[{{a, b}, {a, g}, {b, d}, {b, e}, {b, f},  
{c, a}, {c, d}, {d, e}, {e, f}, {f, g}}];  
MIGrafo[grafo, {a, b, c, d, e, f, g}]  
MIGrafo[grafo, {a, b, c, d, e, f, g}, table -> True]
```

Se obtiene la siguiente salida:

## Solución del ejemplo 5.19

Out[ ] =

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

	$a \leftrightarrow b$	$a \leftrightarrow g$	$b \leftrightarrow d$	$b \leftrightarrow e$	$b \leftrightarrow f$	$c \leftrightarrow a$	$c \leftrightarrow d$	$d \leftrightarrow e$	$e \leftrightarrow f$	$f \leftrightarrow g$
a	1	1	0	0	0	1	0	0	0	0
b	1	0	1	1	1	0	0	0	0	0
c	0	0	0	0	0	1	1	0	0	0
d	0	0	1	0	0	0	1	1	0	0
e	0	0	0	1	0	0	0	1	1	0
f	0	0	0	0	1	0	0	0	1	1
g	0	1	0	0	0	0	0	0	0	1



## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-115.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-115.zip)

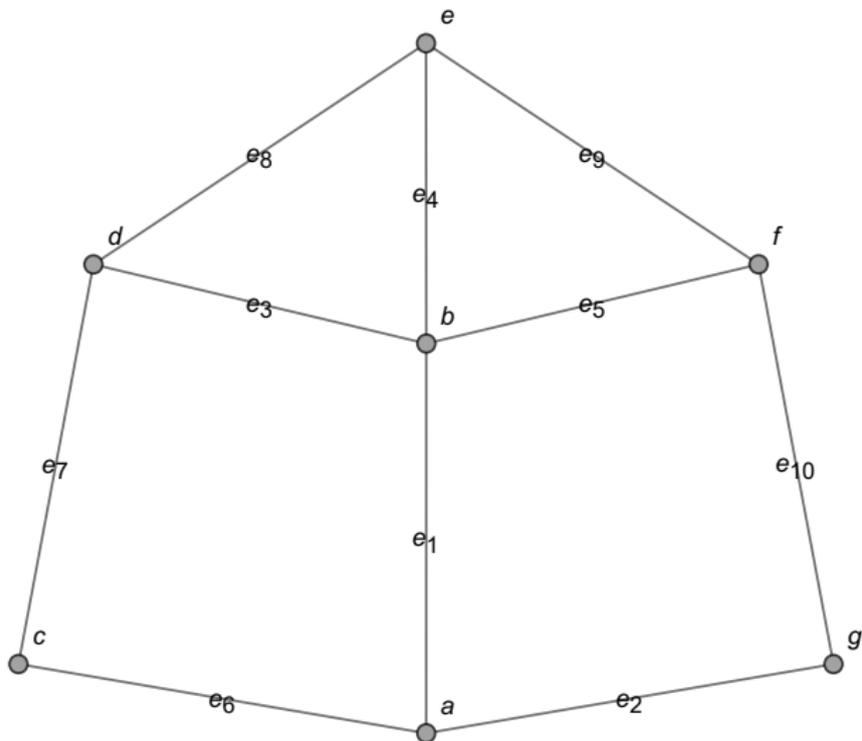


Figura: Orden de los lados en el grafo del ejemplo 28

### Example (5.20)

Determine una matriz de incidencia para el grafo del ejemplo 19.

## Solución del ejemplo 5.20

Usando el comando MIGrafo de **VilCretas**, se tiene:

**In[ ] :=**

```
grafo = Grafo[{{a, a}, {a, a}, {a, b}, {a, b}, {a, g},
{b, b}, {b, d}, {b, e}, {b, f}, {c, a}, {c, d}, {c, d},
{c, d}, {c, d}, {d, e}, {e, f}, {e, f}, {e, f}, {f, g}}];
MIGrafo[grafo, {a, b, c, d, e, f, g}]
```

**Out[ ] =**

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

## Solución del ejemplo 5.20

El grafo del ejemplo 19 contiene lazos y aristas múltiples, pese a ello, la matriz de incidencia encontrada es una matriz booleana, de hecho, cualquier otra matriz de incidencia continuará satisfaciendo esta condición. También es observable en la salida, que en la primera, segunda y sexta columna de la matriz de incidencia, solo hay un uno, no una pareja de unos, esto es debido a la presencia de tres ciclos o lazos en el grafo correspondiente.



**Descargue un archivo**

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-116.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-116.zip)

La definición 26 es generalizable a un digrafo. La incidencia de un lado en un grafo dirigido requiere señalar cuál vértice constituye el nodo de partida de la flecha que corresponde a la arista y cuál vértice es el nodo de llegada. En este sentido, una flecha que inicia en un vértice  $x$  y finaliza en un nodo  $y$ , se muestra en una matriz de incidencia del digrafo, colocando en la columna respectiva, un  $-1$  en la fila que corresponde al vértice  $x$  y un  $1$  en la fila asociada con el nodo  $y$ . Bajo este principio, cualquier matriz de incidencia de un grafo dirigido tendrá tres tipos de entradas:  $-1$ ,  $0$  y  $1$ .

### Example (5.21)

Halle una matriz de incidencia sobre el digrafo del ejemplo 20.

## Solución del ejemplo 5.21

La sentencia MIGrafo funciona también, con grafos dirigidos:

**In[ ] :=**

```
grafo = Grafo[{{a, b}, {a, c}, {a, g}, {b, c}, {b, d},
{b, f}, {c, d}, {d, a}, {e, f}, {e, g}}, dirigido -> True];
MIGrafo[grafo, {a, b, c, d, e, f, g}, table -> True]
```

**Out[ ] =**

	$a \bullet \rightarrow b$	$a \bullet \rightarrow c$	$a \bullet \rightarrow g$	$b \bullet \rightarrow c$	$b \bullet \rightarrow d$	$b \bullet \rightarrow f$	$c \bullet \rightarrow d$	$d \bullet \rightarrow a$	$e \bullet \rightarrow f$	$e \bullet \rightarrow g$
a	-1	-1	-1	0	0	0	0	1	0	0
b	1	0	0	-1	-1	-1	0	0	0	0
c	0	1	0	1	0	0	-1	0	0	0
d	0	0	0	0	1	0	1	-1	0	0
e	0	0	0	0	0	0	0	0	-1	-1
f	0	0	0	0	0	1	0	0	1	0
g	0	0	1	0	0	0	0	0	0	1

## Solución del ejemplo 5.21

En el **Out[ ]** se aprecia en cada columna de la matriz de incidencia retornada, la colocación de un  $-1$  en el nodo de partida del lado dirigido correspondiente del grafo y de un  $1$  en el vértice de llegada.



Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-117.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-117.zip)

# Comando GrafoMI

Análogamente a lo establecido anteriormente, dada una matriz de incidencia de un grafo  $G$  es posible generar a  $G$ . La librería **VilCretas** nos dota con el comando GrafoMI, que efectúa este procedimiento sobre un grafo.



Explicación en video

<https://youtu.be/BrfG1dtF0dw>

## Example (5.22)

Represente el grafo dado por la matriz de incidencia  $A$  compartida, en el orden de sus nodos  $V = \{a, b, c, d, e, f\}$ .

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

## Solución del ejemplo 5.22

Al utilizar la sentencia GrafoMI, se obtiene en *Mathematica*:

**In[ ] :=**

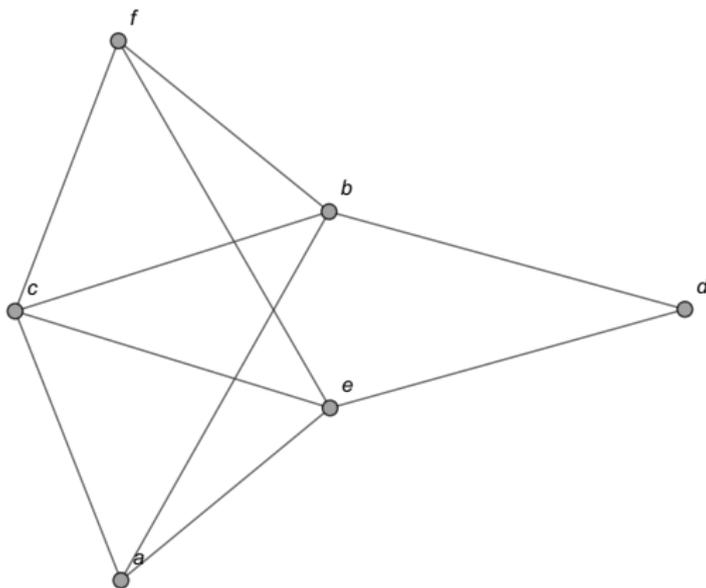
$$A = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix};$$

GrafoMI[A, {a, b, c, d, e, f}]

Se obtiene la siguiente salida:

## Solución del ejemplo 5.22

Out[ ] =





## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-118.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-118.zip)

- La sección que continúa, trata el estudio de los circuitos sobre un grafo. Los circuitos constituyen una unidad de conocimiento esencial por sus distintas aplicaciones en computación u otros campos. De allí que se dedique un apartado exclusivo para desarrollar el tema.

# Circuitos en un grafo

Prof. Enrique Vílchez Quesada

Universidad Nacional de Costa Rica

## Definición 5.8

Iniciaremos con el concepto de circuito.

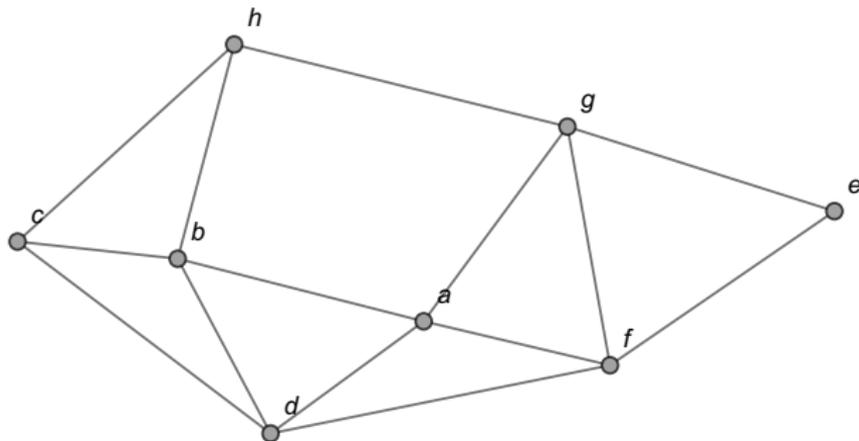
### Definition (5.8)

Sea  $G = (V, E)$  un grafo. A cualquier ruta sobre  $G$  que inicie y finalice en el mismo vértice se le llama circuito. Si un circuito no repite nodos se dice que es simple. Si un circuito contiene todas las aristas de  $G$  se denomina circuito de *Euler*. Si un circuito contiene todos los vértices de  $G$  sin repetición de nodos, se llama circuito de *Hamilton*. Una ruta que incluye todos los lados de  $G$  sin ser un circuito, se define como un camino de *Euler* o trayectoria euleriana. Cualquier trayectoria que no sea un circuito y que contenga una única vez todos los vértices de  $G$ , se llama ruta de *Hamilton* o camino hamiltoniano.

## Comentario sobre la definición 32

Un lazo sobre un grafo que lo contenga es en sí mismo un circuito (de longitud 1) de acuerdo con la definición 32.

En la figura 7 se muestran algunos de ejemplos de circuitos y una ruta euleriana sobre el grafo:



## Comentario sobre la definición 32

A saber:

- En la subfigura 7a se aprecia el circuito  $c \rightarrow g \rightarrow e \rightarrow f \rightarrow a \rightarrow b \rightarrow c$ , que corresponde a un circuito simple pues no se repiten nodos con excepción, desde luego, del vértice  $c$  de partida y de llegada.
- En la subfigura 7b se comparte el circuito  $d \rightarrow a \rightarrow f \rightarrow g \rightarrow a \rightarrow b \rightarrow c \rightarrow d$ , siendo éste un circuito no simple al repetir el nodo  $a$ .
- En la subfigura 7c se muestra al lector una ruta de *Euler* vinculada con el trayecto  $c \rightarrow b \rightarrow a \rightarrow f \rightarrow d \rightarrow b \rightarrow h \rightarrow g \rightarrow e \rightarrow f \rightarrow g \rightarrow a \rightarrow d \rightarrow c \rightarrow h$ . Se llama ruta euleriana, pues se pasa una única vez por cada una de las aristas del grafo. Se considera una ruta y no un circuito de *Euler*, pues el vértice de partida ( $c$ ) es distinto del nodo de llegada ( $h$ ).

## Comentario sobre la definición 32

- En la subfigura 7d se ejemplifica un circuito hamiltoniano sobre el grafo. Éste corresponde al recorrido  $a \rightarrow e \rightarrow g \rightarrow h \rightarrow c \rightarrow d \rightarrow b \rightarrow a$ . El camino descrito se denomina circuito de *Hamilton* pues es un circuito donde se pasa una única vez por cada uno de los vértices del grafo.



**Descargue un archivo**

[https://www.escinf.una.ac.cr/discretas/Archivos/Grafos/  
File-119.zip](https://www.escinf.una.ac.cr/discretas/Archivos/Grafos/File-119.zip)



# Problema de los siete puentes de Königsberg

En la historia de la matemática se considera a *Leonhard Euler* el padre de la teoría de grafos. La razón de ello obedece a una publicación que *Euler* realizó en el año de 1736, donde da respuesta a un famoso problema conocido como “el paso de los siete puentes de *Königsberg*”. Siendo *Leonhard Euler* un genio de su época, se le planteó la siguiente interrogante: por la ciudad de *Königsberg* (en la actualidad *Kaliningrado, Rusia*) pasa el río *Pregel*. Sobre él, se encuentran dos islas comunicadas entre sí y con las riberas u orillas, por siete puentes. ¿Es posible realizar un recorrido circuitual que transite una única vez por cada puente? Esta situación se representa con mayor claridad en la figura 8, que implícitamente forma un grafo cuyos vértices son las islas y las riberas, mientras que sus aristas están constituidas por los puentes respectivos. A razón de este problema, como ya se expuso en la definición 32, en la literatura se le conoce como circuito de *Euler* a cualquier trayectoria que recorra una única vez todos los lados de un grafo.

# Problema de los siete puentes de Königsberg

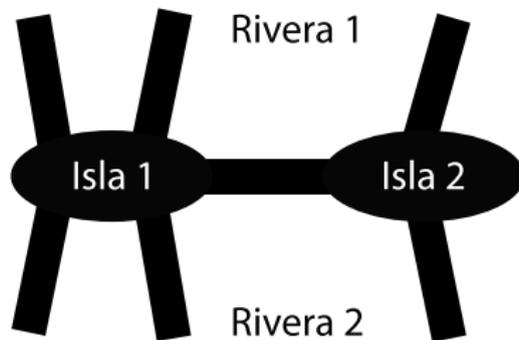


Figura: Problema de los siete puentes de *Königsberg*

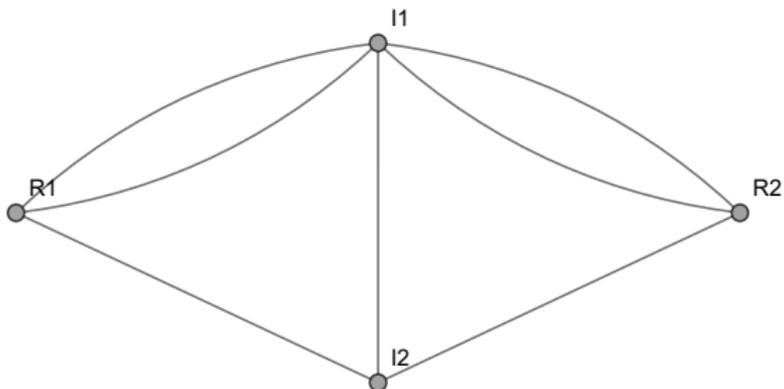
# Problema de los siete puentes de Königsberg

En *Wolfram Mathematica* el grafo vinculado a los siete puentes de *Königsberg* se puede generar usando la instrucción `GrafoKönigsberg` de la librería **VilCretas**:

**In[ ] :=**

`GrafoKönigsberg[]`

**Out[ ] =**



# Problema de los siete puentes de Königsberg

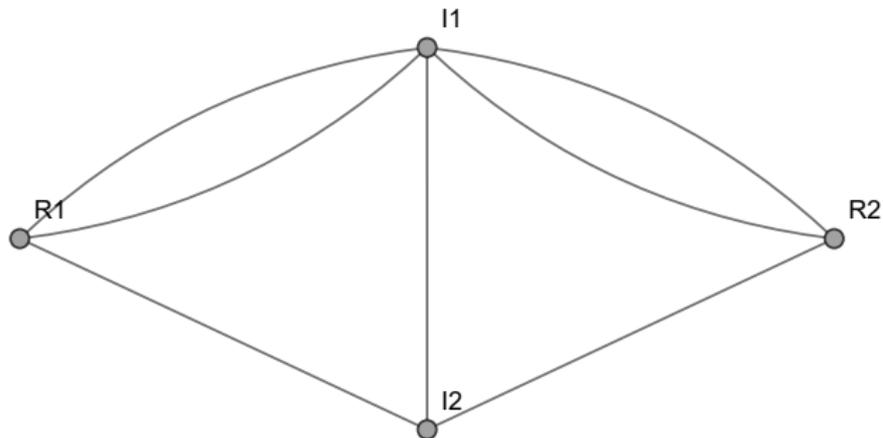
Los vértices  $I_1$  e  $I_2$  representan las islas y los nodos  $R_1$  y  $R_2$  las orillas. *Euler* demostró que un recorrido circuital con todos los lados de este grafo es imposible. El razonamiento empleado se basó en observar las valencias de cada uno de los nodos. Si un vértice no es un punto de partida, necesariamente si se entra a él, se debe contar con una arista distinta que permita salir. Esto significa que cada nodo debe tener una valencia par. En el grafo anterior, todos los grados son impares, por este motivo es imposible formar un circuito de *Euler*. El comando `GrafoKönigsberg` posibilita visualizar las valencias empleando la opción `grados -> True`:

