Análisis y Diseño de Algoritmos

Teoría de Gráficas

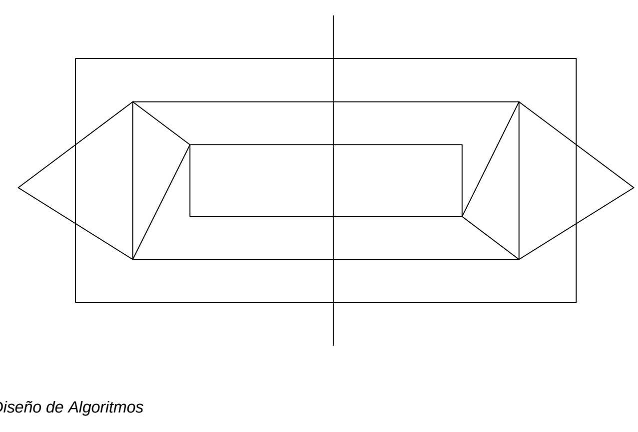
Arturo Díaz Pérez
Sección de Computación
Departamento de Ingeniería Eléctrica
CINVESTAV-IPN
Av. Instituto Politécnico Nacional No. 2508
Col. San Pedro Zacatenco
México, D. F. CP 07300

Tel. (5)747 3800 Ext. 3755
e-mail: `adiaz@cs.cinvestav.mx`

Análisis y Diseño de Algoritmos *Graph*-1

Un Problema