```
In[ ] :=
```

```
GrafoKönigsberg[grados -> True]
```

# Problema de los siete puentes de Königsberg

Out[ ] =



	R2	I1	I2	R1
Grado o valencia	3	5	3	3

# Problema de los siete puentes de Königsberg

Además, `GrafoKönigsberg` integra los atributos `image`  $\rightarrow$  `True` y `dimensions3d`  $\rightarrow$  `True`. El primero, construye una imagen que contiene los siete puentes de *Königsberg* y sobre ella el grafo elaborado por *Euler*, el segundo, retorna el grafo de *Königsberg* en tercera dimensión.



**Descargue un archivo**

<https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-120.zip>



**Explicación en video**

[https://youtu.be/\\_-bSmnxHomA](https://youtu.be/_-bSmnxHomA)

## Teorema 5.3

El razonamiento utilizado por *Leonhard Euler* para resolver el problema de los siete puentes de *Königsberg* constituye en la actualidad un reconocido teorema de caracterización sobre la existencia de un circuito euleriano en un grafo conexo.

### Theorem (5.3)

*Sea  $G = (V, E)$  un grafo conexo. Si todas las valencias de los vértices de  $G$  son pares,  $G$  tiene un circuito de Euler. Además, si hay al menos un grado impar en los nodos de  $G$ , el grafo no posee un circuito euleriano.*

## Comentario sobre el teorema 33

El teorema 33 propone una condición necesaria y suficiente para garantizar la existencia de un circuito de *Euler* en un grafo conexo: “todos los grados de los vértices tienen que ser números pares”. Esta propiedad se considera suficiente y necesaria pues con solo la aparición de una valencia impar en uno de los nodos, provoca que el grafo no contenga un circuito euleriano. Si un grafo es desconexo es imposible que posea circuitos de *Euler* al existir partes que se no encuentran relacionadas o conectadas entre sí mediante aristas. La única excepción puede ocurrir cuando un grafo es desconexo sólo por la presencia de puntos aislados. Si todas las valencias de los nodos son pares y al eliminar los puntos aislados de un grafo desconexo, éste queda conexo entonces en el grafo sí existen circuitos de *Euler*.

## Comentario sobre el teorema 33

En las subfiguras 2a y 2b de la página 13, se comparten dos grafos que no contienen circuitos de *Euler*. En 2a hay vértices con grado impar (por ejemplo el nodo *b*) y en 2b al pasar por la arista  $k \bullet \bullet l$  será imposible luego recorrer todas las demás, al ser el grafo desconexo.

Finalmente, cabe destacar la instrucción `CircuitoEulerQ` de la librería **VilCretas**. Esta sentencia retorna `True` si el grafo recibido como argumento contiene un circuito euleriano (se satisface el teorema 33), o bien, `False`, en caso contrario.



Explicación en video

<https://youtu.be/TegH6J-dd8U>

### Example (5.23)

¿Qué condiciones debe cumplir un grafo completo  $K_n$  para tener circuitos de *Euler*?

### Solución del ejemplo 5.23

En el ejemplo 7 se comprobó que todos los vértices  $v_i$  de  $K_n$ ,  $1 \leq i \leq n$ ,  $n \in \mathbb{N}$ , tienen el mismo grado igual a  $n - 1$ . De acuerdo con el teorema 33,  $K_n$  tendrá un circuito de *Euler* si  $n - 1$  es par, como consecuencia se concluye que  $n$  debe ser un número impar. Es decir,  $K_n$  posee circuitos de *Euler* sí y solo sí  $n$  es impar.

### Example (5.24)

¿Cuáles condiciones se deben satisfacer en un grafo bipartito completo  $K_{n,m}$  para poseer circuitos eulerianos?

## Solución del ejemplo 5.24

En el ejemplo 8 se corroboró que en el grafo  $K_{n,m}$  solo hay dos tipos de valencias. Los  $n$  vértices izquierdos ( $v_i^I$ ,  $1 \leq i \leq n$ ,  $i \in \mathbb{N}$ ) tienen grado igual a  $m$  y los  $m$  nodos derechos ( $v_j^D$ ,  $1 \leq j \leq m$ ,  $j \in \mathbb{N}$ ) poseen valencia igual a  $n$ . Si se desea que todos los grados sean pares para garantizar la existencia de un circuito de *Euler* en  $K_{n,m}$ , se infiere que tanto  $n$  como  $m$  deben ser números pares. Se deduce, por consiguiente, que  $K_{n,m}$  tiene circuitos de *Euler* sí y solo sí  $n$  y  $m$  son números pares.

- Otra situación importante, radica en saber cómo encontrar un circuito de *Euler* sobre un grafo  $G$  que cumpla lo propuesto por el teorema 33. Para ello es usual en teoría de grafos utilizar un método denominado “algoritmo de *Fleury*”.

## Teorema 5.4 Algoritmo de Fleury

### Theorem (5.4)

Sea  $G = (V, E)$  un grafo conexo con grados pares en todos sus vértices y  $L$  un conjunto de aristas sobre  $G$ . Un circuito de Euler en el grafo  $G$  se forma así:

- 1 Se parte de un vértice cualquiera de  $G$  y se agrega a  $L$  una arista inicial  $e$ , que no sea un puente. Un puente es un lado del grafo, que al ser eliminado, lo transforma en desconexo. Sea el conjunto de aristas seleccionables  $CAS$ ,  $CAS = E - \{e\}$ .
- 2 Si  $CAS = \emptyset$ , se ha finalizado,  $L$  contiene un circuito de Euler.

## Teorema 5.4 Algoritmo de Fleury. Continuación.

### Theorem (5.4)

- 3 *Se escoge un nuevo lado  $e = [w, k]$  para el circuito a construir, siempre y cuando  $e$  no corresponda a un puente y siendo  $[v, w]$  la última arista de  $L$ . Si el único lado elegible  $e$  es un puente, se selecciona.*
- 4 *Sea  $CAS = CAS - \{e\}$  y  $L = L \cup \{e\}$ , quedando  $e = [w, k]$  como último elemento de  $L$ . Vaya al paso dos.*

## Comentario sobre el teorema 36

La descripción del algoritmo de *Fleury* planteada en el teorema 36 es muy adecuada si se desea realizar una implementación de este procedimiento en algún lenguaje de programación. La idea se sustentaría en la creación de un método alternativo que verifique en cada iteración, si una arista a seleccionar en  $CAS$  no es un puente, o bien, verificando que en el grafo correspondiente, los únicos lados disponibles son puentes, esto, hasta completar en  $L$  todas las aristas de  $G$ . En el paquete **VilCretas**, el comando `Fleury` ejemplifica en lenguaje *Wolfram* el desarrollo del algoritmo de *Fleury*, facilitando paso a paso una visualización de su empleo.



Explicación en video

<https://youtu.be/0BIJdj86jVk>

## Comentario sobre el teorema 36

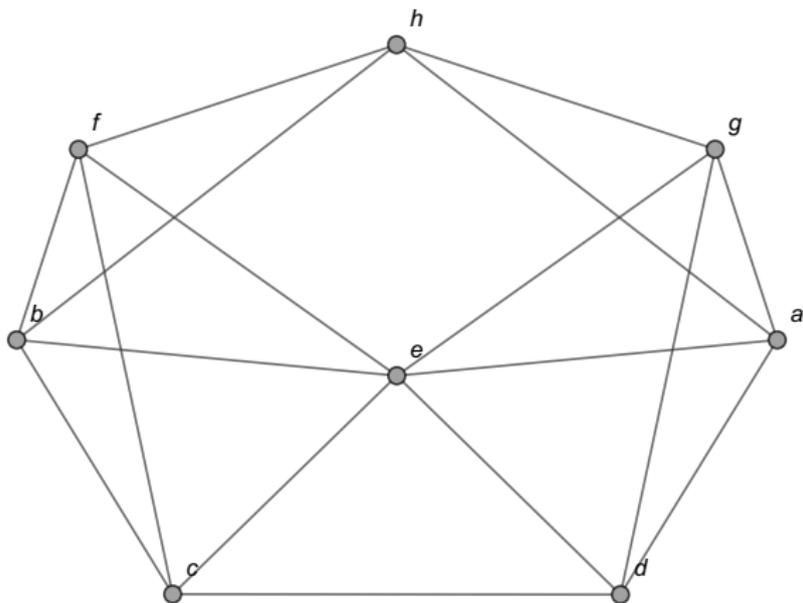
Del teorema 36 se desprende una variante de utilización del algoritmo de *Fleury*, si el objetivo es practicar su uso manualmente con lápiz y papel. Los pasos se podrían resumir de la siguiente manera:

- 1 Se parte de un vértice y se forma un circuito cualesquiera sobre el grafo.
- 2 Se eliminan los lados del circuito y los vértices aislados. Si el subgrafo desaparece resultando una figura vacía, sin aristas y sin nodos, se ha finalizado. El circuito de *Euler* buscado se construye al sustituir de manera iterativa, cada uno de los circuitos encontrados en los pasos precedentes.
- 3 Se elabora un camino circuitual cualesquiera sobre el subgrafo del paso anterior, partiendo de algún vértice contenido en el último circuito. Vaya al paso 2.

Los ejemplos propuestos a continuación tomarán como base esta variante al usar el algoritmo de *Fleury*.

## Example (5.25)

Halle un circuito euleriano sobre el grafo:



## Solución del ejemplo 5.25

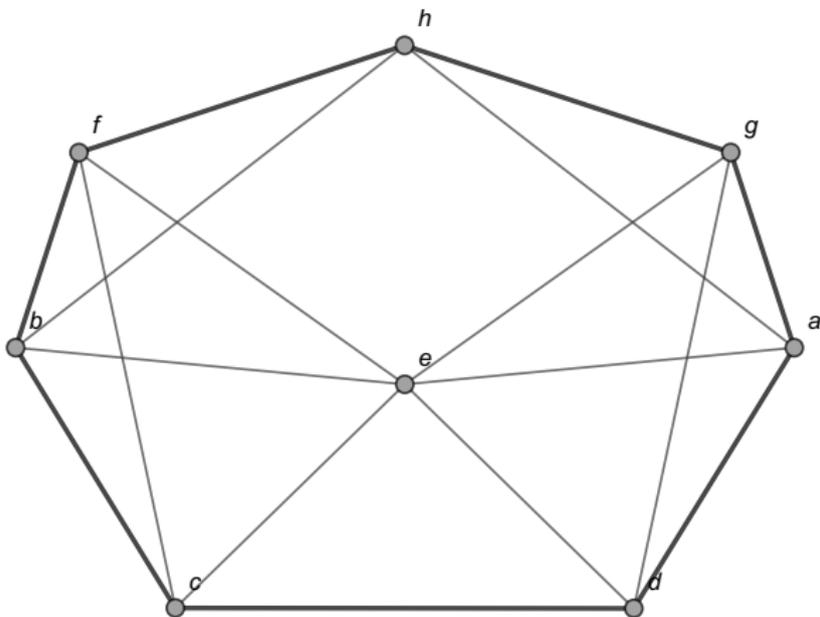
Al encontrar las valencias de los vértices del grafo dado, se obtiene:

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
Grado o valencia	4	4	4	4	6	4	4	4

Todos los grados son pares, por lo que, de acuerdo con el teorema 33, el grafo posee circuitos de *Euler*. Para hallar uno de estos circuitos se procederá en función de la variante de utilización del algoritmo de *Fleury*, ya compartida. Se iniciará formando un circuito cualquiera sobre el grafo, por ejemplo:  $h \rightarrow g \rightarrow a \rightarrow d \rightarrow c \rightarrow b \rightarrow f \rightarrow h$ .

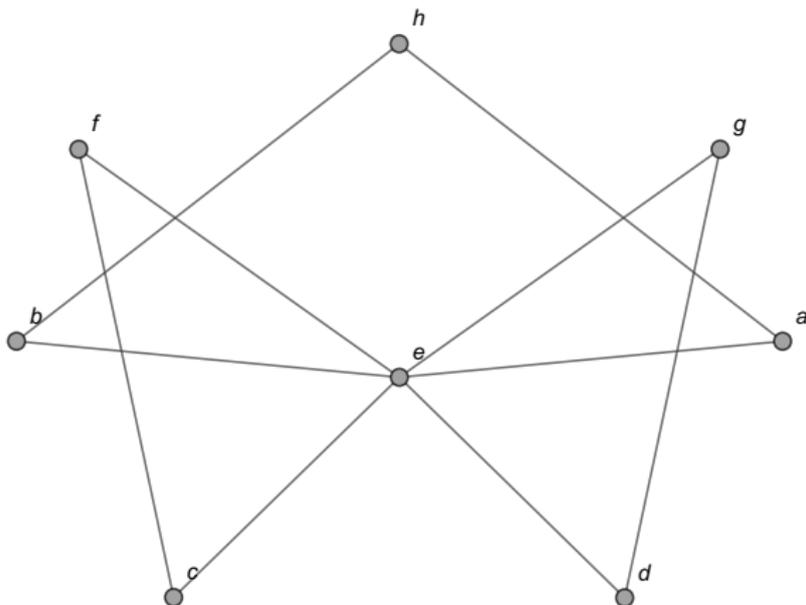
## Solución del ejemplo 5.25

Gráficamente:



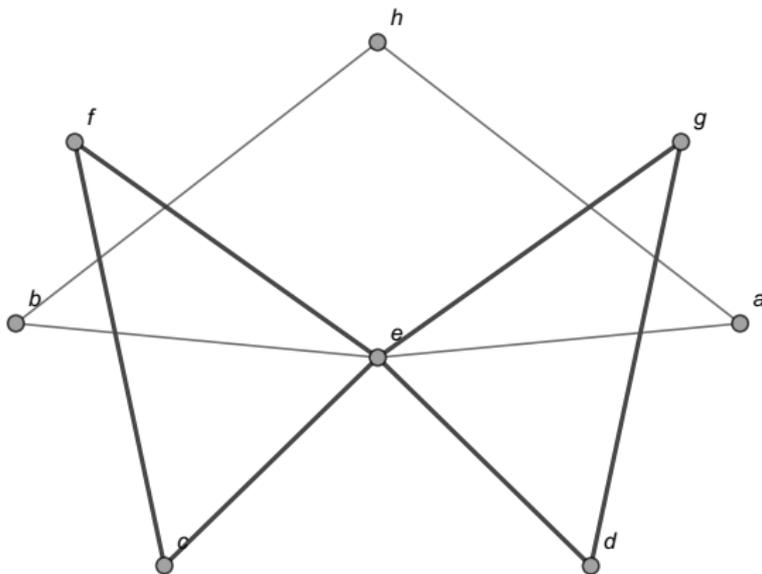
## Solución del ejemplo 5.25

Ahora, se deben eliminar las aristas empleadas en ese camino, quedando el subgrafo:



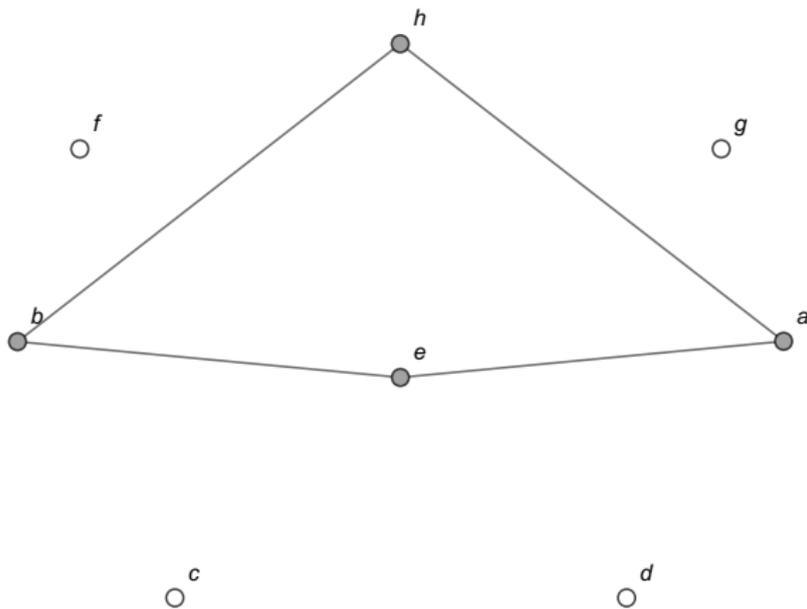
## Solución del ejemplo 5.25

Corresponde construir otro circuito que comience con algún vértice del circuito anterior. Por ejemplo:  $d \rightarrow e \rightarrow f \rightarrow c \rightarrow e \rightarrow g \rightarrow d$ , es decir:



## Solución del ejemplo 5.25

Si se eliminan los nodos aislados y las aristas de la trayectoria descrita entonces queda el subgrafo:

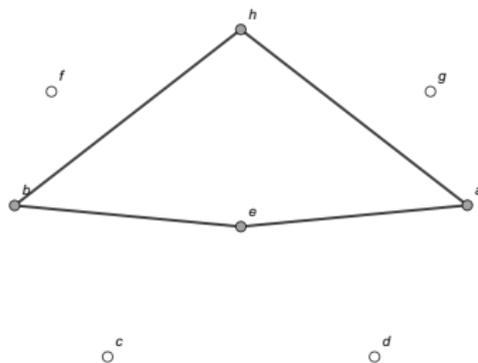


## Solución del ejemplo 5.25

Los puntos huecos en la figura representan la sustracción de los vértices  $c$ ,  $d$ ,  $f$  y  $g$ . Un circuito sobre este nuevo grafo no debería iniciar en los nodos  $a$ ,  $b$  o  $h$  al no estar considerados en el circuito de la iteración anterior:

$d \rightarrow e \rightarrow f \rightarrow c \rightarrow e \rightarrow g \rightarrow d$ . En otras palabras, el nuevo circuito forzosamente iniciará en  $e$ , por ejemplo:  $e \rightarrow b \rightarrow h \rightarrow a \rightarrow e$ .

Gráficamente:



## Solución del ejemplo 5.25

Como al eliminar los vértices aislados y los lados utilizados en ese camino nos quedamos sin grafo, se ha finalizado el proceso. Un circuito euleriano se construye al cambiar de manera iterativa los tres circuitos encontrados. Partiendo del circuito hallado en el primera iteración:

$$h \rightarrow g \rightarrow a \rightarrow d \rightarrow c \rightarrow b \rightarrow f \rightarrow h$$

Se debe sustituir la  $d$  por  $d \rightarrow e \rightarrow f \rightarrow c \rightarrow e \rightarrow g \rightarrow d$ , que conforma el circuito elaborado en la segunda iteración, por lo tanto:

1.  $h \rightarrow g \rightarrow a \rightarrow d \rightarrow c \rightarrow b \rightarrow f \rightarrow h$
2.  $h \rightarrow g \rightarrow a \rightarrow d \rightarrow e \rightarrow f \rightarrow c \rightarrow e \rightarrow g \rightarrow d \rightarrow c \rightarrow b \rightarrow f \rightarrow h$
3.  $h \rightarrow g \rightarrow a \rightarrow d \rightarrow e \rightarrow f \rightarrow c \rightarrow e \rightarrow g \rightarrow d \rightarrow c \rightarrow b \rightarrow f \rightarrow h$

## Solución del ejemplo 5.25

En este recorrido, se debe reemplazar la letra  $e$  por el tercer circuito obtenido:  $e \rightarrow b \rightarrow h \rightarrow a \rightarrow e$ . Como existen dos letras  $e$  se selecciona cualquiera de ellas para realizar la sustitución. Veamos:

1.  $h \rightarrow g \rightarrow a \rightarrow d \rightarrow e \rightarrow f \rightarrow c \rightarrow \boxed{e} \rightarrow g \rightarrow d \rightarrow c \rightarrow b \rightarrow f \rightarrow h$
2.  $h \rightarrow g \rightarrow a \rightarrow d \rightarrow e \rightarrow f \rightarrow c \rightarrow \boxed{e \rightarrow b \rightarrow h \rightarrow a \rightarrow e} \rightarrow g \rightarrow d \rightarrow c \rightarrow b \rightarrow f \rightarrow h$
3.  $h \rightarrow g \rightarrow a \rightarrow d \rightarrow e \rightarrow f \rightarrow c \rightarrow e \rightarrow b \rightarrow h \rightarrow a \rightarrow e \rightarrow g \rightarrow d \rightarrow c \rightarrow b \rightarrow f \rightarrow h$

## Solución del ejemplo 5.25

Si el lector hubiera optado por reemplazar la primera  $e$  y no la segunda, se determinaría otro circuito euleriano distinto. No es correcto, eso sí, sustituir ambas  $e$  en la trayectoria pues, si así se procede, se estaría pasando varias veces por algunos de los lados del grafo, lo cual contradice la definición de ruta. Finalmente, el recorrido siguiente constituye un circuito euleriano:

$$h \rightarrow g \rightarrow a \rightarrow d \rightarrow e \rightarrow f \rightarrow c \rightarrow e \rightarrow b \rightarrow h \rightarrow a \rightarrow e \rightarrow g \rightarrow d \\ \rightarrow c \rightarrow b \rightarrow f \rightarrow h$$

(4)

## Solución del ejemplo 5.25

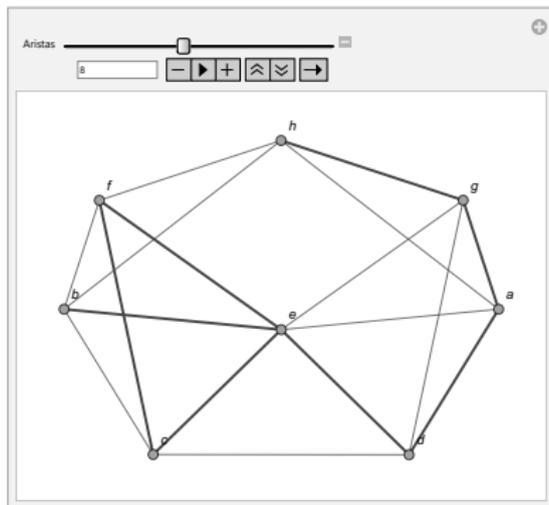
El comando `AnimarGrafo` del paquete **VilCretas** facilita visualizar arista por arista, que el camino 4 pasa una única vez por cada uno de los lados del grafo original:

**In[ ]** :=

```
grafo = Grafo[{{a, d}, {a, e}, {a, g}, {a, h}, {b, e},  
{b, f}, {c, d}, {c, e}, {d, e}, {d, g}, {e, f}, {g, h},  
{g, e}, {b, c}, {f, h}, {h, b}, {f, c}}];  
AnimarGrafo[grafo, {{h, g}, {g, a}, {a, d}, {d, e},  
{e, f}, {f, c}, {c, e}, {e, b}, {b, h}, {h, a}, {a, e},  
{e, g}, {g, d}, {d, c}, {c, b}, {b, f}, {f, h}}]
```

## Solución del ejemplo 5.25

Out[ ] =



Al manipular el deslizador *Aristas* de la animación se verifica que 4 es un circuito de *Euler*.

## Solución del ejemplo 5.25

Por otra parte, se aclara al alumno que el comando `CircuitosEuler` de **VilCretas**, ofrece una alternativa para hallar  $n$  circuitos eulerianos,  $n \in \mathbb{N}$ , sobre un grafo conexo que los contiene. La instrucción recibe el grafo y el valor respectivo de  $n$ . En algunas ocasiones, podría suceder que `CircuitosEuler` retorne una cantidad de circuitos menor a  $n$  y no igual a  $n$ . En este ejercicio al solicitar 3 circuitos eulerianos, `CircuitosEuler` devuelve:

```
In[ ] :=
```

```
CircuitosEuler[grafo, 3]
```

## Solución del ejemplo 5.25

Out[ ] =

$$\{ \{ a \bullet \bullet d, d \bullet \bullet g, g \bullet \bullet e, e \bullet \bullet f, f \bullet \bullet c, c \bullet \bullet b, b \bullet \bullet h, h \bullet \bullet f, f \bullet \bullet b, b \bullet \bullet e, e \bullet \bullet d, d \bullet \bullet c, c \bullet \bullet e, e \bullet \bullet a, a \bullet \bullet h, h \bullet \bullet g, g \bullet \bullet a \}, \{ a \bullet \bullet d, d \bullet \bullet g, g \bullet \bullet e, e \bullet \bullet f, f \bullet \bullet c, c \bullet \bullet b, b \bullet \bullet h, h \bullet \bullet f, f \bullet \bullet b, b \bullet \bullet e, e \bullet \bullet d, d \bullet \bullet c, c \bullet \bullet e, e \bullet \bullet a, a \bullet \bullet g, g \bullet \bullet h, h \bullet \bullet a \}, \{ a \bullet \bullet d, d \bullet \bullet g, g \bullet \bullet e, e \bullet \bullet f, f \bullet \bullet c, c \bullet \bullet b, b \bullet \bullet h, h \bullet \bullet f, f \bullet \bullet b, b \bullet \bullet e, e \bullet \bullet c, c \bullet \bullet d, d \bullet \bullet e, e \bullet \bullet a, a \bullet \bullet h, h \bullet \bullet g, g \bullet \bullet a \} \}$$

## Solución del ejemplo 5.25

La sentencia `CircuitosEuler`, en este caso, sí fue capaz de encontrar los 3 circuitos correspondientes. También, `CircuitosEuler` integra la opción `ruta -> True` que muestra una animación de recorrido con uno de los circuitos eulerianos, seleccionado por el software de manera pseudoaleatoria.



## Descargue un archivo

<https://www.escinf.una.ac.cr/discretas/Archivos/Grafos/File-121.zip>



## Explicación en video

<https://youtu.be/T4t14bwCJbI>

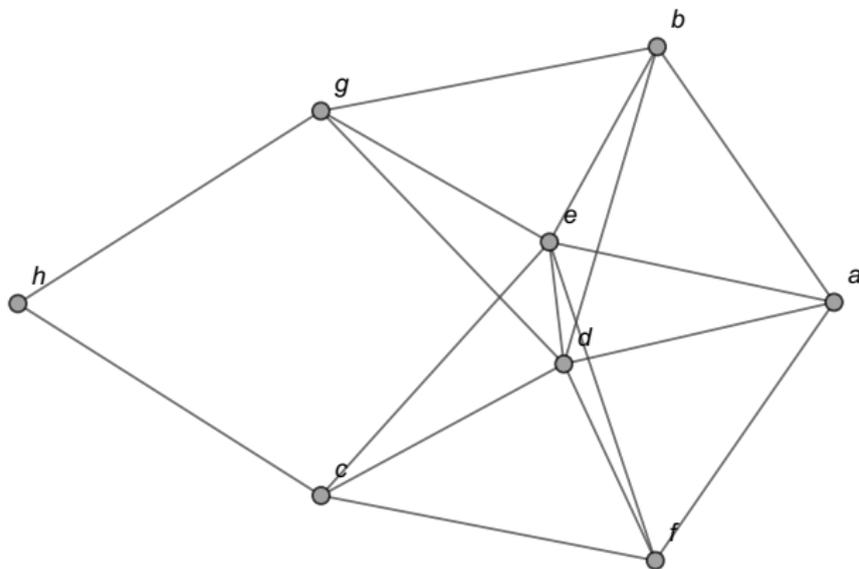


## Explicación en video

<https://youtu.be/lo9EOHLLUXw>

## Example (5.26)

Determine sobre el grafo dado un circuito de *Euler*.



## Solución del ejemplo 5.26

Las valencias de los vértices del grafo corresponden a:

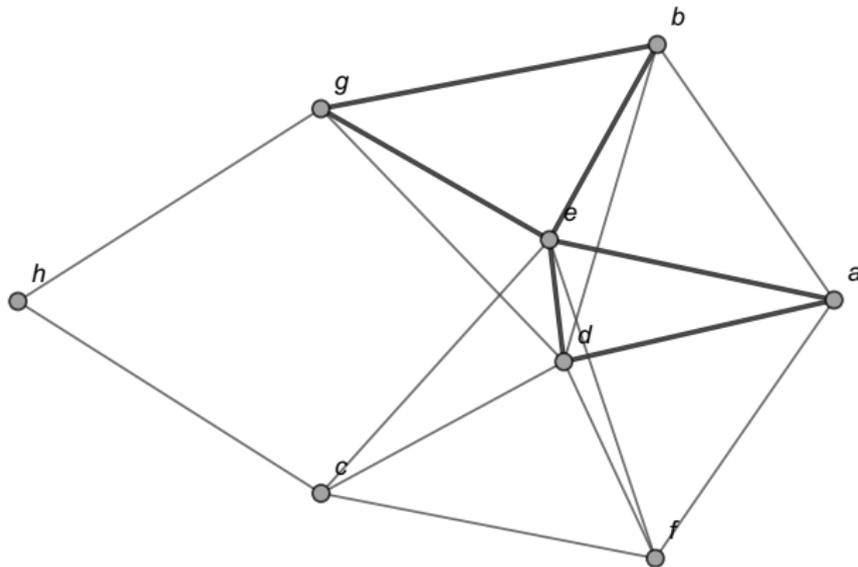
	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
Grado o valencia	4	4	4	6	6	4	4	2

Al ser todos los grados números pares, se garantiza que el grafo contiene circuitos eulerianos, por el teorema 33. Con la intención de encontrar uno de estos caminos se tomará, por ejemplo, el siguiente circuito:

$a \rightarrow d \rightarrow e \rightarrow b \rightarrow g \rightarrow e \rightarrow a$ .

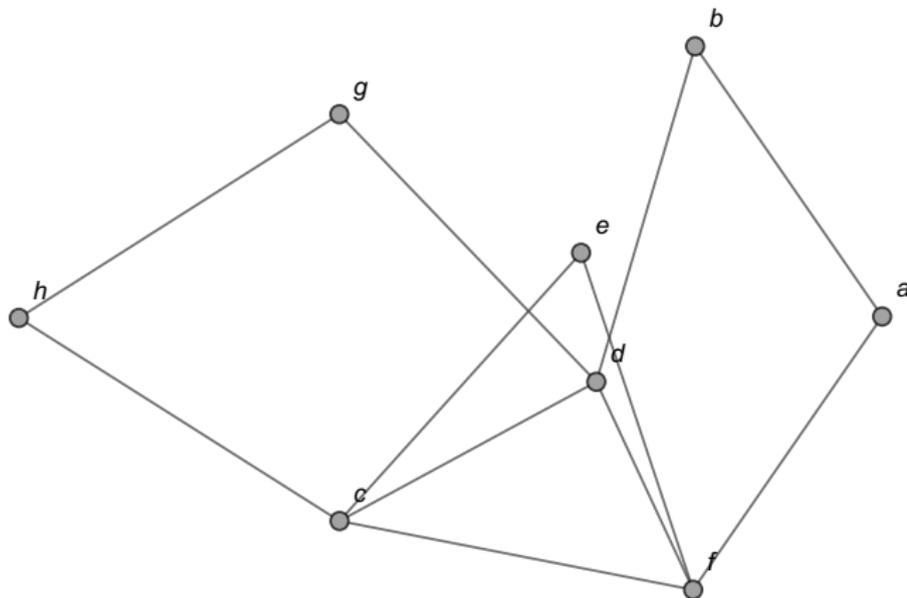
## Solución del ejemplo 5.26

Gráficamente:



## Solución del ejemplo 5.26

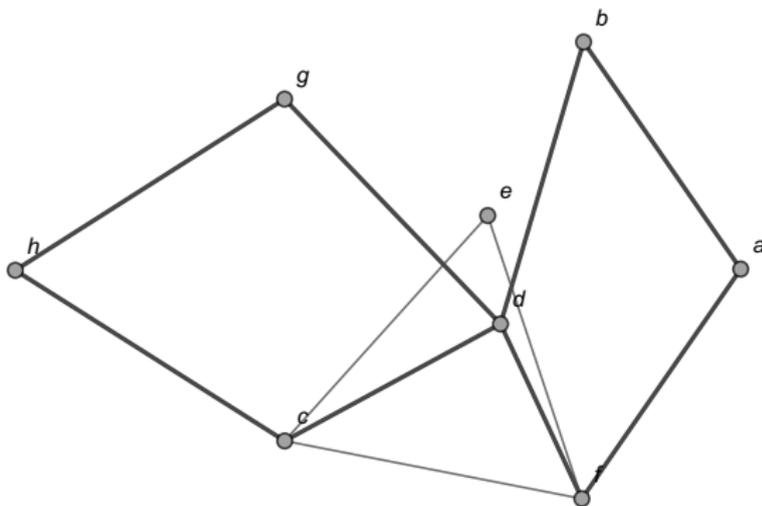
Se eliminan del grafo los lados involucrados en esa trayectoria, obteniéndose el subgrafo:



## Solución del ejemplo 5.26

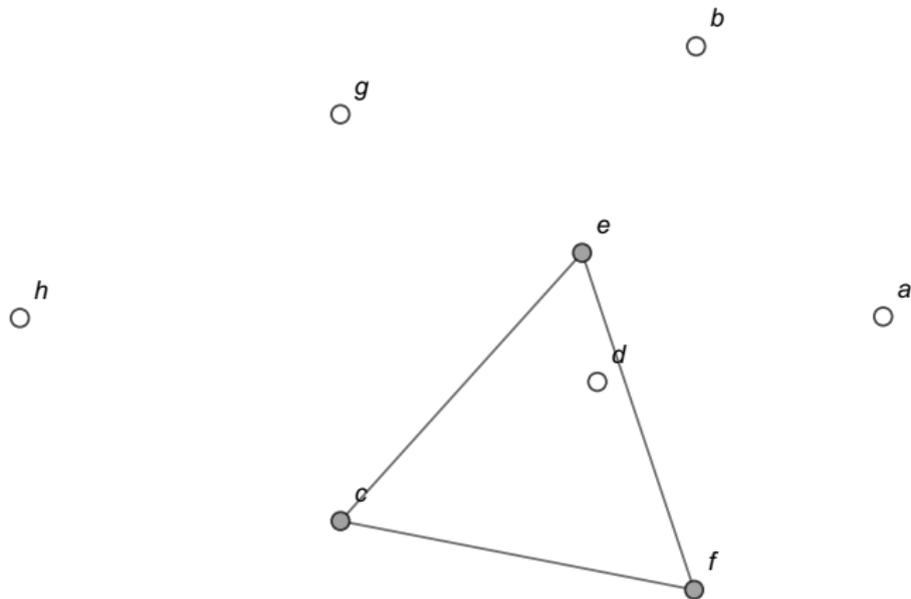
Ahora, eligiendo el vértice  $g$  contenido en el primer circuito elaborado, se construye el camino circuital:  $g \rightarrow h \rightarrow c \rightarrow d \rightarrow f \rightarrow a \rightarrow b \rightarrow d \rightarrow g$ .

En el grafo:



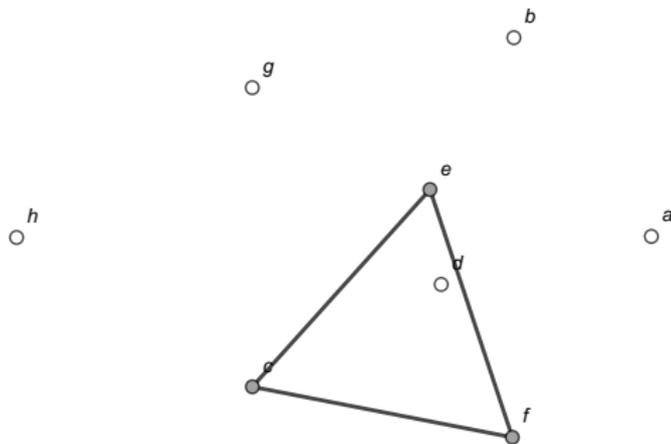
## Solución del ejemplo 5.26

Eliminando las aristas de esta ruta y los nodos aislados, se genera:



## Solución del ejemplo 5.26

Sobre este subgrafo el circuito a construir no puede partir de  $e$ , al no formar parte la letra  $e$  del circuito anterior, por consiguiente, el nuevo circuito debe iniciar en  $c$  o en  $f$ . Al tomar:  $f \rightarrow e \rightarrow c \rightarrow f$ , se finaliza el proceso pues se eliminarían todos los lados y vértices del grafo original:



## Solución del ejemplo 5.26

Para hallar un circuito de *Euler* de interés, se debe reemplazar la letra  $g$  por  $g \rightarrow h \rightarrow c \rightarrow d \rightarrow f \rightarrow a \rightarrow b \rightarrow d \rightarrow g$  en  $a \rightarrow d \rightarrow e \rightarrow b \rightarrow g \rightarrow e \rightarrow a$  y la letra  $f$  por  $f \rightarrow e \rightarrow c \rightarrow f$  en  $g \rightarrow h \rightarrow c \rightarrow d \rightarrow f \rightarrow a \rightarrow b \rightarrow d \rightarrow g$ , es decir:

$$1. a \rightarrow d \rightarrow e \rightarrow b \rightarrow \boxed{g} \rightarrow e \rightarrow a$$

$$2. a \rightarrow d \rightarrow e \rightarrow b \rightarrow \boxed{g \rightarrow h \rightarrow c \rightarrow d \rightarrow f \rightarrow a \rightarrow b \rightarrow d \rightarrow g} \rightarrow e \rightarrow a$$

$$3. a \rightarrow d \rightarrow e \rightarrow b \rightarrow g \rightarrow h \rightarrow c \rightarrow d \rightarrow \boxed{f} \rightarrow a \rightarrow b \rightarrow d \rightarrow g \rightarrow e \rightarrow a$$

$$4. a \rightarrow d \rightarrow e \rightarrow b \rightarrow g \rightarrow h \rightarrow c \rightarrow d \rightarrow \boxed{f \rightarrow e \rightarrow c \rightarrow f} \rightarrow a \rightarrow b \rightarrow d \rightarrow g \rightarrow e \rightarrow a$$

$$5. a \rightarrow d \rightarrow e \rightarrow b \rightarrow g \rightarrow h \rightarrow c \rightarrow d \rightarrow f \rightarrow e \rightarrow c \rightarrow f \rightarrow a \rightarrow b \rightarrow d \rightarrow g \rightarrow e \rightarrow a$$

## Solución del ejemplo 5.26

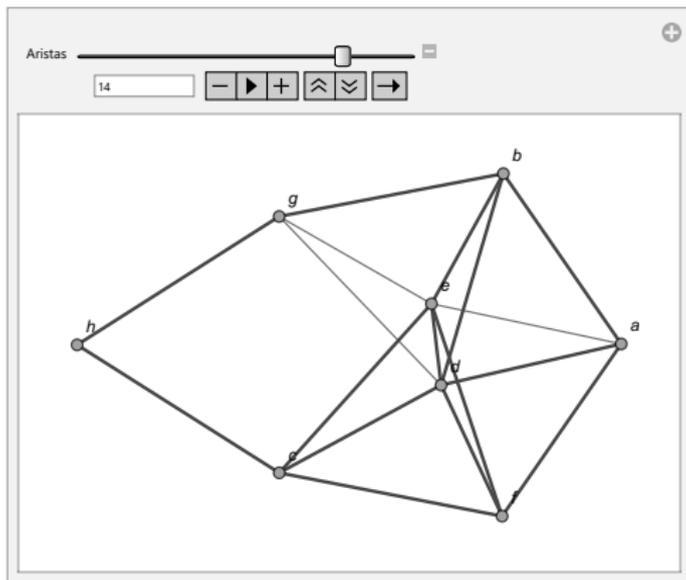
La última trayectoria es un circuito euleriano, lo cual es comprobable mediante la instrucción `AnimarGrafo`:

**In[ ] :=**

```
grafo = Grafo[{{a, e}, {d, f}, {e, f}, {a, d}, {d, e},  
{h, c}, {h, g}, {a, f}, {b, d}, {b, e}, {b, g}, {c, d},  
{c, e}, {c, f}, {d, g}, {e, g}, {a, b}}];  
AnimarGrafo[grafo, {{a, d}, {d, e}, {e, b}, {b, g},  
{g, h}, {h, c}, {c, d}, {d, f}, {f, e}, {e, c}, {c, f},  
{f, a}, {a, b}, {b, d}, {d, g}, {g, e}, {e, a}}]
```

## Solución del ejemplo 5.26

Out[ ] =





## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-122.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-122.zip)

## Teorema 5.5

En ciertos grafos que no poseen un circuito de *Euler* es posible determinar una ruta de *Euler*. Las trayectorias eulerianas como ya se mencionó en la definición 32, incluyen todas las aristas de un grafo donde el nodo inicial es distinto del vértice final. Consideremos el siguiente teorema.

### Theorem (5.5)

*Sea  $G = (V, E)$  un grafo conexo. Si  $G$  tiene exactamente dos nodos  $a$  y  $b$  de valencia impar,  $G$  posee una ruta de *Euler* que une el vértice  $a$  con el nodo  $b$ . Si en  $G$  hay más o menos de dos vértices con grado impar entonces  $G$  no contiene ningún camino euleriano.*

## Comentario sobre el teorema 40

El teorema 40 propone una condición suficiente y necesaria para garantizar la existencia de una ruta euleriana en un grafo conexo: “exactamente dos nodos del grafo tienen que tener grados impares”. La condición es suficiente y necesaria dado que si hay más o menos vértices con valencia impar, el grafo no contendría una ruta de *Euler*. Un camino de *Euler* en un grafo que satisfaga las condiciones del teorema 40, se puede hallar usando el algoritmo de *Fleury*. Si los dos nodos con grados impares en el grafo de interés son adyacentes, se elimina temporalmente la arista que los une. Esto provoca que todos los vértices queden con valencias pares.

## Comentario sobre el teorema 40

Luego, se usa el algoritmo de *Fleury* para encontrar un circuito de *Euler*, eso sí, que inicie en alguno de los vértices cuyo grado es impar en el grafo original y al hallar el circuito euleriano, se le añade a la trayectoria el lado eliminado temporalmente, sea al principio o al final del camino. Esta técnica de trabajo produce una ruta de *Euler*. Si los dos vértices con valencia impar no son adyacentes, se añade una arista temporal que los una (todos los grados de los nodos en el nuevo grafo quedarían como números pares), se usa el algoritmo de *Fleury* comenzando con ese lado temporal y al obtener el circuito euleriano correspondiente, se suprime la arista agregada, formándose un camino de *Euler*.

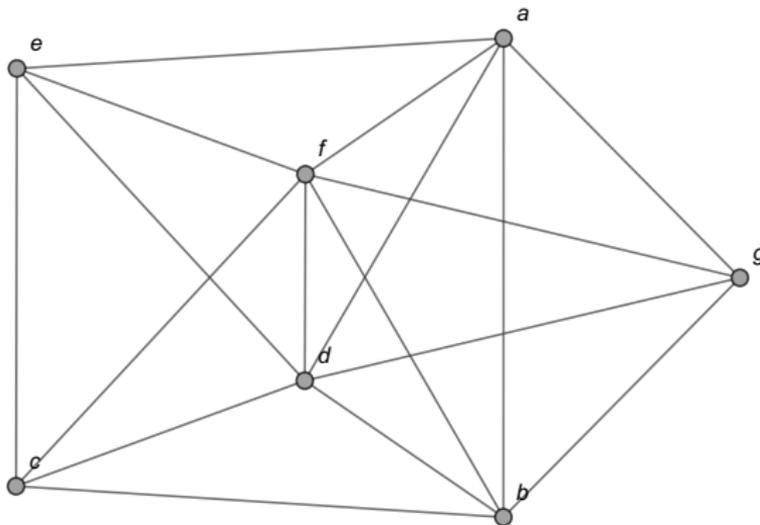
## Nota

También, por los teoremas 33 y 40 se concluye que un grafo no puede poseer un circuito y una ruta euleriana simultáneamente. Si el grafo tiene un circuito euleriano todos los grados de los nodos son pares, lo cual descartaría la existencia de un camino de *Euler* y si el grafo contiene una ruta euleriana hay exactamente dos vértices con valencia impar, lo que recusa la existencia de un circuito euleriano.

Se abordarán algunos ejemplos de construcción de una ruta de *Euler*.

### Example (5.27)

Halle una ruta euleriana sobre el grafo:



## Solución del ejemplo 5.27

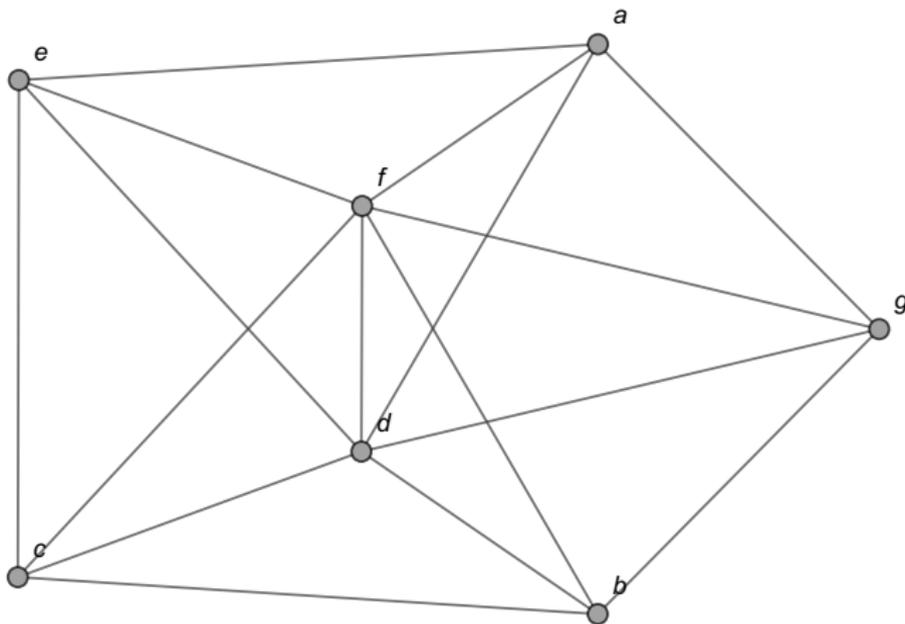
La tabla de grados de este grafo corresponde a:

	$a$	$b$	$c$	$d$	$e$	$f$	$g$
Grado o valencia	5	5	4	6	4	6	4

Existen exactamente dos vértices con valencia impar, los nodos  $a$  y  $b$ . De acuerdo con el teorema 40, hay una ruta de *Euler* que los une. Para encontrar un camino euleriano se aprecia en el grafo que los vértices  $a$  y  $b$  son adyacentes.

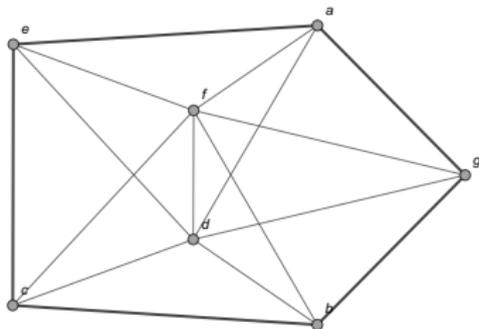
## Solución del ejemplo 5.27

Asumimos ahora, que el lado  $a \bullet \bullet b$  debe ser eliminado temporalmente del grafo, obteniéndose:



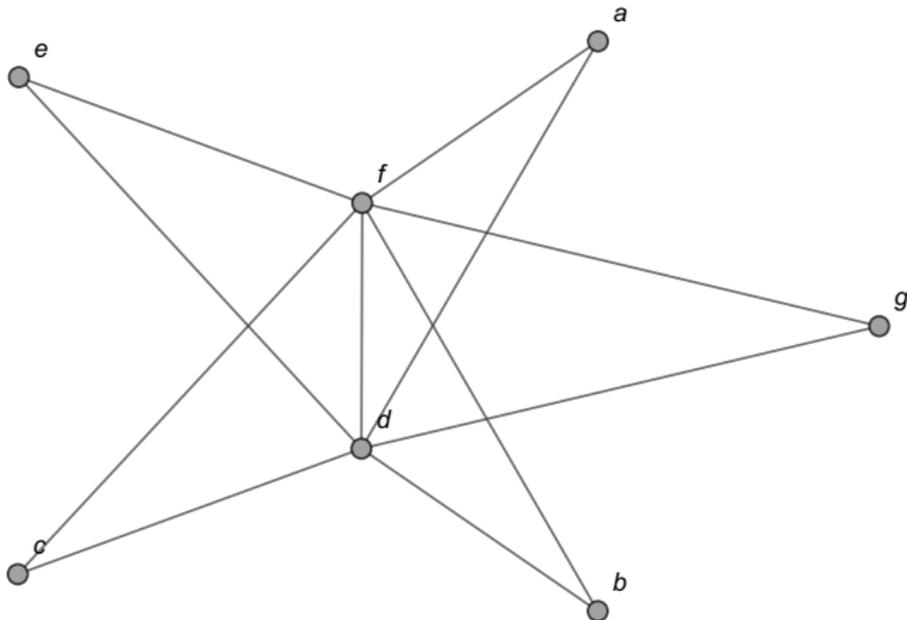
## Solución del ejemplo 5.27

En este nuevo grafo existe un circuito de *Euler* pues todas las valencias de los nodos son pares. Al utilizar el algoritmo de *Fleury*, se inicia formando un circuito que debe comenzar estrictamente en el vértice  $a$ , o bien, en el nodo  $b$ , para poder añadir al final del proceso la arista sustraída temporalmente. Por ejemplo, un circuito que inicia en  $b$  es:  
 $b \rightarrow c \rightarrow e \rightarrow a \rightarrow g \rightarrow b$ . Gráficamente:



## Solución del ejemplo 5.27

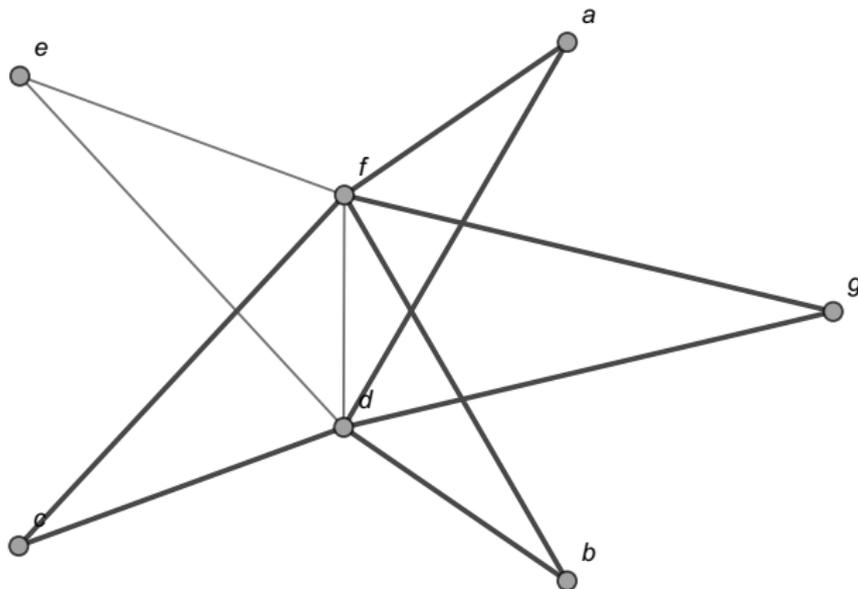
Al eliminar los lados del circuito nos queda:



## Solución del ejemplo 5.27

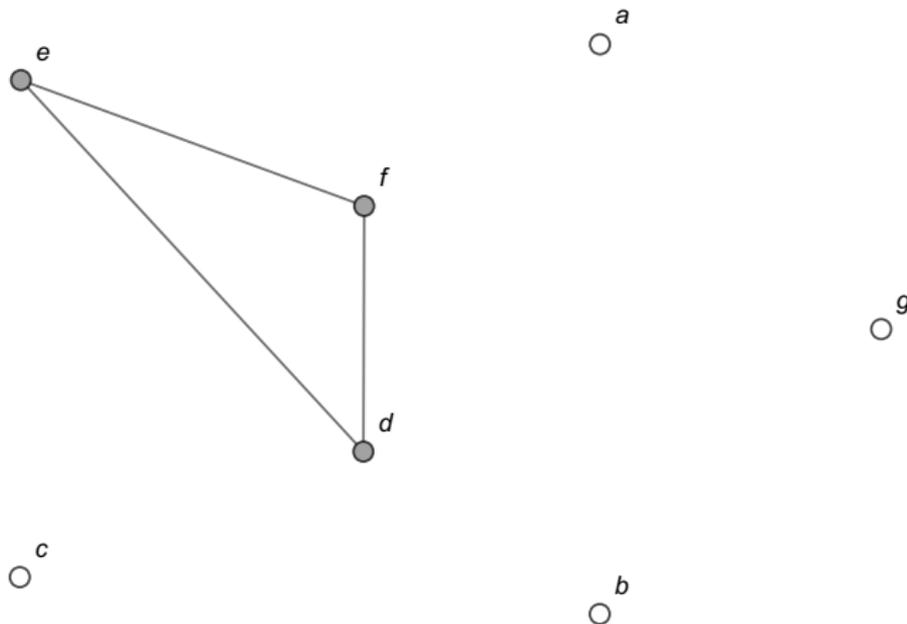
En este subgrafo, un circuito en  $c$  corresponde a:

$c \rightarrow f \rightarrow a \rightarrow d \rightarrow b \rightarrow f \rightarrow g \rightarrow d \rightarrow c$ . Es decir:



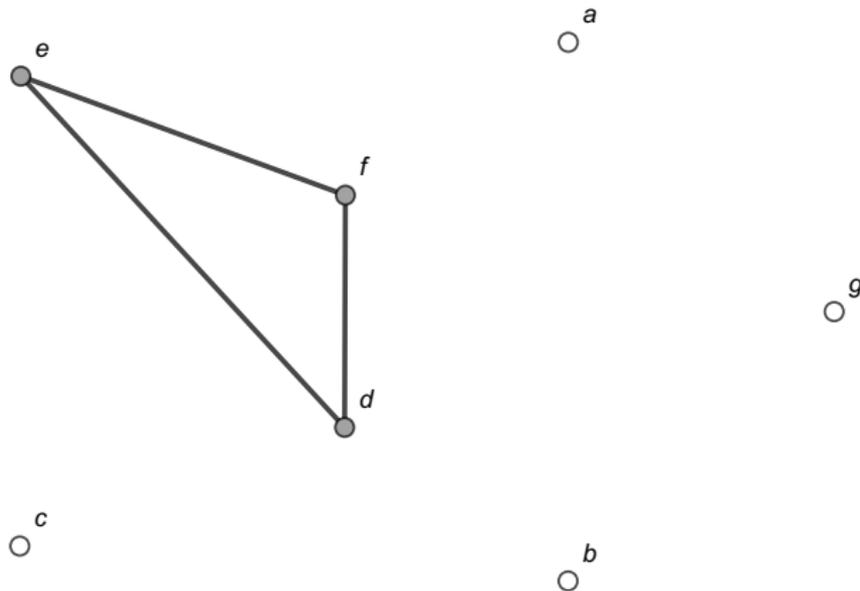
## Solución del ejemplo 5.27

Al eliminar las aristas utilizadas en ese camino y los vértices aislados se genera el siguiente subgrafo:



## Solución del ejemplo 5.27

En él, un circuito que comienza con  $d$  es:  $d \rightarrow e \rightarrow f \rightarrow d$ . Gráficamente:



## Solución del ejemplo 5.27

El vértice  $f$  también es elegible para construir el circuito de esta iteración, sin embargo, el nodo  $e$  no es seleccionable al no estar contenido en el circuito de la iteración anterior. Se ha finalizado, pues al eliminar los lados y nodos del camino  $d \rightarrow e \rightarrow f \rightarrow d$  la figura queda vacía. Luego, al sustituir de manera iterativa los tres circuitos hallados:

$$1. b \rightarrow \boxed{c} \rightarrow e \rightarrow a \rightarrow g \rightarrow b$$

$$2. b \rightarrow \boxed{c \rightarrow f \rightarrow a \rightarrow d \rightarrow b \rightarrow f \rightarrow g \rightarrow d \rightarrow c} \rightarrow e \rightarrow a \rightarrow g \rightarrow b$$

$$3. b \rightarrow c \rightarrow f \rightarrow a \rightarrow \boxed{d} \rightarrow b \rightarrow f \rightarrow g \rightarrow d \rightarrow c \rightarrow e \rightarrow a \rightarrow g \rightarrow b$$

$$4. b \rightarrow c \rightarrow f \rightarrow a \rightarrow \boxed{d \rightarrow e \rightarrow f \rightarrow d} \rightarrow b \rightarrow f \rightarrow g \rightarrow d \rightarrow c \rightarrow e \rightarrow a \rightarrow g \rightarrow b$$

$$5. b \rightarrow c \rightarrow f \rightarrow a \rightarrow d \rightarrow e \rightarrow f \rightarrow d \rightarrow b \rightarrow f \rightarrow g \rightarrow d \rightarrow c \rightarrow e \rightarrow a \rightarrow g \rightarrow b$$

## Solución del ejemplo 5.27

La última trayectoria es un circuito de *Euler*, para formar la ruta euleriana implícita, se debe añadir al circuito el lado  $a \bullet \bullet b$  suprimido temporalmente. Esa arista se puede agregar al inicio, o bien, al final del circuito euleriano. Si se agrega al inicio:

$$\begin{aligned}
 a \rightarrow b \rightarrow c \rightarrow f \rightarrow a \rightarrow d \rightarrow e \rightarrow f \rightarrow d \rightarrow b \rightarrow f \rightarrow g \rightarrow d \rightarrow c \\
 \rightarrow e \rightarrow a \rightarrow g \rightarrow b
 \end{aligned}$$

## Solución del ejemplo 5.27

Este camino es una ruta de *Euler* al recorrer una única vez todos los lados del grafo original, comenzando en el vértice  $a$  y finalizando en el nodo  $b$ .

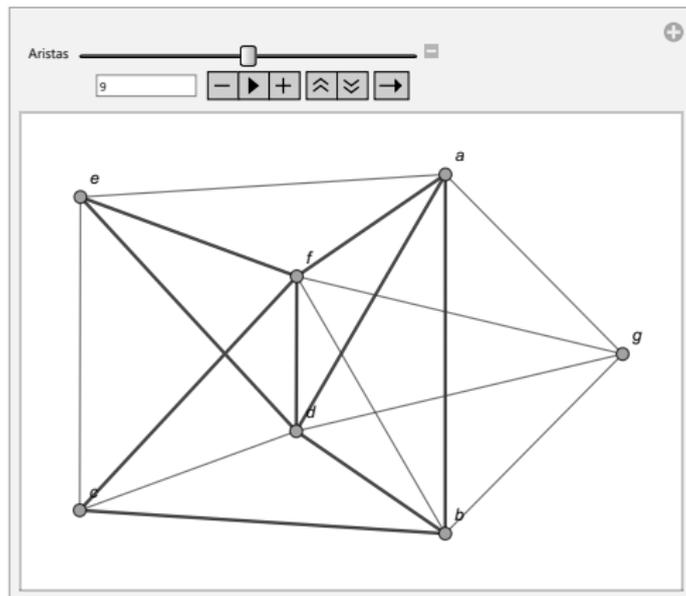
Lo anterior, se puede verificar empleando la instrucción `AnimarGrafo`:

**In[ ] :=**

```
grafo = Grafo[{{a, b}, {a, d}, {a, e}, {a, f}, {a, g},  
{b, c}, {b, d}, {b, f}, {b, g}, {c, d}, {c, e}, {c, f},  
{d, e}, {d, f}, {d, g}, {e, f}, {f, g}}];  
AnimarGrafo[grafo, Partition[{a, b, c, f, a, d, e, f, d, b,  
f, g, d, c, e, a, g, b}, 2, 1]]
```

## Solución del ejemplo 5.27

Out[ ] =



## Solución del ejemplo 5.27

Partition es un comando de *Wolfram Mathematica* que permite en este caso, transformar la trayectoria en nodos: {a, b, c, f, a, d, e, f, d, b, f, g, d, c, e, a, g, b}, a una notación en aristas: {{a, b}, {b, c}, {c, f}, {f, a}, {a, d}, {d, e}, {e, f}, {f, d}, {d, b}, {b, f}, {f, g}, {g, d}, {d, c}, {c, e}, {e, a}, {a, g}, {g, b}}, que es lo que recibe la sentencia AnimarGrafo.

La librería **VilCretas** presenta dos instrucciones para estudiar rutas eulerianas sobre un grafo, éstas son: RutaEulerQ y RutaEuler.

RutaEulerQ es una sentencia booleana que retorna True si el grafo pasado como parámetro posee un camino de *Euler*, o, False, si sucede lo contrario. RutaEuler devuelve explícitamente una ruta de *Euler*, si el grafo recibido como argumento, la contiene.

## Solución del ejemplo 5.27

**In[ ] :=**

RutaEulerQ[grafo]

RutaEuler[grafo]

**Out[ ] =**

True

 $\{\{a, d\}, \{d, b\}, \{b, c\}, \{c, d\}, \{d, e\}, \{e, a\}, \{a, f\}, \{f, b\}, \{b, g\},$   
 $\{g, d\}, \{d, f\}, \{f, c\}, \{c, e\}, \{e, f\}, \{f, g\}, \{g, a\}, \{a, b\}\}$ 

RutaEuler cuenta con la opción `ruta -> True` en caso de querer visualizar en una animación el camino euleriano encontrado por el comando.



## Descargue un archivo

<https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-123.zip>



## Explicación en video

[https://youtu.be/ugy4P4-a-\\_4](https://youtu.be/ugy4P4-a-_4)

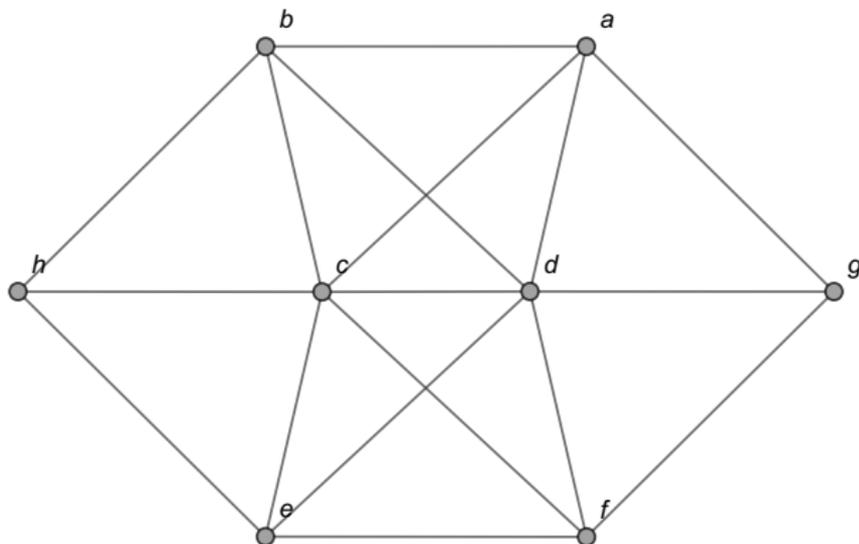


## Explicación en video

<https://youtu.be/7eeaepoPcBE>

## Example (5.28)

Determine un camino de *Euler* sobre el grafo compartido a continuación.

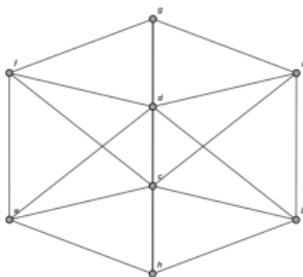


## Solución del ejemplo 5.28

La tabla de valencias de este grafo es la siguiente:

	$a$	$b$	$c$	$d$	$e$	$f$	$g$	$h$
Grado o valencia	4	4	6	6	4	4	3	3

Los vértices  $g$  y  $h$  son los únicos con grado impar. Por el teorema 40, al ser el grafo conexo, contiene una ruta euleriana entre  $g$  y  $h$ . Estos vértices no son adyacentes. Con el objetivo de emplear el algoritmo de *Fleury* como un puente para encontrar un camino de *Euler*, se construirá temporalmente una arista que una los nodos  $g$  y  $h$ , quedando un nuevo grafo:

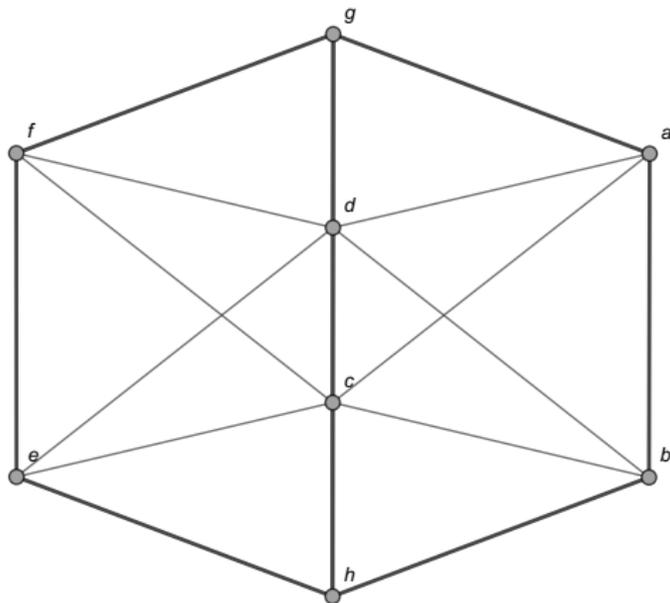


## Solución del ejemplo 5.28

En esta figura, el lector debe apreciar que el lado  $h \bullet \bullet g$  incluido, se ha dibujado por parte del software *Mathematica*, sobrepuesto a las aristas  $g \bullet \bullet d$ ,  $d \bullet \bullet c$  y  $c \bullet \bullet h$ . El nuevo grafo posee únicamente valencias pares en sus vértices, por lo que contiene un circuito de *Euler*. Usando el algoritmo de *Fleury* para hallar un circuito euleriano, se inicia necesariamente con un circuito que tome como su primer lado, la arista temporal  $h \bullet \bullet g$  que será eliminada al finalizar todo el proceso. Por ejemplo:  $h \rightarrow g \rightarrow f \rightarrow e \rightarrow h \rightarrow b \rightarrow a \rightarrow g \rightarrow d \rightarrow c \rightarrow h$ .

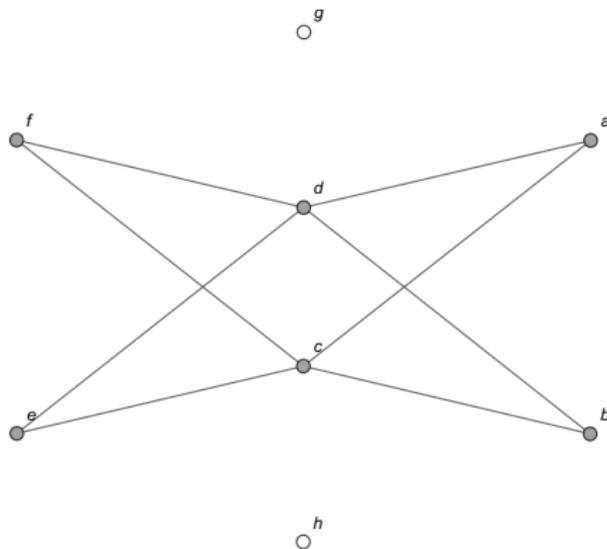
## Solución del ejemplo 5.28

Visualmente:



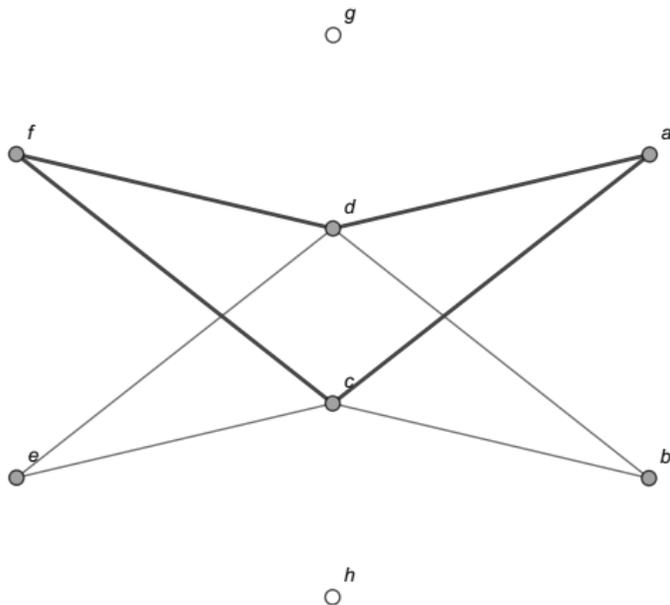
## Solución del ejemplo 5.28

Al excluir los nodos aislados y las aristas utilizadas en ese camino, se forma el subgrafo:



## Solución del ejemplo 5.28

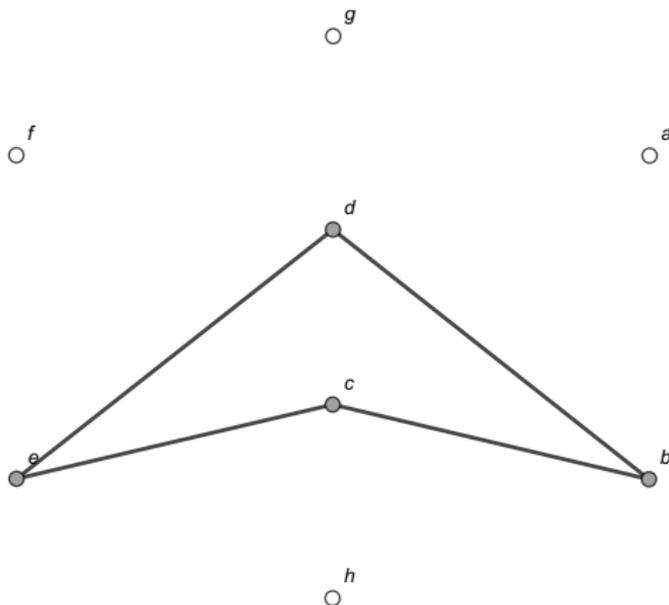
En él, un circuito corresponde a:  $f \rightarrow d \rightarrow a \rightarrow c \rightarrow f$ . Gráficamente:





## Solución del ejemplo 5.28

En este subgrafo, una trayectoria circuital que comienza con un nodo del circuito de la iteración anterior es:  $d \rightarrow e \rightarrow c \rightarrow b \rightarrow d$ . Visualmente:



## Solución del ejemplo 5.28

Al sustraer los lados del camino y los vértices aislados, la figura queda vacía. Luego, un circuito de *Euler* se halla así:

$$1. h \rightarrow g \rightarrow \boxed{f} \rightarrow e \rightarrow h \rightarrow b \rightarrow a \rightarrow g \rightarrow d \rightarrow c \rightarrow h$$

$$2. h \rightarrow g \rightarrow \boxed{f \rightarrow d \rightarrow a \rightarrow c \rightarrow f} \rightarrow e \rightarrow h \rightarrow b \rightarrow a \rightarrow g \rightarrow d \rightarrow c \rightarrow h$$

$$3. h \rightarrow g \rightarrow f \rightarrow \boxed{d} \rightarrow a \rightarrow c \rightarrow f \rightarrow e \rightarrow h \rightarrow b \rightarrow a \rightarrow g \rightarrow d \rightarrow c \rightarrow h$$

$$4. h \rightarrow g \rightarrow f \rightarrow \boxed{d \rightarrow e \rightarrow c \rightarrow b \rightarrow d} \rightarrow a \rightarrow c \rightarrow f \rightarrow e \rightarrow h \rightarrow b \rightarrow a \rightarrow g \rightarrow d \rightarrow c \rightarrow h$$

$$5. h \rightarrow g \rightarrow f \rightarrow d \rightarrow e \rightarrow c \rightarrow b \rightarrow d \rightarrow a \rightarrow c \rightarrow f \rightarrow e \rightarrow h \rightarrow b \rightarrow a \rightarrow g \rightarrow d \rightarrow c \rightarrow h$$

## Solución del ejemplo 5.28

Una ruta de *Euler* se determina al omitir la arista temporal  $h \bullet\bullet g$  en la última trayectoria:

$$g \rightarrow f \rightarrow d \rightarrow e \rightarrow c \rightarrow b \rightarrow d \rightarrow a \rightarrow c \rightarrow f \rightarrow e \rightarrow h \rightarrow b \rightarrow a \\ \rightarrow g \rightarrow d \rightarrow c \rightarrow h$$

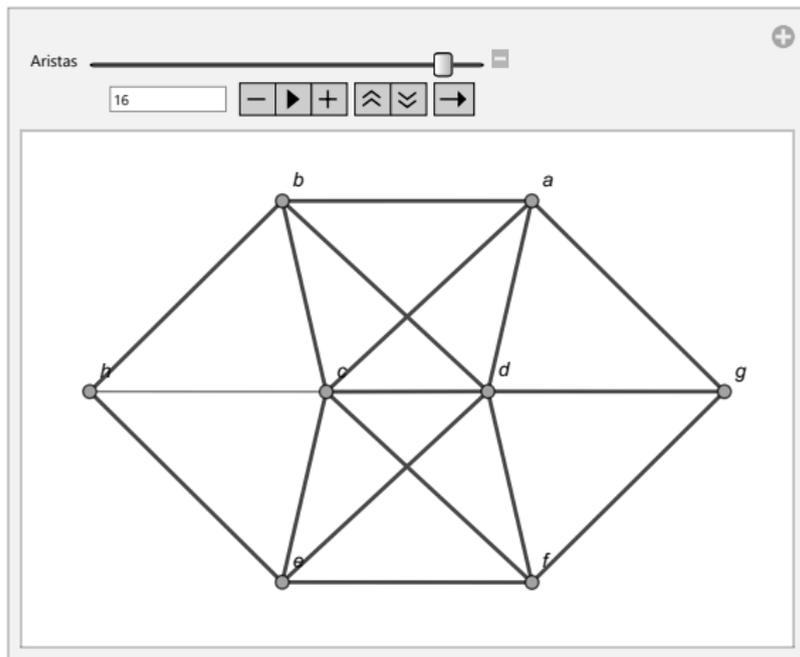
La instrucción `AnimarGrafo` facilita comprobar que el camino anterior pasa una única vez por todos los lados del grafo original:

**In[ ] :=**

```
grafo = Grafo[{{c, e}, {d, e}, {d, a}, {c, d}, {e, h},
{c, h}, {c, f}, {a, b}, {a, c}, {b, c}, {b, d}, {b, h},
{d, f}, {e, f}, {f, g}, {a, g}, {g, d}}];
AnimarGrafo[grafo, Partition[{g, f, d, e, c, b, d, a, c, f,
e, h, b, a, g, d, c, h}, 2, 1]]
```

## Solución del ejemplo 5.28

Out[ ] =



## Solución del ejemplo 5.28

El comando `RutaEuler` del paquete **VilCretas**, retorna un camino de *Euler* distinto al encontrado en los pasos previos, veamos:

**In[ ]** :=

`RutaEuler[grafo]`

**Out[ ]** =

$\{\{h, b\}, \{b, a\}, \{a, c\}, \{c, b\}, \{b, d\}, \{d, c\}, \{c, e\}, \{e, d\}, \{d, a\},$   
 $\{a, g\}, \{g, f\}, \{f, c\}, \{c, h\}, \{h, e\}, \{e, f\}, \{f, d\}, \{d, g\}\}$



## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-124.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-124.zip)

# Circuitos de Hamilton

Los circuitos de *Hamilton* constituyen otro tema esencial. No se abordará en este libro ningún teorema que caracterice la existencia de un circuito de *Hamilton* sobre un grafo conexo, como sí se hizo para los circuitos de *Euler*, sin embargo, el teorema expuesto a continuación, establece algunas propiedades suficientes pero no necesarias que garantizan la existencia de un circuito de *Hamilton*.

## Teorema 5.6

### Theorem (5.6)

Sea  $G = (V, E)$  un grafo simple y conexo, con  $n$  nodos y  $p$  lados,  $n \geq 3$ , luego:

- 1 [Propiedad de la cantidad de aristas] Si  $p \geq \frac{n^2 - 3n + 6}{2}$  entonces  $G$  contiene un circuito de Hamilton.
- 2 [Propiedad de Dirac] Si para todo vértice  $a$  de  $V$ ,  $\theta(a) \geq \frac{n}{2}$  entonces  $G$  posee un circuito hamiltoniano.
- 3 [Propiedad de Ore] Si para todo par de vértices no adyacentes  $a$  y  $b$  de  $V$ , se cumple  $\theta(a) + \theta(b) \geq n$  entonces  $G$  tiene un circuito de Hamilton.

## Comentario sobre el teorema 43

Ninguna de las propiedades enunciadas en el teorema 43 es de caracterización, es decir, si un grafo simple y conexo  $G$  no cumple alguna de ellas, no es posible concluir que  $G$  no contiene un circuito de *Hamilton*. Inclusive si  $G$  no satisface ninguna de las propiedades, la única respuesta válida, consiste en afirmar la ausencia de un criterio para determinar la existencia de un circuito hamiltoniano. Esto quiere decir, que las propiedades enunciadas en el teorema 43 son suficientes pero no son necesarias, dado que si alguna de ellas o todas son falsas no hay criterio para afirmar que el grafo  $G$  no contiene un circuito de *Hamilton*. En el teorema 43 el número de vértices  $n$  de  $G$  debe ser mayor o igual a 3, pues para que un grafo simple y conexo  $G$  posea un circuito, la cantidad mínima de nodos que debe contener es igual a 3.

## Comentario sobre el teorema 43

En la práctica se recomienda al alumno verificar el valor de verdad de las propiedades del teorema 43, en el orden en cómo allí aparecen. Se verifica primero si el número de lados  $p$  del grafo  $G$  cumple con la desigualdad  $p \geq \frac{n^2 - 3n + 6}{2}$ . Si esto es falso, se corrobora la propiedad de *Dirac*, es decir, el grado de cada vértice tiene que ser mayor o igual a la mitad del número de nodos  $n$  de  $G$ . Si a su vez esta condición es falsa, se comprueba la propiedad de *Ore*. Esta última es la más laboriosa de verificar, pues implica encontrar todas las parejas posibles de nodos no adyacentes, donde a su vez, se debe satisfacer que la suma de sus valencias es mayor o igual a la cantidad de vértices  $n$  del grafo. En algunas ocasiones este número de parejas de nodos no adyacentes puede ser relativamente elevado. Finalmente, cabe aclarar que con solo el cumplimiento de una de las propiedades enunciadas en el teorema 43, se asegura la existencia de un circuito de *Hamilton*.

### Example (5.29)

Determine si el grafo del ejemplo 38 satisface las propiedades del teorema 43. Halle un circuito hamiltoniano en caso de existir.

## Solución del ejemplo 5.29

El grafo del ejemplo 38 es simple y conexo. La cantidad  $p$  de lados de este grafo se puede hallar recurriendo al teorema 9:

	$a$	$b$	$c$	$d$	$e$	$f$	$g$	$h$	$p$
Grado o valencia	4	4	4	4	6	4	4	4	$\sum \theta(v) \div 2 = \frac{34}{2} = 17$

(5)

Además,  $n = 8$ , de donde:

$$\frac{n^2 - 3n + 6}{2} = \frac{8^2 - 3 \cdot 8 + 6}{2} = 23$$

## Solución del ejemplo 5.29

La primera propiedad del teorema 43 no se satisface:

$$p = 17 \not\geq \frac{n^2 - 3n + 6}{2} = 23$$

Como ya se mencionó, a pesar de no cumplirse esta condición, esto no implica que el grafo no contenga circuitos hamiltonianos.

Con respecto a la propiedad de *Dirac*, sí se satisface, pues al observar la tabla de grados 5, en todos los casos la valencia de los nodos es mayor o igual a  $\frac{n}{2} = \frac{8}{2} = 4$ . Esto es suficiente para garantizar la existencia de un circuito de *Hamilton* sobre el grafo.

## Solución del ejemplo 5.29

Con relación a la propiedad de *Ore*, se deben extraer todas las parejas de vértices no adyacentes y sumar sus valencias:

Nodos no adyacentes	Suma de las valencias
$a$ y $b$	$\theta(a) + \theta(b) = 4 + 4 = 8$
$a$ y $c$	$\theta(a) + \theta(c) = 4 + 4 = 8$
$a$ y $f$	$\theta(a) + \theta(f) = 4 + 4 = 8$
$b$ y $d$	$\theta(b) + \theta(d) = 4 + 4 = 8$
$b$ y $g$	$\theta(b) + \theta(g) = 4 + 4 = 8$
$c$ y $g$	$\theta(c) + \theta(g) = 4 + 4 = 8$
$c$ y $h$	$\theta(c) + \theta(h) = 4 + 4 = 8$
$d$ y $f$	$\theta(d) + \theta(f) = 4 + 4 = 8$
$d$ y $h$	$\theta(d) + \theta(h) = 4 + 4 = 8$
$e$ y $h$	$\theta(e) + \theta(h) = 6 + 4 = 10$
$f$ y $g$	$\theta(e) + \theta(h) = 6 + 4 = 10$



## Solución del ejemplo 5.29

Este circuito de *Hamilton* claramente no es un circuito de *Euler* al no recorrer todas las aristas del grafo. En este sentido, no necesariamente un circuito de *Hamilton* será uno euleriano y viceversa, no necesariamente un circuito de *Euler* será uno hamiltoniano. Algunas veces, podría suceder que un circuito de *Hamilton* sobre un grafo corresponda a un circuito de *Euler* o viceversa.

## Nota

En este texto no se abordará ningún algoritmo para encontrar un circuito de *Hamilton* sobre un grafo que lo posea. Normalmente esta tarea es más sencilla si se le compara con el problema de hallar un circuito euleriano, por lo que, se recurrirá esencialmente al sentido común para la construcción de circuitos hamiltonianos.

## Solución del ejemplo 5.29

La librería **VilCretas** integra también, las instrucciones `CircuitoHamiltonQ` y `CircuitosHamilton`, similares a los comandos `CircuitoEulerQ` y `CircuitosEuler`. `CircuitoHamiltonQ` devuelve `True` si el grafo evaluado contiene circuitos de *Hamilton* y de lo contrario, retorna `False`. `CircuitosHamilton` determina circuitos hamiltonianos en el grafo especificado como argumento. Por ejemplo, en este ejercicio:

```
In[ ] :=
```

```
grafo = Grafo[{{a, d}, {a, e}, {a, g}, {a, h}, {b, e},  
{b, f}, {c, d}, {c, e}, {d, e}, {d, g}, {e, f}, {g, h},  
{g, e}, {b, c}, {f, h}, {h, b}, {f, c}}];  
CircuitoHamiltonQ[grafo]  
CircuitosHamilton[grafo, 3]
```

## Solución del ejemplo 5.29

Out[ ] =

True

$$\{ \{ a \bullet \bullet d, d \bullet \bullet c, c \bullet \bullet e, e \bullet \bullet f, f \bullet \bullet b, b \bullet \bullet h, h \bullet \bullet g, g \bullet \bullet a \}, \\ \{ a \bullet \bullet g, g \bullet \bullet d, d \bullet \bullet c, c \bullet \bullet e, e \bullet \bullet f, f \bullet \bullet b, b \bullet \bullet h, h \bullet \bullet a \}, \\ \{ a \bullet \bullet d, d \bullet \bullet c, c \bullet \bullet e, e \bullet \bullet b, b \bullet \bullet f, f \bullet \bullet h, h \bullet \bullet g, g \bullet \bullet a \} \}$$

El 3 en CircuitosHamilton solicita a *Wolfram Mathematica* tres circuitos hamiltonianos que el software para este ejemplo, sí fue capaz de hallar. En ciertas ocasiones, dependiendo del grafo, la instrucción CircuitosHamilton devolverá un número menor de circuitos respecto a lo requerido por el usuario.



## Descargue un archivo

<https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-125.zip>



## Explicación en video

<https://youtu.be/cZrOfREwIR8>



## Explicación en video

<https://youtu.be/C1QDNgDFyK4>

### Example (5.30)

Halle el valor de verdad de las propiedades enunciadas en el teorema 43 sobre el grafo del ejemplo 39. Encuentre, si existe, un circuito de *Hamilton*.

## Solución del ejemplo 5.30

El grafo del enunciado es simple y conexo. La cantidad  $p$  de aristas del grafo según el teorema 9, se calcula así:

	$a$	$b$	$c$	$d$	$e$	$f$	$g$	$h$	$p$
Grado o valencia	4	4	4	6	6	4	4	2	$\sum \theta(v) \div 2 = \frac{34}{2} = 17$

Además,  $n = 8$ , luego:

$$\frac{n^2 - 3n + 6}{2} = \frac{8^2 - 3 \cdot 8 + 6}{2} = 23$$

De donde:

$$p = 17 \neq \frac{n^2 - 3n + 6}{2} = 23$$

La primera propiedad del teorema 43 es falsa.

## Solución del ejemplo 5.30

La propiedad de *Dirac* también es falsa, pues al seleccionar el nodo  $h$  que posee el menor de los grados,  $\theta(h) = 2 \not\geq \frac{n}{2} = \frac{8}{2} = 4$ . Con solo que exista un vértice que contradiga la condición de *Dirac*, ya se tiene un contraejemplo que permite confirmar que la propiedad no se satisface. Con respecto a la propiedad de *Ore*, al elegir dos vértices no adyacentes con valencias mínimas, por ejemplo, los nodos  $f$  y  $h$ , se tiene que:  $\theta(f) + \theta(h) = 6 \not\geq n = 8$ . Al haber encontrado un contraejemplo, se concluye que el valor de verdad de la propiedad de *Ore* es falso.

## Nota

El alumno debe apreciar que corroborar la falsedad de las propiedades de *Dirac* y *Ore* es más sencillo que verificar su cumplimiento, pues si estas propiedades son falsas basta con hallar un contraejemplo, mientras que si son verdaderas se debe demostrar la veracidad sobre todos los casos posibles. En el contexto de este libro, se recurrirá al uso del software *Mathematica* a través de la instrucción `CircuitoHamiltonQ`, con el objetivo de estudiar la existencia de circuitos de *Hamilton* si no se cuenta con un criterio teórico.

## Solución del ejemplo 5.30

En este ejemplo, las tres propiedades del teorema 43 son falsas, pese a ello, lo único que podemos afirmar es que no existe ningún criterio para avalar la existencia de un circuito hamiltoniano.

En este ejercicio:

**In[ ] :=**

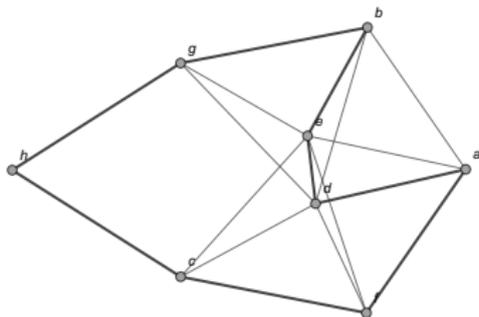
```
grafo = Grafo[{{a, e}, {d, f}, {e, f}, {a, d}, {d, e},  
{h, c}, {h, g}, {a, f}, {b, d}, {b, e}, {b, g}, {c, d},  
{c, e}, {c, f}, {d, g}, {e, g}, {a, b}}];  
CircuitoHamiltonQ[grafo]
```

**Out[ ] =**

True

## Solución del ejemplo 5.30

La salida indica que el grafo sí contiene circuitos de *Hamilton*. Al analizar por sentido común, una trayectoria circuital que pasa una única vez por todos los nodos, se obtiene:  $h \rightarrow g \rightarrow b \rightarrow e \rightarrow d \rightarrow a \rightarrow f \rightarrow c \rightarrow h$ . Gráficamente:



El comando `CircuitosHamilton` también pudo haber contribuido en la construcción de uno o varios circuitos hamiltonianos.



## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-126.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-126.zip)

### Example (5.31)

Determine si se cumplen las propiedades del teorema 43 en el grafo del ejemplo 41. Halle un circuito hamiltoniano, si existe.

### Solución del ejemplo 5.31

Este grafo es simple y conexo. El número  $p$  de lados del grafo, de acuerdo con el teorema 9 es:

	$a$	$b$	$c$	$d$	$e$	$f$	$g$	$p$	(6)
Grado o valencia	5	5	4	6	4	6	4	$\sum \theta(v) \div 2 = \frac{34}{2} = 17$	

Por otra parte:

$$\frac{n^2 - 3n + 6}{2} = \frac{7^2 - 3 \cdot 7 + 6}{2} = 17$$

Luego, la propiedad 1 del teorema 43 es verdadera:

$$p = 17 \geq \frac{n^2 - 3n + 6}{2} = 17$$

## Solución del ejemplo 5.31

En la tabla 6 de valencias se aprecia que cada uno de los grados (4, 5 o 6) de los vértices del grafo es mayor que  $\frac{n}{2} = \frac{7}{2} = 3,5$ , cumpliéndose la propiedad de *Dirac*.

Con respecto a la propiedad de Ore, la siguiente tabla integra todas las parejas posibles de nodos no adyacentes:

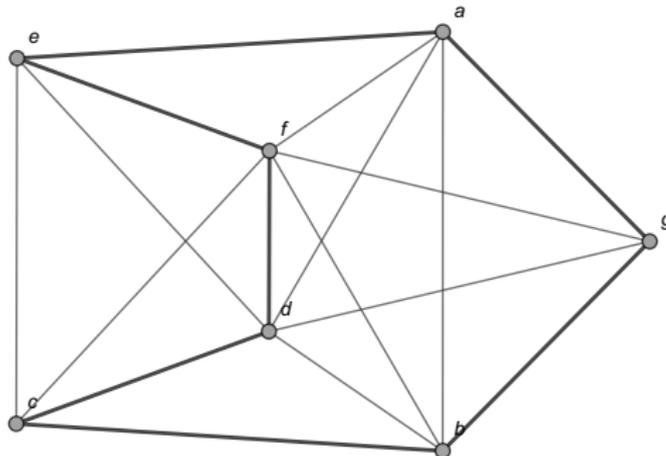
Vértices no adyacentes	Suma de las valencias
$a$ y $c$	$\theta(a) + \theta(c) = 5 + 4 = 9$
$b$ y $e$	$\theta(b) + \theta(e) = 5 + 4 = 9$
$c$ y $g$	$\theta(c) + \theta(g) = 4 + 4 = 8$
$e$ y $g$	$\theta(e) + \theta(g) = 4 + 4 = 8$

La suma de sus valencias siempre da como resultado un número mayor o igual a la cantidad de vértices  $n = 7$ .

Se concluye que todas las propiedades del teorema 43 son verdaderas.

## Solución del ejemplo 5.31

Si se desea elaborar un circuito de *Hamilton* se debe pensar en un camino circuital que pase una única vez por todos los vértices del grafo, por ejemplo:  $e \rightarrow f \rightarrow d \rightarrow c \rightarrow b \rightarrow g \rightarrow a \rightarrow e$ . Visualmente:





## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-127.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-127.zip)

### Example (5.32)

¿El grafo del ejemplo 42 contiene circuitos hamiltonianos?

### Solución del ejemplo 5.32

El grafo se caracteriza por ser simple y conexo. La cantidad  $p$  de aristas del grafo viene dada por:

	$a$	$b$	$c$	$d$	$e$	$f$	$g$	$h$	$p$
Grado o valencia	4	4	6	6	4	4	3	3	$\sum \theta(v) \div 2 = \frac{34}{2} = 17$

(7)

Por otro lado:

$$\frac{n^2 - 3n + 6}{2} = \frac{8^2 - 3 \cdot 8 + 6}{2} = 23$$

La primera propiedad del teorema 43 es falsa.

Con respecto a la propiedad de *Dirac*, al tomar el grado mínimo 3 de la tabla 7, se obtienen dos contraejemplos, pues

$$\theta(g) = \theta(h) = 3 \not\geq \frac{n}{2} = \frac{8}{2} = 4.$$

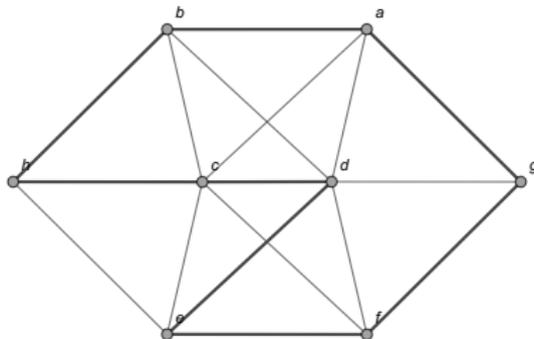
### Solución del ejemplo 5.32

Con relación a la propiedad de *Ore*, también es falsa. Los vértices  $g$  y  $h$  con valencias mínimas iguales a 3, no son adyacentes y la suma de sus grados da como resultado 6. Este valor no es mayor o igual al número  $n$  de vértices del grafo,  $n = 8$ .

Las tres propiedades el teorema 43 tienen un valor de verdad falso, por lo que, no hay criterio para garantizar la existencia de un circuito hamiltoniano, sin embargo, al estudiar el grafo buscando un circuito que pase una única vez por cada uno de los nodos, se concluye que el grafo sí contiene circuitos de *Hamilton*. Un ejemplo de una trayectoria de este tipo es:  $b \rightarrow h \rightarrow c \rightarrow d \rightarrow e \rightarrow f \rightarrow g \rightarrow a \rightarrow b$ .

## Solución del ejemplo 5.32

Gráficamente:



Además, las instrucciones `CircuitoHamiltonQ` y `CircuitosHamilton` podrían ser utilizadas por el alumno para comprobar la existencia de circuitos hamiltonianos en el grafo y hallar uno o varios caminos de esa naturaleza.

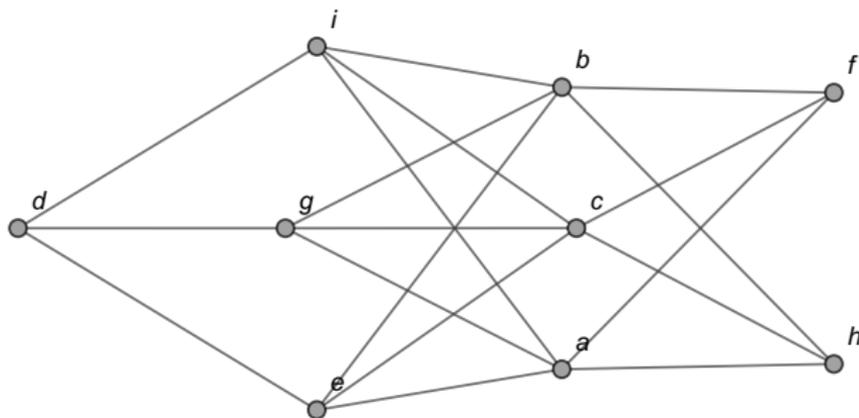


## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-128.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-128.zip)

### Example (5.33)

Determine la existencia de un circuito de *Hamilton* sobre el grafo compartido a continuación.



## Solución del ejemplo 5.33

El grafo es simple y conexo. El número  $p$  de lados se encuentra como sigue:

	$a$	$b$	$c$	$d$	$e$	$f$	$g$	$h$	$i$	$p$
Grado o valencia	5	5	5	3	4	3	4	3	4	$\sum \theta(v) \div 2 = \frac{36}{2} = 18$

(8)

Luego:

$$\frac{n^2 - 3n + 6}{2} = \frac{9^2 - 3 \cdot 9 + 6}{2} = 30$$

La propiedad 1 del teorema 43 es falsa:

$$p = 18 \not\geq \frac{n^2 - 3n + 6}{2} = 30$$

Al considerar la valencia mínima 3 en la tabla 8, se tiene que  $\theta(d) = \theta(f) = \theta(h) = 3 \not\geq \frac{n}{2} = \frac{9}{2} = 4,5$ . La propiedad de *Dirac*, también tiene un valor de verdad falso.

## Solución del ejemplo 5.33

En el grafo los vértices  $d$  y  $f$  presentan un contraejemplo sobre el cumplimiento de la propiedad de Ore, los nodos son no adyacentes y  $\theta(d) + \theta(f) = 6 \not\geq n = 9$ .

Las tres propiedades del teorema 43 son falsas. Al intentar buscar manualmente un recorrido que pase una única vez por cada uno de los vértices del grafo, la tarea se torna imposible. Esto se puede corroborar usando el comando `CircuitoHamiltonQ`:

```
In[ ] :=
```

```
grafo = Grafo[{{c, f}, {a, e}, {a, f}, {a, g}, {a, h},  
{a, i}, {b, e}, {b, f}, {b, g}, {b, h}, {b, i}, {c, e},  
{c, g}, {c, h}, {c, i}, {d, e}, {d, g}, {d, i}}];  
CircuitoHamiltonQ[grafo]
```

```
Out[ ] =
```

```
False
```

## Solución del ejemplo 5.33

El False indica que el grafo no posee circuitos de *Hamilton*.

Las propiedades del teorema 43, se pueden comprobar de forma automática sobre un grafo simple y conexo, mediante la sentencia `ExisteCircuitoHamilton` del paquete **VilCretas**. Esta instrucción recibe el grafo y muestra como salida el valor de verdad de cada una de las propiedades del teorema 43, ofreciendo un contraejemplo en caso de que alguna de ellas sea falsa. Adicionalmente, comparte el valor de verdad sobre la existencia de un circuito hamiltoniano si todas las propiedades son falsas. En este ejercicio, al emplear `ExisteCircuitoHamilton`:

```
In[ ] :=
```

```
ExisteCircuitoHamilton[grafo]
```

## Solución del ejemplo 5.33

**Out[ ] =**

No se satisface la propiedad de Ore, un contraejemplo ocurre en:

$$\text{Valencia}[c] + \text{Valencia}[d] = 8 < n = 9$$

No se cumple la propiedad de Dirac, un contraejemplo ocurre en:

$$\text{Valencia}[f] = 3 < n/2 = 9/2$$

No se satisface la propiedad de la cantidad de aristas,

$$m = 18 < (n^2 - 3n + 6)/2 = 30$$

No hay criterio, sin embargo, el valor lógico sobre la existencia de un circuito de Hamilton en el grafo es: False

## Nota

El estudiante debe apreciar que en los ejemplos 45 y 47, las tres propiedades del teorema 43, también resultaron ser falsas, pero por el contrario, los grafos de esos ejercicios sí se caracterizaron por tener circuitos hamiltonianos. De allí, que se ha venido insistiendo en el hecho de que las propiedades del teorema 43 son suficientes pero no necesarias para garantizar la existencia de este tipo de circuitos.



## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-129.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-129.zip)



## Explicación en video

<https://youtu.be/rLmbmvMI1Sw>

### Example (5.34)

¿Cuáles condiciones debe cumplir un grafo completo  $K_n$  para poseer circuitos hamiltonianos?

## Solución del ejemplo 5.34

En el ejemplo 13 se halló una fórmula que calcula el número  $p$  de aristas de  $K_n$  :

$$p = \frac{n(n-1)}{2}$$

Luego, si se piensa en satisfacer la primera propiedad del teorema 43, sería suficiente para garantizar la existencia de un circuito de *Hamilton*, es decir, se asume:

$$\frac{n(n-1)}{2} \geq \frac{n^2 - 3n + 6}{2} \Rightarrow n(n-1) \geq n^2 - 3n + 6$$

$$\Rightarrow n^2 - n \geq n^2 - 3n + 6 \Rightarrow 3n - n \geq 6 \Rightarrow 2n \geq 6 \Rightarrow n \geq 3$$

También, se observa que si  $n \geq 3$  se infiere la desigualdad de partida. En consecuencia: “ $K_n$  tiene un circuito de *Hamilton* sí y solo sí  $n \geq 3$ ”.

## Nota

Al analizar los grafos  $K_1$  y  $K_2$  mostrados en la figura 9, se intuye que el menor orden de  $K_n$  donde pueden ocurrir circuitos, se da cuando  $n = 3$ , por lo que, si  $K_n$  posee rutas circuitales siempre contendrá circuitos hamiltonianos.

①

(a) Grafo  $K_1$ (b) Grafo  $K_2$ 

Figura: Grafos completos de orden 1 y 2

### Example (5.35)

¿Qué condiciones se deben satisfacer en un grafo bipartito completo  $K_{n,m}$  para contener circuitos de *Hamilton*?

## Solución del ejemplo 5.35

Se utilizará la propiedad de *Dirac* con el objetivo de encontrar estas condiciones. El número de vértices en  $K_{n,m}$  es igual a  $n + m$  y ya se ha comentado en ejemplos anteriores, que en un grafo bipartito completo existen únicamente dos tipos de grados: los  $n$  vértices izquierdos ( $v_i^I$ ,  $1 \leq i \leq n$ ,  $i \in \mathbb{N}$ ) tienen como valencia el número  $m$  y los  $m$  nodos derechos ( $v_j^D$ ,  $1 \leq j \leq m$ ,  $j \in \mathbb{N}$ ) poseen como grado el valor de  $n$ . Hay, por lo tanto, dos casos a considerar:

$$\textcircled{1} \quad \theta(v_i^I) \geq \frac{n+m}{2} \Rightarrow m \geq \frac{n+m}{2} \Rightarrow 2m \geq n+m \Rightarrow 2m - m \geq n \\ \Rightarrow m \geq n$$

$$\textcircled{2} \quad \theta(v_j^D) \geq \frac{n+m}{2} \Rightarrow n \geq \frac{n+m}{2} \Rightarrow 2n \geq n+m \Rightarrow 2n - n \geq m \\ \Rightarrow n \geq m$$

## Solución del ejemplo 5.35

Para que se cumpla la propiedad de *Dirac*:

$$\theta(v_i^I) \geq \frac{n+m}{2} \text{ y } \theta(v_j^D) \geq \frac{n+m}{2}$$

A razón de ello y lo ya resuelto:

$$m \geq n \wedge n \geq m \Rightarrow n = m$$

Además, si  $m \geq n$  se obtiene  $\theta(v_i^I) \geq \frac{n+m}{2}$  y si  $n \geq m$  se deduce  $\theta(v_j^D) \geq \frac{n+m}{2}$ . En conclusión: " $K_{n,m}$  tiene un circuito de *Hamilton* sí y solo sí  $n = m$ ", claro está, excluyendo a  $K_{1,1}$  que es igual al grafo  $K_2$  mostrado en la subfigura 9b, donde no hay ningún tipo de circuito.

# Comando AgenteViajero

A manera de cultura general es importante señalar al estudiante un problema tradicional relacionado con los circuitos de *Hamilton*, que recibe el nombre de: “problema del agente viajero”. Este consiste en encontrar, en un grafo con pesos, un circuito de *Hamilton* de longitud mínima. Se denomina “agente viajero” pues resuelve la situación, donde se asume que los vértices del grafo representan los lugares que un viajero debe visitar y los pesos de los lados, costos asociados a cada viaje. Luego, el “agente viajero” está interesado en conocer la trayectoria que le permita incurrir en gastos mínimos, visitando todos los sitios hasta llegar nuevamente al punto de partida.

# Comando AgenteViajero

El paquete **VilCretas** facilita el comando `AgenteViajero` que resuelve el problema recibiendo como argumento el grafo ponderado respectivo. Se dejará al lector la tarea de explorar la funcionalidad de esta sentencia. Además del problema del agente viajero es clásico en teoría de grafos, contar con un método que determine la longitud y una ruta, de un camino más corto entre dos vértices en un grafo conexo y ponderado. A continuación, se considera este interesante tópico.

# Algoritmo del camino más corto

Prof. Enrique Vílchez Quesada

Universidad Nacional de Costa Rica

# Algoritmo del camino más corto

Encontrar la longitud de un camino más corto entre dos nodos en un grafo conexo con pesos asociados, tiene importantes aplicaciones en distintas teorías que utilizan los grafos como un medio de representación en problemas con cierto rango de complejidad. Un algoritmo bastante conocido que resuelve este problema, se denomina “algoritmo de *Dijkstra*”.

Theorem (5.7. Algoritmo de *Dijkstra*)

Sea  $G = (V, E)$  un grafo conexo y ponderado, con  $M_{AP} = (c_{ij})$  la matriz de adyacencia de pesos de  $G$  en el orden de los elementos de  $V$ . Si se desea obtener la longitud de un camino más corto o de peso mínimo entre dos nodos "a" y "z" de  $G$ , se realizan los siguientes pasos:

- 1 [Inicialización] Sea  $\text{Marca}(a) = 0$  y  $\text{Marca}(v) = \infty, \forall v, v \in V - \{a\}$ . Sea el conjunto  $LPN = V$ .  $LPN$  se llama lista parcial de nodos.
- 2 Si  $z \notin LPN$ , se ha terminado y  $\text{Marca}(z)$  es la longitud de un camino más corto de "a" a "z".
- 3 Se elige un vértice  $w$  de  $LPN$ , donde  $\text{Marca}(w)$  tiene el valor mínimo y se actualiza  $LPN$  como  $LPN = LPN - \{w\}$ .
- 4 La marca de cada nodo  $r$  adyacente a  $w$ , tal que  $r \in LPN$ , se renueva así:

$$\text{Marca}(r) = \text{Min}(\{\text{Marca}(r), \text{Marca}(w) + c_{ij}\})$$

siendo  $c_{ij}$  la entrada de la matriz  $M_{AP}$  que une el vértice  $w$  con el nodo  $r$ . Si  $r \notin LPN$  su marca no se actualiza. Vuelva al paso dos.

# Comentario sobre el teorema 51

Al finalizar la ejecución del algoritmo de *Dijkstra*, la marca de  $z$  corresponde al valor mínimo de la suma de todos los pesos de las aristas que definen una trayectoria de “ $a$ ” a “ $z$ ”. El algoritmo de *Dijkstra*, por consiguiente, no retorna una ruta sino un número real. Sin embargo, realizando algunos ajustes, a través de éste procedimiento se puede determinar también, una trayectoria de longitud más corta, como así se explicará en los ejemplos que prosiguen.

### Example (5.36)

Encuentre la longitud de un camino más corto del vértice  $d$  al nodo  $f$ , en el grafo dado por medio de la siguiente matriz  $A$  de adyacencia de pesos, considerando el orden de sus vértices  $V = \{a, b, c, d, e, f, g, h, i\}$  :

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 2 & 2 & 7 & 1 & 4 \\ 0 & 0 & 0 & 0 & 4 & 2 & 2 & 9 & 8 \\ 0 & 0 & 0 & 0 & 8 & 2 & 7 & 8 & 3 \\ 0 & 0 & 0 & 0 & 2 & 0 & 7 & 0 & 7 \\ 2 & 4 & 8 & 2 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 2 & 7 & 7 & 0 & 0 & 0 & 0 & 0 \\ 1 & 9 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 8 & 3 & 7 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

## Solución del ejemplo 5.36

Anteriormente se mencionó que el comando GrafoMP permite construir un grafo definido por una matriz de adyacencia de pesos. En este ejercicio, al utilizarlo:

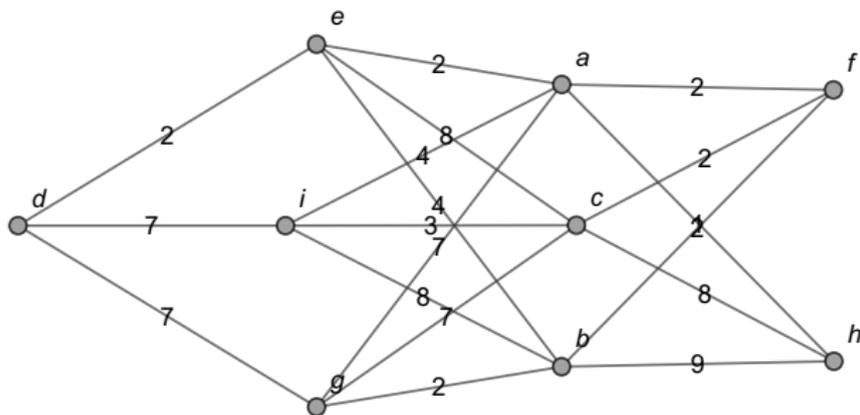
**In[ ] :=**

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 2 & 2 & 7 & 1 & 4 \\ 0 & 0 & 0 & 0 & 4 & 2 & 2 & 9 & 8 \\ 0 & 0 & 0 & 0 & 8 & 2 & 7 & 8 & 3 \\ 0 & 0 & 0 & 0 & 2 & 0 & 7 & 0 & 7 \\ 2 & 4 & 8 & 2 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 2 & 7 & 7 & 0 & 0 & 0 & 0 & 0 \\ 1 & 9 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 8 & 3 & 7 & 0 & 0 & 0 & 0 & 0 \end{pmatrix};$$

```
grafo = GrafoMP[A, {a, b, c, d, e, f, g, h, i},
mostrarpesos -> True]
```

Solución del ejemplo 5.36

Out[ ] =



## Solución del ejemplo 5.36

Se empleará el algoritmo de *Dijkstra* con la intención de hallar una ruta de longitud más corta entre los nodos  $d$  y  $f$ . Para ello, se irá construyendo una tabla cuyas filas contendrán las iteraciones del proceso y cuyas columnas representarán los vértices del grafo. La información dentro de las filas en la tabla, corresponderá a las marcas de cada uno de los nodos en la iteración respectiva. Inicialmente, en la iteración 0, al nodo de partida  $d$  se le asigna marca 0 y a los demás vértices marca igual a  $\infty$ , esto de acuerdo con el teorema 51:

	$a$	$b$	$c$	$d$	$e$	$f$	$g$	$h$	$i$
0	$\infty$	$\infty$	$\infty$	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$

## Solución del ejemplo 5.36

Ahora, se debe seleccionar el vértice con menor marca, que en la iteración 1, ese nodo siempre será igual al vértice de partida ( $d$  en este ejercicio) pues su marca es 0 y 0 es más pequeño que infinito positivo ( $\infty$ ). En el grafo, se observa que los nodos  $e$ ,  $g$  e  $i$  son adyacentes a  $d$ , como consecuencia sus marcas deben actualizarse. La marca de  $e$  es el valor mínimo entre su marca actual  $\infty$  y la suma de la marca de  $d$  y el peso 2 de la arista que une  $d$  con  $e$ , luego:

$$\text{Marca}(e) = \text{Min} \left( \left\{ \underbrace{\infty}_{\text{Marca}(e)}, \underbrace{0}_{\text{Marca}(d)} + \underbrace{2}_{\text{Peso } d \bullet e} \right\} \right) = \text{Min}(\{\infty, 2\}) = 2$$

## Solución del ejemplo 5.36

Para los vértices  $g$  e  $i$  se procede de forma similar:

$$\begin{aligned}
 \text{Marca}(g) &= \text{Min} \left( \left\{ \underbrace{\infty}_{\text{Marca}(g)}, \underbrace{0}_{\text{Marca}(d)} + \underbrace{7}_{\text{Peso } d \bullet g} \right\} \right) = \text{Min}(\{\infty, 7\}) = 7 \\
 \text{Marca}(i) &= \text{Min} \left( \left\{ \underbrace{\infty}_{\text{Marca}(i)}, \underbrace{0}_{\text{Marca}(d)} + \underbrace{7}_{\text{Peso } d \bullet i} \right\} \right) = \text{Min}(\{\infty, 7\}) = 7
 \end{aligned}$$

## Solución del ejemplo 5.36

En la tabla, se agregan los resultados así:

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>
0	$\infty$	$\infty$	$\infty$	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	$\infty$	$\infty$	$\infty$	x	{2, {d}}	$\infty$	{7, {d}}	$\infty$	{7, {d}}

## Solución del ejemplo 5.36

La “x” colocada en la columna de  $d$  señala que dicho vértice ya fue seleccionado y como consecuencia, se elimina de la lista parcial de nodos ( $LPN$ ). En las columnas de los vértices  $e$ ,  $g$  e  $i$  se ha colocado entre llaves, la nueva marca respectiva y el nodo  $d$  del cual se proviene. Llevar el control del vértice de procedencia, ayudará a devolver no solo la longitud de un camino más corto, sino también, una ruta.

En la iteración 2, se elige el nodo  $e$  al tener la menor marca igual a 2 y se elimina  $e$  de la  $LPN$ . En el grafo los vértices adyacentes a  $e$  son  $a$ ,  $b$  y  $c$ , no se toma en cuenta a  $d$  dado que ya fue suprimido de la lista parcial de nodos.

## Solución del ejemplo 5.36

La arista que une  $e$  con  $a$  tiene peso 2, el lado que relaciona  $e$  con  $b$  posee una ponderación igual a 4 y la arista  $e \bullet\bullet c$  tiene peso 8, luego:

$$\begin{aligned}
 \text{Marca}(a) &= \text{Min} \left( \left\{ \underbrace{\infty}_{\text{Marca}(a)}, \underbrace{2}_{\text{Marca}(e)} + \underbrace{2}_{\text{Peso } e \bullet\bullet a} \right\} \right) = \text{Min}(\{\infty, 4\}) = 4 \\
 \text{Marca}(b) &= \text{Min} \left( \left\{ \underbrace{\infty}_{\text{Marca}(b)}, \underbrace{2}_{\text{Marca}(e)} + \underbrace{4}_{\text{Peso } e \bullet\bullet b} \right\} \right) = \text{Min}(\{\infty, 6\}) = 6 \\
 \text{Marca}(c) &= \text{Min} \left( \left\{ \underbrace{\infty}_{\text{Marca}(c)}, \underbrace{2}_{\text{Marca}(e)} + \underbrace{8}_{\text{Peso } e \bullet\bullet c} \right\} \right) = \text{Min}(\{\infty, 10\}) = 10
 \end{aligned}$$

## Solución del ejemplo 5.36

En la tabla, se agrega una fila más, con los datos de la iteración 2 del algoritmo de *Dijkstra*:

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>
0	$\infty$	$\infty$	$\infty$	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	$\infty$	$\infty$	$\infty$	x	{2, { <i>d</i> }}	$\infty$	{7, { <i>d</i> }}	$\infty$	{7, { <i>d</i> }}
2	{4, { <i>e</i> }}	{6, { <i>e</i> }}	{10, { <i>e</i> }}	x	x	$\infty$	{7, { <i>d</i> }}	$\infty$	{7, { <i>d</i> }}

## Solución del ejemplo 5.36

El alumno debe apreciar que las marcas no actualizadas simplemente se arrastran en la nueva iteración. En la iteración 3, de todas las marcas de los nodos pertenecientes a la *LPN*, la menor es igual a 4. Se escoge el vértice *a* y se elimina de la lista parcial de nodos. Ahora, los vértices adyacentes a *a* que se encuentran en la *LPN* son *f*, *g*, *h* e *i*. Luego:

$$\begin{aligned} \text{Marca}(f) &= \text{Min} \left( \left\{ \underbrace{\infty}_{\text{Marca}(f)}, \underbrace{4}_{\text{Marca}(a)} + \underbrace{2}_{\text{Peso } a \bullet f} \right\} \right) = \text{Min}(\{\infty, 6\}) = 6 \\ \text{Marca}(g) &= \text{Min} \left( \left\{ \underbrace{7}_{\text{Marca}(g)}, \underbrace{4}_{\text{Marca}(a)} + \underbrace{7}_{\text{Peso } a \bullet g} \right\} \right) = \text{Min}(\{7, 11\}) = 7 \\ \text{Marca}(h) &= \text{Min} \left( \left\{ \underbrace{\infty}_{\text{Marca}(h)}, \underbrace{4}_{\text{Marca}(a)} + \underbrace{1}_{\text{Peso } a \bullet h} \right\} \right) = \text{Min}(\{\infty, 5\}) = 5 \\ \text{Marca}(i) &= \text{Min} \left( \left\{ \underbrace{7}_{\text{Marca}(i)}, \underbrace{4}_{\text{Marca}(a)} + \underbrace{4}_{\text{Peso } a \bullet i} \right\} \right) = \text{Min}(\{7, 8\}) = 7 \end{aligned}$$

## Solución del ejemplo 5.36

Se adiciona una fila más a la tabla:

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>
0	$\infty$	$\infty$	$\infty$	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	$\infty$	$\infty$	$\infty$	x	{2, {d}}	$\infty$	{7, {d}}	$\infty$	{7, {d}}
2	{4, {e}}	{6, {e}}	{10, {e}}	x	x	$\infty$	{7, {d}}	$\infty$	{7, {d}}
3	x	{6, {e}}	{10, {e}}	x	x	{6, {a}}	{7, {d}}	{5, {a}}	{7, {d}}

## Solución del ejemplo 5.36

En la iteración 4, la menor marca es 5, se elige el vértice  $h$ . Los nodos adyacentes a  $h$  contenidos en la  $LPN$  son  $b$  y  $c$ . De donde:

$$\begin{aligned}
 \text{Marca}(b) &= \text{Min} \left( \left\{ \underbrace{6}_{\text{Marca}(b)}, \underbrace{5}_{\text{Marca}(h)} + \underbrace{9}_{\text{Peso } h \bullet b} \right\} \right) = \text{Min}(\{6, 14\}) = 6 \\
 \text{Marca}(c) &= \text{Min} \left( \left\{ \underbrace{10}_{\text{Marca}(c)}, \underbrace{5}_{\text{Marca}(h)} + \underbrace{8}_{\text{Peso } h \bullet c} \right\} \right) = \text{Min}(\{10, 13\}) = 10
 \end{aligned}$$

## Solución del ejemplo 5.36

Luego:

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>
0	$\infty$	$\infty$	$\infty$	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	$\infty$	$\infty$	$\infty$	x	{2, {d}}	$\infty$	{7, {d}}	$\infty$	{7, {d}}
2	{4, {e}}	{6, {e}}	{10, {e}}	x	x	$\infty$	{7, {d}}	$\infty$	{7, {d}}
3	x	{6, {e}}	{10, {e}}	x	x	{6, {a}}	{7, {d}}	{5, {a}}	{7, {d}}
4	x	{6, {e}}	{10, {e}}	x	x	{6, {a}}	{7, {d}}	x	{7, {d}}

## Solución del ejemplo 5.36

En la iteración 5, la menor marca es 6 y ésta ocurre en los vértices  $b$  y  $f$ . En este sentido, el algoritmo de *Dijkstra* nos permite seleccionar cualquiera de los dos nodos. Naturalmente, por una cuestión de conveniencia, se escogerá el vértice  $f$  que corresponde al nodo de llegada. De acuerdo con el paso dos del teorema 51, esto provocará que se finalice la ejecución del algoritmo de *Dijkstra*. Si se elige el nodo  $b$ , el resultado final seguirá siendo el mismo en cuanto a la longitud de un camino más corto, con la diferencia de requerir una iteración más.

## Solución del ejemplo 5.36

Seleccionando el vértice  $f$  y observando que los nodos adyacentes a  $f$  que pertenecen a la lista parcial de nodos son  $b$  y  $c$ , se tiene:

$$\begin{aligned}
 \text{Marca}(b) &= \text{Min} \left( \left\{ \underbrace{6}_{\text{Marca}(b)}, \underbrace{6}_{\text{Marca}(f)} + \underbrace{2}_{\text{Peso } f \bullet b} \right\} \right) = \text{Min}(\{6, 8\}) = 6 \\
 \text{Marca}(c) &= \text{Min} \left( \left\{ \underbrace{10}_{\text{Marca}(c)}, \underbrace{6}_{\text{Marca}(f)} + \underbrace{2}_{\text{Peso } f \bullet c} \right\} \right) = \text{Min}(\{10, 8\}) = 8
 \end{aligned}$$

## Solución del ejemplo 5.36

La tabla final corresponde a:

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>
0	$\infty$	$\infty$	$\infty$	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	$\infty$	$\infty$	$\infty$	x	{2, {d}}	$\infty$	{7, {d}}	$\infty$	{7, {d}}
2	{4, {e}}	{6, {e}}	{10, {e}}	x	x	$\infty$	{7, {d}}	$\infty$	{7, {d}}
3	x	{6, {e}}	{10, {e}}	x	x	{6, {a}}	{7, {d}}	{5, {a}}	{7, {d}}
4	x	{6, {e}}	{10, {e}}	x	x	{6, {a}}	{7, {d}}	x	{7, {d}}
5	x	{6, {e}}	{8, {f}}	x	x	x	{7, {d}}	x	{7, {d}}

(9)

## Solución del ejemplo 5.36

La marca de  $f$  es igual a 6 y según el teorema 51,  $\text{Marca}(f) = 6$  es la longitud de un camino más corto entre los vértices  $d$  y  $f$ . Si se desea encontrar una ruta de longitud más corta se procede hacia atrás, comenzando en el nodo  $f$  de llegada. En la tabla 9, la columna de  $f$  señala que se proviene del vértice  $a$ . Ahora, en esa misma tabla, la columna de  $a$  indica que se proviene de  $e$ . Luego, la columna del nodo  $e$  señala que se proviene del vértice  $d$ . El recorrido descrito forma una trayectoria de longitud más corta igual a 6 :

$$d \xrightarrow{2} e \xrightarrow{2} a \xrightarrow{2} f$$

Al establecer que esta ruta es de longitud más corta, significa que cualquier otro camino que vaya del nodo  $d$  al vértice  $f$  en el grafo, tendrá la característica de que la suma de los pesos de las aristas utilizadas en esa ruta, dará como resultado un valor mayor o igual a 6.

## Nota

En algunas ocasiones, podrían existir varios caminos de longitud más corta. En este ejercicio, la trayectoria de longitud mínima es única.



**Descargue un archivo**

[https://www.escinf.una.ac.cr/discretas/Archivos/Grafos/  
File-130.zip](https://www.escinf.una.ac.cr/discretas/Archivos/Grafos/File-130.zip)

## Example (5.37)

Determine una ruta de longitud más corta de  $a$  a  $g$ , en el grafo dado por la matriz  $A$  de adyacencia de pesos en el orden de sus vértices

$V = \{a, e, d, f, h, c, g, b\}$  :

$$A = \begin{pmatrix} 0 & 5 & 4 & 1 & 0 & 0 & 0 & 4 \\ 5 & 0 & 2 & 8 & 0 & 8 & 8 & 2 \\ 4 & 2 & 0 & 5 & 0 & 5 & 4 & 2 \\ 1 & 8 & 5 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 \\ 0 & 8 & 5 & 5 & 2 & 0 & 0 & 0 \\ 0 & 8 & 4 & 0 & 1 & 0 & 0 & 1 \\ 4 & 2 & 2 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

## Solución del ejemplo 5.37

Al recurrir al uso del comando GrafoMP se obtiene el grafo de interés:

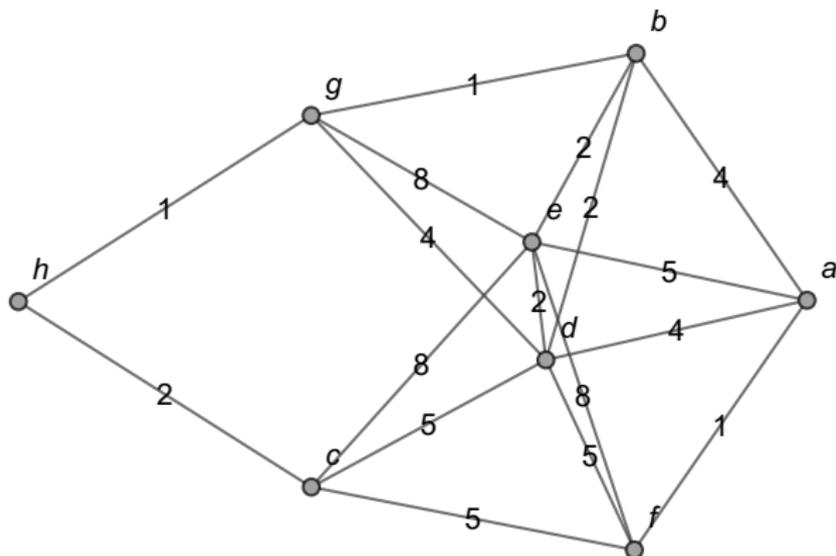
**In[ ] :=**

$$A = \begin{pmatrix} 0 & 5 & 4 & 1 & 0 & 0 & 0 & 4 \\ 5 & 0 & 2 & 8 & 0 & 8 & 8 & 2 \\ 4 & 2 & 0 & 5 & 0 & 5 & 4 & 2 \\ 1 & 8 & 5 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 \\ 0 & 8 & 5 & 5 & 2 & 0 & 0 & 0 \\ 0 & 8 & 4 & 0 & 1 & 0 & 0 & 1 \\ 4 & 2 & 2 & 0 & 0 & 0 & 1 & 0 \end{pmatrix};$$

```
grafo = GrafoMP[A, {a, e, d, f, h, c, g, b},
mostrarpesos -> True]
```

## Solución del ejemplo 5.37

Out[ ] =



### Solución del ejemplo 5.37

En la iteración 0, se asigna a  $a$  una marca igual a 0 y a los demás nodos una marca igual a  $\infty$  :

	$a$	$b$	$c$	$d$	$e$	$f$	$g$	$h$
0	0	$\infty$						

## Solución del ejemplo 5.37

En la iteración 1, se elige a  $a$  y observando el grafo, los vértices adyacentes a  $a$  son  $b$ ,  $d$ ,  $e$  y  $f$ . En consecuencia:

$$\begin{aligned}
 \text{Marca}(b) &= \text{Min} \left( \left\{ \underbrace{\infty}_{\text{Marca}(b)}, \underbrace{0}_{\text{Marca}(a)} + \underbrace{4}_{\text{Peso } a \bullet b} \right\} \right) = \text{Min}(\{\infty, 4\}) = 4 \\
 \text{Marca}(d) &= \text{Min} \left( \left\{ \underbrace{\infty}_{\text{Marca}(d)}, \underbrace{0}_{\text{Marca}(a)} + \underbrace{4}_{\text{Peso } a \bullet d} \right\} \right) = \text{Min}(\{\infty, 4\}) = 4 \\
 \text{Marca}(e) &= \text{Min} \left( \left\{ \underbrace{\infty}_{\text{Marca}(e)}, \underbrace{0}_{\text{Marca}(a)} + \underbrace{5}_{\text{Peso } a \bullet e} \right\} \right) = \text{Min}(\{\infty, 5\}) = 5 \\
 \text{Marca}(f) &= \text{Min} \left( \left\{ \underbrace{\infty}_{\text{Marca}(f)}, \underbrace{0}_{\text{Marca}(a)} + \underbrace{1}_{\text{Peso } a \bullet f} \right\} \right) = \text{Min}(\{\infty, 1\}) = 1
 \end{aligned}$$

## Solución del ejemplo 5.37

Por lo que, la tabla de iteraciones se actualiza así:

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	x	{4, { <i>a</i> }}	$\infty$	{4, { <i>a</i> }}	{5, { <i>a</i> }}	{1, { <i>a</i> }}	$\infty$	$\infty$

En la iteración 2, se selecciona el nodo *f* con marca mínima 1. Los vértices adyacentes a *f* que pertenecen a la lista parcial de nodos son *c*, *d* y *e*. Luego:

$$\begin{aligned} \text{Marca}(c) &= \text{Min} \left( \left\{ \underbrace{\infty}_{\text{Marca}(c)}, \underbrace{1}_{\text{Marca}(f)} + \underbrace{5}_{\text{Peso } f \bullet c} \right\} \right) = \text{Min}(\{\infty, 6\}) = 6 \\ \text{Marca}(d) &= \text{Min} \left( \left\{ \underbrace{4}_{\text{Marca}(d)}, \underbrace{1}_{\text{Marca}(f)} + \underbrace{5}_{\text{Peso } f \bullet d} \right\} \right) = \text{Min}(\{4, 6\}) = 4 \\ \text{Marca}(e) &= \text{Min} \left( \left\{ \underbrace{5}_{\text{Marca}(e)}, \underbrace{1}_{\text{Marca}(f)} + \underbrace{8}_{\text{Peso } f \bullet e} \right\} \right) = \text{Min}(\{5, 9\}) = 5 \end{aligned}$$

## Solución del ejemplo 5.37

La tabla queda de la forma:

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	x	{4, { <i>a</i> }}	$\infty$	{4, { <i>a</i> }}	{5, { <i>a</i> }}	{1, { <i>a</i> }}	$\infty$	$\infty$
2	x	{4, { <i>a</i> }}	{6, { <i>f</i> }}	{4, { <i>a</i> }}	{5, { <i>a</i> }}	x	$\infty$	$\infty$

## Solución del ejemplo 5.37

En la iteración 3, la marca mínima es 4 compartida por los vértices  $b$  y  $d$ . Se tiene la libertad de tomar cualquiera de los dos. Supongamos que se escoge a  $d$ . En el grafo, se aprecia que los nodos de la *LPN* adyacentes a  $d$  son  $b$ ,  $c$ ,  $e$  y  $g$ . Se deben actualizar las marcas de estos vértices:

$$\begin{aligned}
 \text{Marca}(b) &= \text{Min} \left( \left\{ \underbrace{4}_{\text{Marca}(b)}, \underbrace{4}_{\text{Marca}(d)} + \underbrace{2}_{\text{Peso } d \bullet b} \right\} \right) = \text{Min}(\{4, 6\}) = 4 \\
 \text{Marca}(c) &= \text{Min} \left( \left\{ \underbrace{6}_{\text{Marca}(c)}, \underbrace{4}_{\text{Marca}(d)} + \underbrace{5}_{\text{Peso } d \bullet c} \right\} \right) = \text{Min}(\{6, 9\}) = 6 \\
 \text{Marca}(e) &= \text{Min} \left( \left\{ \underbrace{5}_{\text{Marca}(e)}, \underbrace{4}_{\text{Marca}(d)} + \underbrace{2}_{\text{Peso } d \bullet e} \right\} \right) = \text{Min}(\{5, 6\}) = 5 \\
 \text{Marca}(g) &= \text{Min} \left( \left\{ \underbrace{\infty}_{\text{Marca}(g)}, \underbrace{4}_{\text{Marca}(d)} + \underbrace{4}_{\text{Peso } d \bullet g} \right\} \right) = \text{Min}(\{\infty, 8\}) = 8
 \end{aligned}$$

## Solución del ejemplo 5.37

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
0	0	$\infty$						
1	x	{4, {a}}	$\infty$	{4, {a}}	{5, {a}}	{1, {a}}	$\infty$	$\infty$
2	x	{4, {a}}	{6, {f}}	{4, {a}}	{5, {a}}	x	$\infty$	$\infty$
3	x	{4, {a}}	{6, {f}}	x	{5, {a}}	x	{8, {d}}	$\infty$

## Solución del ejemplo 5.37

Ahora, en la iteración 4, la menor marca 4 ocurre en el nodo  $b$ . Los vértices de la lista parcial de nodos adyacentes a  $b$  son  $e$  y  $g$ . Sus marcas se actualizan así:

$$\text{Marca}(e) = \text{Min} \left( \left\{ \underbrace{5}_{\text{Marca}(e)}, \underbrace{4}_{\text{Marca}(b)} + \underbrace{2}_{\text{Peso } b \bullet e} \right\} \right) = \text{Min}(\{5, 6\}) = 5$$

$$\text{Marca}(g) = \text{Min} \left( \left\{ \underbrace{8}_{\text{Marca}(g)}, \underbrace{4}_{\text{Marca}(b)} + \underbrace{1}_{\text{Peso } b \bullet g} \right\} \right) = \text{Min}(\{8, 5\}) = 5$$

## Solución del ejemplo 5.37

Luego, la tabla de iteraciones corresponde a:

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	x	{4, { <i>a</i> }}	$\infty$	{4, { <i>a</i> }}	{5, { <i>a</i> }}	{1, { <i>a</i> }}	$\infty$	$\infty$
2	x	{4, { <i>a</i> }}	{6, { <i>f</i> }}	{4, { <i>a</i> }}	{5, { <i>a</i> }}	x	$\infty$	$\infty$
3	x	{4, { <i>a</i> }}	{6, { <i>f</i> }}	x	{5, { <i>a</i> }}	x	{8, { <i>d</i> }}	$\infty$
4	x	x	{6, { <i>f</i> }}	x	{5, { <i>a</i> }}	x	{5, { <i>b</i> }}	$\infty$

## Solución del ejemplo 5.37

En la iteración 5, la menor marca es 5. Hay dos nodos con esa marca, los vértices  $e$  y  $g$ . Sabemos que si se selecciona a  $g$  se finalizará el proceso, sin embargo, para efectos didácticos, se escogerá el nodo  $e$  con la intención de corroborar junto al lector, que al finalizar la ejecución del algoritmo, la marca de  $g$  continuará siendo 5. Supongamos que se elige el vértice  $e$ . En la *LPN*, los nodos adyacentes a  $e$  son  $c$  y  $g$ :

$$\begin{aligned} \text{Marca}(c) &= \text{Min} \left( \left\{ \underbrace{6}_{\text{Marca}(c)}, \underbrace{5}_{\text{Marca}(e)} + \underbrace{8}_{\text{Peso } e \bullet c} \right\} \right) = \text{Min}(\{6, 13\}) = 6 \\ \text{Marca}(g) &= \text{Min} \left( \left\{ \underbrace{5}_{\text{Marca}(g)}, \underbrace{5}_{\text{Marca}(e)} + \underbrace{8}_{\text{Peso } e \bullet g} \right\} \right) = \text{Min}(\{5, 13\}) = 5 \end{aligned}$$

## Solución del ejemplo 5.37

Ahora, la tabla con esta nueva iteración se actualiza así:

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
0	0	$\infty$						
1	x	{4, {a}}	$\infty$	{4, {a}}	{5, {a}}	{1, {a}}	$\infty$	$\infty$
2	x	{4, {a}}	{6, {f}}	{4, {a}}	{5, {a}}	x	$\infty$	$\infty$
3	x	{4, {a}}	{6, {f}}	x	{5, {a}}	x	{8, {d}}	$\infty$
4	x	x	{6, {f}}	x	{5, {a}}	x	{5, {b}}	$\infty$
5	x	x	{6, {f}}	x	x	x	{5, {b}}	$\infty$

(10)

## Solución del ejemplo 5.37

La menor marca ocurre en  $g$ . El único vértice adyacente en la lista parcial de nodos es  $h$ :

$$\text{Marca}(h) = \text{Min} \left( \left\{ \underbrace{\infty}_{\text{Marca}(h)}, \underbrace{5}_{\text{Marca}(g)} + \underbrace{1}_{\text{Peso } g \bullet h} \right\} \right) = \text{Min}(\{\infty, 6\}) = 6$$

Por lo tanto:

	$a$	$b$	$c$	$d$	$e$	$f$	$g$	$h$
0	0	$\infty$						
1	x	{4, {a}}	$\infty$	{4, {a}}	{5, {a}}	{1, {a}}	$\infty$	$\infty$
2	x	{4, {a}}	{6, {f}}	{4, {a}}	{5, {a}}	x	$\infty$	$\infty$
3	x	{4, {a}}	{6, {f}}	x	{5, {a}}	x	{8, {d}}	$\infty$
4	x	x	{6, {f}}	x	{5, {a}}	x	{5, {b}}	$\infty$
5	x	x	{6, {f}}	x	x	x	{5, {b}}	$\infty$
6	x	x	{6, {f}}	x	x	x	x	{6, {g}}

(11)

### Solución del ejemplo 5.37

Si el proceso se hubiera finalizado en 5 iteraciones, en lugar de 6, la longitud de un camino más corto del vértice  $a$  al nodo  $g$  continuaría siendo 5, como se puede corroborar en la marca de  $g$  de la tabla 10.

En este ejemplo, hay una trayectoria única de longitud 5 de  $a$  a  $g$ . En la columna de  $g$  de la tabla 11, se observa que se proviene de  $b$  y en la columna de  $b$ , se aprecia que se proviene de  $a$ , formándose la ruta de longitud mínima:

$$a \xrightarrow{4} b \xrightarrow{1} g$$



## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-131.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-131.zip)

## Example (5.38)

Halle una ruta de longitud más corta de  $a$  a  $d$  en el grafo cuya matriz de adyacencia de pesos es la matriz  $A$  compartida, en el orden

$V = \{a, b, c, d, e, f, g\}$  :

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 2 & 0 & 0 \\ 1 & 0 & 7 & 0 & 3 & 0 & 5 \\ 0 & 7 & 0 & 8 & 0 & 1 & 2 \\ 0 & 0 & 8 & 0 & 0 & 3 & 0 \\ 2 & 3 & 0 & 0 & 0 & 6 & 4 \\ 0 & 0 & 1 & 3 & 6 & 0 & 4 \\ 0 & 5 & 2 & 0 & 4 & 4 & 0 \end{pmatrix}$$

## Solución del ejemplo 5.38

La sentencia GrafoMP nos retorna el grafo del ejercicio:

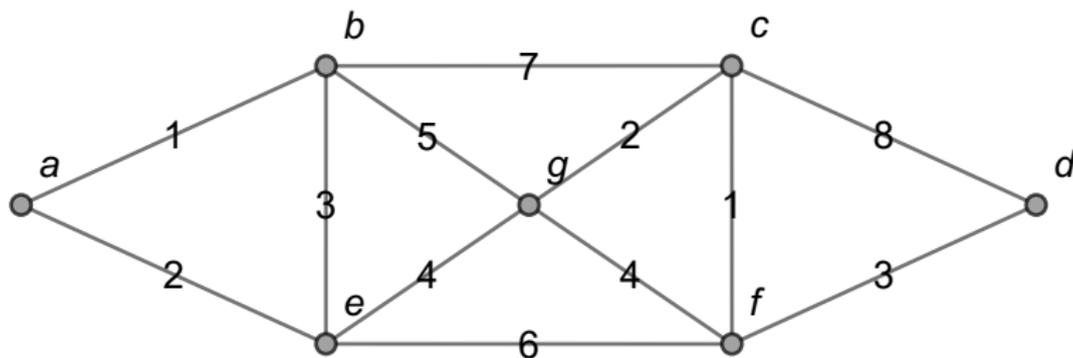
**In[ ] :=**

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 2 & 0 & 0 \\ 1 & 0 & 7 & 0 & 3 & 0 & 5 \\ 0 & 7 & 0 & 8 & 0 & 1 & 2 \\ 0 & 0 & 8 & 0 & 0 & 3 & 0 \\ 2 & 3 & 0 & 0 & 0 & 6 & 4 \\ 0 & 0 & 1 & 3 & 6 & 0 & 4 \\ 0 & 5 & 2 & 0 & 4 & 4 & 0 \end{pmatrix};$$

```
grafo = GrafoMP[A, {a, b, c, d, e, f, g},  
mostrarpesos -> True]
```

Solución del ejemplo 5.38

Out[ ] =



## Solución del ejemplo 5.38

En la iteración 0 del algoritmo de *Dijkstra*, se otorga al nodo  $a$  marca 0 y a los demás vértices marca  $\infty$  :

	$a$	$b$	$c$	$d$	$e$	$f$	$g$
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$

Los nodos adyacentes a  $a$  en el grafo son  $b$  y  $e$  :

$$\begin{aligned}
 \text{Marca}(b) &= \text{Min} \left( \left\{ \underbrace{\infty}_{\text{Marca}(b)}, \underbrace{0}_{\text{Marca}(a)} + \underbrace{1}_{\text{Peso } a \bullet b} \right\} \right) = \text{Min}(\{\infty, 1\}) = 1 \\
 \text{Marca}(e) &= \text{Min} \left( \left\{ \underbrace{\infty}_{\text{Marca}(e)}, \underbrace{0}_{\text{Marca}(a)} + \underbrace{2}_{\text{Peso } a \bullet e} \right\} \right) = \text{Min}(\{\infty, 2\}) = 2
 \end{aligned}$$

## Solución del ejemplo 5.38

Luego:

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	x	$\{1, \{a\}\}$	$\infty$	$\infty$	$\{2, \{a\}\}$	$\infty$	$\infty$

En la iteración 2, la menor marca ocurre en *b*. Los vértices adyacentes a *b* pertenecientes a la lista parcial de nodos son *c*, *e* y *g* :

$$\begin{aligned}
 \text{Marca}(c) &= \text{Min} \left( \left\{ \underbrace{\infty}_{\text{Marca}(c)}, \underbrace{1}_{\text{Marca}(b)} + \underbrace{7}_{\text{Peso } b \bullet c} \right\} \right) = \text{Min}(\{\infty, 8\}) = 8 \\
 \text{Marca}(e) &= \text{Min} \left( \left\{ \underbrace{2}_{\text{Marca}(e)}, \underbrace{1}_{\text{Marca}(b)} + \underbrace{3}_{\text{Peso } b \bullet e} \right\} \right) = \text{Min}(\{2, 4\}) = 2 \\
 \text{Marca}(g) &= \text{Min} \left( \left\{ \underbrace{\infty}_{\text{Marca}(g)}, \underbrace{1}_{\text{Marca}(b)} + \underbrace{5}_{\text{Peso } b \bullet g} \right\} \right) = \text{Min}(\{\infty, 6\}) = 6
 \end{aligned}$$

## Solución del ejemplo 5.38

La tabla de iteraciones se actualiza así:

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	x	{1, { <i>a</i> }}	$\infty$	$\infty$	{2, { <i>a</i> }}	$\infty$	$\infty$
2	x	x	{8, { <i>b</i> }}	$\infty$	{2, { <i>a</i> }}	$\infty$	{6, { <i>b</i> }}

En la iteración 3, se elige el vértice *e* que posee la menor marca. Los nodos adyacentes a *e* del grafo que se encuentran en la *LPN* son *f* y *g* :

$$\begin{aligned}
 \text{Marca}(f) &= \text{Min} \left( \left\{ \underbrace{\infty}_{\text{Marca}(f)}, \underbrace{2}_{\text{Marca}(e)} + \underbrace{6}_{\text{Peso } e \bullet f} \right\} \right) = \text{Min}(\{\infty, 8\}) = 8 \\
 \text{Marca}(g) &= \text{Min} \left( \left\{ \underbrace{6}_{\text{Marca}(g)}, \underbrace{2}_{\text{Marca}(e)} + \underbrace{4}_{\text{Peso } e \bullet g} \right\} \right) = \text{Min}(\{6, 6\}) = 6
 \end{aligned}$$

## Solución del ejemplo 5.38

Se aprecia que en el valor mínimo de la marca de  $g$  ha ocurrido un empate ( $Min(\{6, 6\})$ ), esto significa que se puede provenir no solo del vértice  $b$ , sino también, del nodo  $e$ . Esta información será incluida en la tabla colocando en la columna de  $g$  la expresión  $\{6, \{b, e\}\}$ . Cuando se puede provenir de distintos vértices, algunas veces, esto da cabida a la presencia de varios caminos de longitud mínima. Luego:

	$a$	$b$	$c$	$d$	$e$	$f$	$g$
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	x	$\{1, \{a\}\}$	$\infty$	$\infty$	$\{2, \{a\}\}$	$\infty$	$\infty$
2	x	x	$\{8, \{b\}\}$	$\infty$	$\{2, \{a\}\}$	$\infty$	$\{6, \{b\}\}$
3	x	x	$\{8, \{b\}\}$	$\infty$	x	$\{8, \{e\}\}$	$\{6, \{b, e\}\}$

## Solución del ejemplo 5.38

En la iteración 4 del algoritmo, se selecciona el nodo  $g$  al contener la menor marca igual a 6. En el grafo, los vértices adyacentes a  $g$  de la lista parcial de nodos son  $c$  y  $f$  :

$$\begin{aligned}
 \text{Marca}(c) &= \text{Min} \left( \left\{ \underbrace{8}_{\text{Marca}(c)}, \underbrace{6}_{\text{Marca}(g)} + \underbrace{2}_{\text{Peso } g \bullet c} \right\} \right) = \text{Min}(\{8, 8\}) = 8 \\
 \text{Marca}(f) &= \text{Min} \left( \left\{ \underbrace{8}_{\text{Marca}(f)}, \underbrace{6}_{\text{Marca}(g)} + \underbrace{4}_{\text{Peso } g \bullet f} \right\} \right) = \text{Min}(\{8, 10\}) = 8
 \end{aligned}$$

## Solución del ejemplo 5.38

Se ha dado un empate en la actualización de la marca en el nodo  $c$  ( $Min(\{8, 8\})$ ), como consecuencia se puede provenir también del vértice  $g$ . En la columna de  $c$  esta información será incluida sustituyendo  $\{8, \{b\}\}$  por  $\{8, \{b, g\}\}$  :

	$a$	$b$	$c$	$d$	$e$	$f$	$g$
0	$\boxed{0}$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	x	$\boxed{1}, \{a\}$	$\infty$	$\infty$	$\{2, \{a\}\}$	$\infty$	$\infty$
2	x	x	$\{8, \{b\}\}$	$\infty$	$\boxed{2}, \{a\}$	$\infty$	$\{6, \{b\}\}$
3	x	x	$\{8, \{b\}\}$	$\infty$	x	$\{8, \{e\}\}$	$\boxed{6}, \{b, e\}$
4	x	x	$\{8, \{b, g\}\}$	$\infty$	x	$\{8, \{e\}\}$	x

## Solución del ejemplo 5.38

En la iteración 5, se tiene la libertad de elegir al nodo  $c$ , o bien, al vértice  $f$  al poseer ambos la misma marca mínima. Si se toma al nodo  $c$ , los vértices adyacentes en el grafo incluidos en la  $LPN$  son  $d$  y  $f$ , luego actualizando sus marcas, se tiene:

$$\begin{aligned}
 \text{Marca}(d) &= \text{Min} \left( \left\{ \underbrace{\infty}_{\text{Marca}(d)}, \underbrace{8}_{\text{Marca}(c)} + \underbrace{8}_{\text{Peso } c \bullet d} \right\} \right) = \text{Min}(\{\infty, 16\}) = 16 \\
 \text{Marca}(f) &= \text{Min} \left( \left\{ \underbrace{8}_{\text{Marca}(f)}, \underbrace{8}_{\text{Marca}(c)} + \underbrace{1}_{\text{Peso } c \bullet f} \right\} \right) = \text{Min}(\{8, 9\}) = 8
 \end{aligned}$$

## Solución del ejemplo 5.38

Por consiguiente:

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	x	{1, {a}}	$\infty$	$\infty$	{2, {a}}	$\infty$	$\infty$
2	x	x	{8, {b}}	$\infty$	{2, {a}}	$\infty$	{6, {b}}
3	x	x	{8, {b}}	$\infty$	x	{8, {e}}	{6, {b, e}}
4	x	x	{8, {b, g}}	$\infty$	x	{8, {e}}	x
5	x	x	x	{16, {c}}	x	{8, {e}}	x

## Solución del ejemplo 5.38

En la iteración 6, se selecciona el vértice  $f$  con marca mínima igual a 8. El nodo  $d$  es el único vértice adyacente a  $f$  de la lista parcial de nodos. Su marca se actualiza así:

$$\text{Marca}(d) = \text{Min} \left( \left\{ \underbrace{16}_{\text{Marca}(d)}, \underbrace{8}_{\text{Marca}(f)} + \underbrace{3}_{\text{Peso } f \bullet d} \right\} \right) = \text{Min}(\{16, 11\}) = 11$$

La nueva tabla de iteraciones corresponde a:

	$a$	$b$	$c$	$d$	$e$	$f$	$g$
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	x	{1, {a}}	$\infty$	$\infty$	{2, {a}}	$\infty$	$\infty$
2	x	x	{8, {b}}	$\infty$	{2, {a}}	$\infty$	{6, {b}}
3	x	x	{8, {b}}	$\infty$	x	{8, {e}}	{6, {b, e}}
4	x	x	{8, {b, g}}	$\infty$	x	{8, {e}}	x
5	x	x	x	{16, {c}}	x	{8, {e}}	x
6	x	x	x	{11, {f}}	x	x	x

## Solución del ejemplo 5.38

En la iteración 7, se elige al único vértice disponible, el nodo  $d$ . No hay vértices adyacentes a  $d$  en la lista parcial de nodos. La tabla de iteraciones final es:

	$a$	$b$	$c$	$d$	$e$	$f$	$g$
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	x	{1, {a}}	$\infty$	$\infty$	{2, {a}}	$\infty$	$\infty$
2	x	x	{8, {b}}	$\infty$	{2, {a}}	$\infty$	{6, {b}}
3	x	x	{8, {b}}	$\infty$	x	{8, {e}}	{6, {b, e}}
4	x	x	{8, {b, g}}	$\infty$	x	{8, {e}}	x
5	x	x	x	{16, {c}}	x	{8, {e}}	x
6	x	x	x	{11, {f}}	x	x	x
7	x	x	x	x	x	x	x

(12)

### Solución del ejemplo 5.38

Al observar la columna de  $d$  en la tabla 12, se proviene del vértice  $f$ , en  $f$  se proviene del nodo  $e$  y en la columna de  $e$  se infiere que se proviene del vértice  $a$ , formándose el camino de longitud más corta:

$$a \xrightarrow{2} e \xrightarrow{6} f \xrightarrow{3} d$$

## Solución del ejemplo 5.38

Esta trayectoria es única, no existe otra ruta de longitud mínima. Se aclara al alumno que en el grafo del presente ejemplo, si el nodo de llegada hubiera sido  $g$ , existirían dos caminos de longitud más corta. Ejecutando el algoritmo de *Dijkstra* de  $a$  a  $g$ , se obtendría la tabla 12 hasta la iteración 4, es decir:

	$a$	$b$	$c$	$d$	$e$	$f$	$g$
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	x	{1, {a}}	$\infty$	$\infty$	{2, {a}}	$\infty$	$\infty$
2	x	x	{8, {b}}	$\infty$	{2, {a}}	$\infty$	{6, {b}}
3	x	x	{8, {b}}	$\infty$	x	{8, {e}}	{6, {b, e}}
4	x	x	{8, {b, g}}	$\infty$	x	{8, {e}}	x

### Solución del ejemplo 5.38

Al ubicarnos en la columna de  $g$ , se proviene de  $b$  o de  $e$ . Si se elige a  $b$ , se proviene de  $a$  y si se selecciona a  $e$ , también se proviene de  $a$ , produciéndose dos trayectorias de longitud mínima:

$$\begin{array}{l} a \xrightarrow{1} b \xrightarrow{5} g \\ a \xrightarrow{2} e \xrightarrow{4} g \end{array}$$

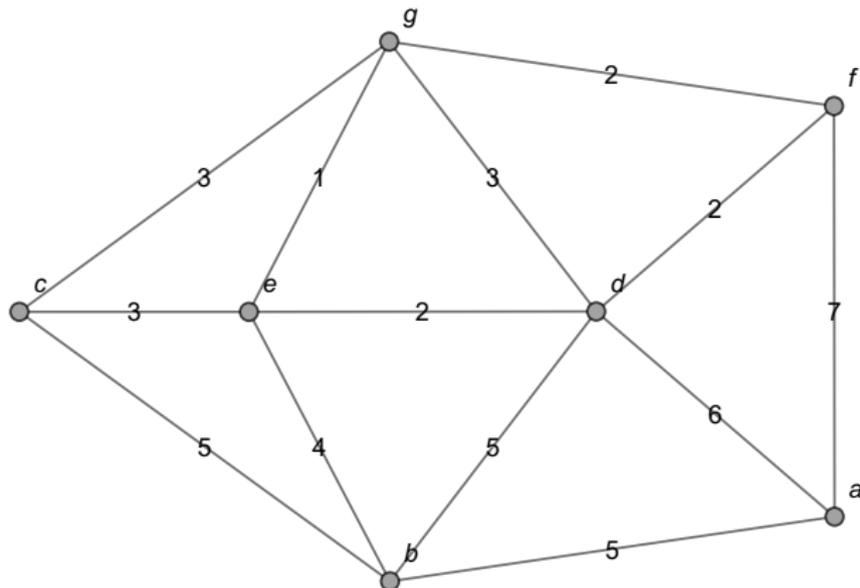


## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
File-132.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-132.zip)

## Example (5.39)

En el grafo dado encuentre una trayectoria de longitud mínima de  $a$  a  $g$ .



## Solución del ejemplo 5.39

En los ejemplos 52, 53 y 54 la tabla de iteraciones correspondiente a la aplicación del algoritmo de *Dijkstra*, se construyó paso a paso por razones didácticas. En la práctica, el estudiante brindará como respuesta en ejercicios de esta naturaleza, directamente la última tabla hallada, sin necesidad de presentar los pasos previos. En esta solución, por lo tanto, se compartirá con el lector, únicamente la tabla final de iteraciones. Al aplicar el algoritmo de *Dijkstra* se obtiene:

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	x	{5, { <i>a</i> }}	$\infty$	{6, { <i>a</i> }}	$\infty$	{7, { <i>a</i> }}	$\infty$
2	x	x	{10, { <i>b</i> }}	{6, { <i>a</i> }}	{9, { <i>b</i> }}	{7, { <i>a</i> }}	$\infty$
3	x	x	{10, { <i>b</i> }}	x	{8, { <i>d</i> }}	{7, { <i>a</i> }}	{9, { <i>d</i> }}
4	x	x	{10, { <i>b</i> }}	x	{8, { <i>d</i> }}	x	{9, { <i>d, f</i> }}
5	x	x	{10, { <i>b</i> }}	x	x	x	{9, { <i>d, e, f</i> }}
6	x	x	{10, { <i>b</i> }}	x	x	x	x

### Solución del ejemplo 5.39

Observando la columna de  $g$ , existen tres trayectorias de peso mínimo 9, al provenir de  $d$ ,  $e$  o  $f$ . Si se escoge el nodo  $d$  se proviene de  $a$ . Si se selecciona el vértice  $e$ , se proviene de  $d$  y en  $d$  se proviene de  $a$ . Si se elige a  $f$ , se proviene de  $a$ . En conclusión, se han derivado tres caminos de longitud más corta:

$$\begin{aligned} a &\xrightarrow{6} d \xrightarrow{3} g \\ a &\xrightarrow{6} d \xrightarrow{2} e \xrightarrow{1} g \\ a &\xrightarrow{7} f \xrightarrow{2} g \end{aligned}$$

## Solución del ejemplo 5.39

Los comandos `AlDijkstra` y `LongitudCaminoMC` del paquete **VilCretas** ofrecen la alternativa de hallar la longitud y una ruta de un camino de extensión mínima entre dos vértices.

La sentencia `AlDijkstra` recibe un grafo conexo y ponderado y dos de sus nodos sobre los cuales se desea ejecutar el algoritmo de *Dijkstra*, retornando la tabla final de iteraciones y una animación con un camino de longitud minimal. En este ejercicio:

`In[ ] :=`

```
grafo = Grafo[{{a, b}, {a, d}, {a, f}, {b, c}, {b, d},  
{b, e}, {c, e}, {c, g}, {d, e}, {d, f}, {d, g}, {e, g},  
{f, g}}, pesos -> {5, 6, 7, 5, 5, 4, 3, 3, 2, 2, 3, 1, 2},  
mostrarpesos -> True];  
AlDijkstra[grafo, a, g]
```

## Solución del ejemplo 5.39

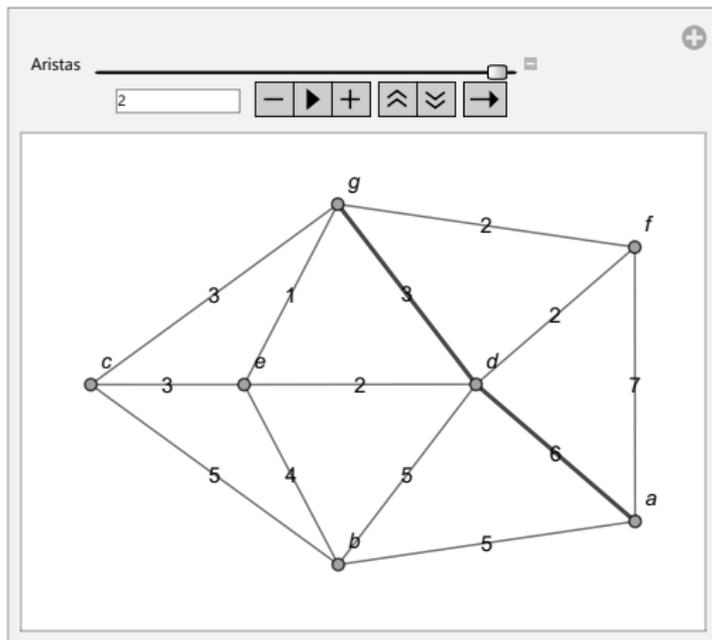
Out[ ] =

La longitud de un camino más corto es: 9

Iteraciones:

$a$	$b$	$d$	$f$	$c$	$e$	$g$
$\{0, \{\}\}$	$\{\infty, \{\}\}$	$\{\infty, \{\}\}$	$\{\infty, \{\}\}$	$\{\infty, \{\}\}$	$\{\infty, \{\}\}$	$\{\infty, \{\}\}$
$\{0, \{\}\}$	$\{5, \{a\}\}$	$\{6, \{a\}\}$	$\{7, \{a\}\}$	$\{\infty, \{\}\}$	$\{\infty, \{\}\}$	$\{\infty, \{\}\}$
$\{0, \{\}\}$	$\{5, \{a\}\}$	$\{6, \{a\}\}$	$\{7, \{a\}\}$	$\{10, \{b\}\}$	$\{9, \{b\}\}$	$\{\infty, \{\}\}$
$\{0, \{\}\}$	$\{5, \{a\}\}$	$\{6, \{a\}\}$	$\{7, \{a\}\}$	$\{10, \{b\}\}$	$\{8, \{d\}\}$	$\{9, \{d\}\}$
$\{0, \{\}\}$	$\{5, \{a\}\}$	$\{6, \{a\}\}$	$\{7, \{a\}\}$	$\{10, \{b\}\}$	$\{8, \{d\}\}$	$\{9, \{d, f\}\}$
$\{0, \{\}\}$	$\{5, \{a\}\}$	$\{6, \{a\}\}$	$\{7, \{a\}\}$	$\{10, \{b\}\}$	$\{8, \{d\}\}$	$\{9, \{d, f, e\}\}$
$\{0, \{\}\}$	$\{5, \{a\}\}$	$\{6, \{a\}\}$	$\{7, \{a\}\}$	$\{10, \{b\}\}$	$\{8, \{d\}\}$	$\{9, \{d, f, e\}\}$
$\{9., \{\{a, d\}, \{d, g\}\}\}$						

## Solución del ejemplo 5.39



## Solución del ejemplo 5.39

La instrucción `AlDijkstra` posee el atributo `all -> True` que otorga la facultad, además, de ver paso a paso los resultados de la aplicación del algoritmo de *Dijkstra*.

Por otra parte, el comando `LongitudCaminoMC` devuelve el peso mínimo y una trayectoria de longitud más corta entre dos vértices, sobre un grafo conexo y ponderado pasado como parámetro. En este ejemplo:

**In[ ] :=**

```
LongitudCaminoMC[grafo, a, g]
```

**Out[ ] =**

```
{9., {{a, d}, {d, g}}}
```

## Solución del ejemplo 5.39

Además, LongitudCaminoMC genera una animación con una ruta de longitud mínima, al agregar la opción ruta  $\rightarrow$  True.

## Nota

Tanto, AlDijkstra como LongitudCaminoMC son comandos que ofrecen excelentes recursos de verificación de resultados.



## Descargue un archivo

<https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/File-133.zip>



## Explicación en video

<https://youtu.be/8OKGR7nK3V8>

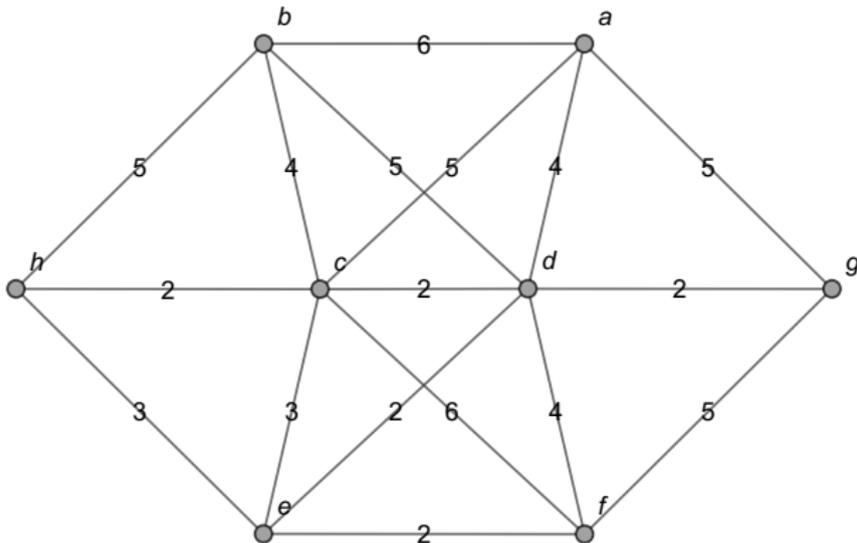


## Explicación en video

<https://youtu.be/mXb7IWkVQJA>

### Example (5.40)

Considere el grafo presentado a continuación. Compruebe con apoyo de software, que el algoritmo de *Dijkstra* no es capaz de encontrar la longitud de un camino más largo del nodo  $h$  al vértice  $g$ .



## Solución del ejemplo 5.40

El algoritmo de *Dijkstra* descrito en el teorema 51, halla la longitud de un camino más corto entre dos vértices de un grafo conexo y ponderado. Bajo esta perspectiva es interesante notar que si los pesos positivos del grafo de interés se pasan a negativos, podría pensarse que al usar el algoritmo de *Dijkstra*, éste retornará la longitud de un camino más largo, en valor absoluto, en lugar del peso de una trayectoria más corta, sin embargo, lo anterior es falso. Supongamos por un momento válida esa forma de razonamiento. Si se desea encontrar la longitud máxima de una ruta de  $h$  a  $g$  en este ejercicio, bastaría entonces con reconstruir el grafo cambiando sus ponderaciones a números negativos y posteriormente, ejecutando sobre él, el algoritmo de *Dijkstra*.

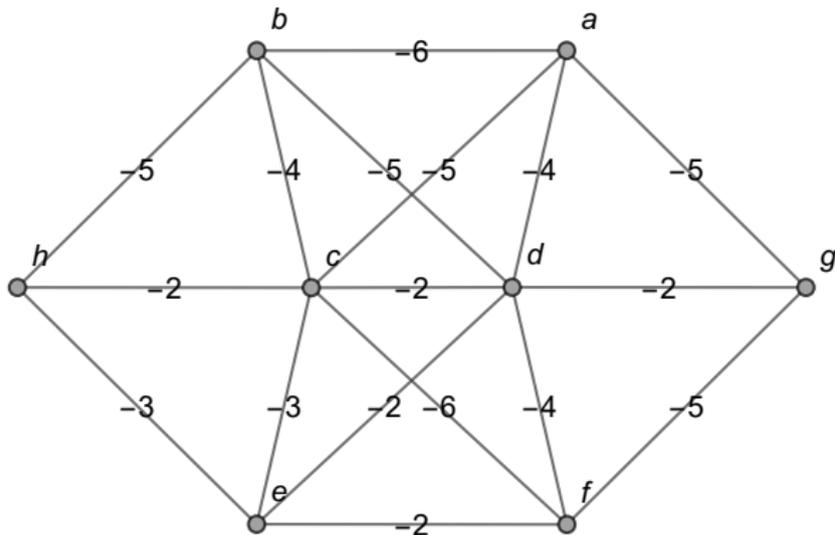
## Solución del ejemplo 5.40

En *Wolfram Mathematica*:**In[ ] :=**

```
grafo = Grafo[{{c, e}, {d, e}, {d, a}, {c, d}, {e, h},  
{c, h}, {c, f}, {a, b}, {a, c}, {b, c}, {b, d}, {b, h},  
{d, f}, {e, f}, {f, g}, {a, g}, {g, d}}, pesos -> -1 {3, 2,  
4, 2, 3, 2, 6, 6, 5, 4, 5, 5, 4, 2, 5, 5, 2},  
mostrarpesos -> True]  
AlDijkstra[grafo, h, g]
```

Solución del ejemplo 5.40

Out[ ] =



## Solución del ejemplo 5.40

La longitud de un camino más corto es: -27

Iteraciones:

$c$	$e$	$d$	$a$	$h$	$f$	$b$	$g$
$\{\infty, \{\}\}$	$\{\infty, \{\}\}$	$\{\infty, \{\}\}$	$\{\infty, \{\}\}$	$\{0, \{\}\}$	$\{\infty, \{\}\}$	$\{\infty, \{\}\}$	$\{\infty, \{\}\}$
$\{-2, \{h\}\}$	$\{-3, \{h\}\}$	$\{\infty, \{\}\}$	$\{\infty, \{\}\}$	$\{0, \{\}\}$	$\{\infty, \{\}\}$	$\{-5, \{h\}\}$	$\{\infty, \{\}\}$
$\{-9, \{b\}\}$	$\{-3, \{h\}\}$	$\{-10, \{b\}\}$	$\{-11, \{b\}\}$	$\{0, \{\}\}$	$\{\infty, \{\}\}$	$\{-5, \{h\}\}$	$\{\infty, \{\}\}$
$\{-16, \{a\}\}$	$\{-3, \{h\}\}$	$\{-15, \{a\}\}$	$\{-11, \{b\}\}$	$\{0, \{\}\}$	$\{\infty, \{\}\}$	$\{-5, \{h\}\}$	$\{-16, \{a\}\}$
$\{-16, \{a\}\}$	$\{-19, \{c\}\}$	$\{-18, \{c\}\}$	$\{-11, \{b\}\}$	$\{0, \{\}\}$	$\{-22, \{c\}\}$	$\{-5, \{h\}\}$	$\{-16, \{a\}\}$
$\{-16, \{a\}\}$	$\{-24, \{f\}\}$	$\{-26, \{f\}\}$	$\{-11, \{b\}\}$	$\{0, \{\}\}$	$\{-22, \{c\}\}$	$\{-5, \{h\}\}$	$\{-27, \{f\}\}$
$\{-16, \{a\}\}$	$\{-24, \{f\}\}$	$\{-29, \{g\}\}$	$\{-11, \{b\}\}$	$\{0, \{\}\}$	$\{-22, \{c\}\}$	$\{-5, \{h\}\}$	$\{-27, \{f\}\}$

## Solución del ejemplo 5.40

La marca del vértice  $g$  es  $-27$ , en valor absoluto:  $|-27| = 27$ . Se concluiría que el peso de una trayectoria de longitud máxima de  $h$  a  $g$  es igual a 27 usando *Dijkstra*. El comando `AlDijkstra` en este ejemplo, no fue capaz de retornar una animación con un camino de longitud más larga, al contener el grafo pesos negativos. Pese a ello, usando la tabla de iteraciones se infiere dicha ruta. En  $g$  se proviene de  $f$ , en  $f$  se proviene de  $c$ , en  $c$  se proviene de  $a$ , en  $a$  se proviene de  $b$  y en  $b$  se proviene de  $h$ . Luego, un supuesto camino más largo de  $h$  a  $g$  es:

$$h \xrightarrow{5} b \xrightarrow{6} a \xrightarrow{5} c \xrightarrow{6} f \xrightarrow{5} g$$

También es importante señalar que si se pasa un grafo con pesos negativos a la sentencia `LongitudCaminoMC`, ésta no devuelve ningún resultado.

## Solución del ejemplo 5.40

Aquí es esencial recalcar el fallo mostrado por *Dijkstra* para hallar la longitud de un camino más largo, dado que en este ejemplo existe otra ruta de  $h$  a  $g$  con un peso acumulado mayor, igual a 28. Este camino corresponde a  $h \xrightarrow{5} b \xrightarrow{6} a \xrightarrow{5} c \xrightarrow{6} f \xrightarrow{4} d \xrightarrow{2} g$ , por lo que el algoritmo de *Dijkstra* no está resolviendo de forma efectiva el problema propuesto. En general, se advierte al estudiante que el algoritmo de *Dijkstra* no provee una solución al problema del cálculo de la longitud de una ruta más larga, inclusive si se aplicara seleccionando en cada iteración el nodo con la mayor marca (lo que resulta equivalente a cambiar a negativo el signo de los pesos del grafo original).



Descargue un archivo

[https://www.escinf.una.ac.cr/discretas/Archivos/Grafos/  
File-134.zip](https://www.escinf.una.ac.cr/discretas/Archivos/Grafos/File-134.zip)

# Propiedades sobre un grafo

Se deja a consideración del alumno un documento con un formato computable que reúne el análisis de distintas propiedades de los grafos, ya estudiadas en el capítulo. El *CDF* recibe el conjuntos de vértices, aristas y pesos de un grafo, devolviendo: los grados de sus nodos, la cantidad de vértices y de lados, si el grafo es conexo, una matriz de adyacencia, una matriz de adyacencia de ponderaciones, una matriz de incidencia, un circuito de *Euler* y de *Hamilton* en caso de existir y la aplicación del algoritmo de *Dijkstra* sobre dos nodos especificados por el usuario. En la figura 10 se muestra el funcionamiento de este documento con un formato computable, sobre un grafo que forma un triángulo.

# Propiedades sobre un grafo



Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/CDFs/  
Propiedades\\_de\\_un\\_grafo.cdf.zip](https://www.esconf.una.ac.cr/discretas/Archivos/CDFs/Propiedades_de_un_grafo.cdf.zip)

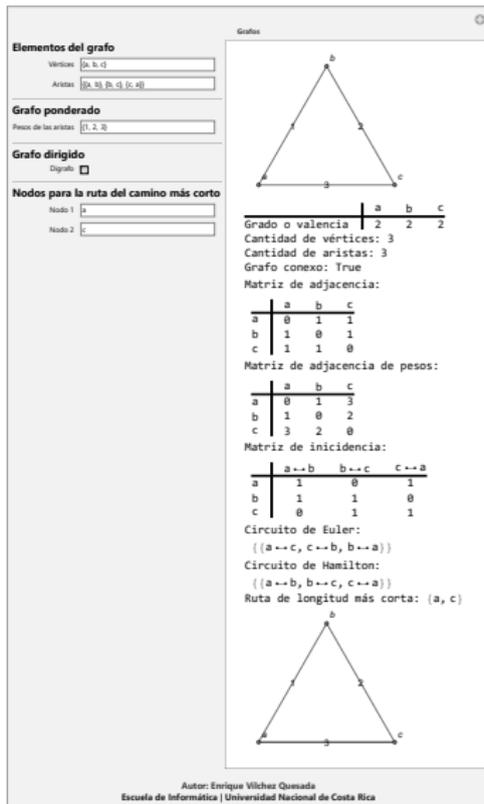


Figura: Funcionamiento del CDF “propiedades sobre un grafo”



## Descargue un archivo

<https://www.esconf.una.ac.cr/discretas/Archivos/Cuadernos/Grafos.pdf.rar>



## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/Quiz\\_grafos.rar](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/Quiz_grafos.rar)



## Abra un sitio web

<https://www.symbaloo.com/mix/vilcretasgrafos>

# ¡Recuerde resolver los ejercicios asignados!



## Descargue un archivo

[https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/  
Exercises.zip](https://www.esconf.una.ac.cr/discretas/Archivos/Grafos/Exercises.zip)

[enrique.vilchez.quesada@una.cr](mailto:enrique.vilchez.quesada@una.cr)

<http://www.esconf.una.ac.cr/discretas>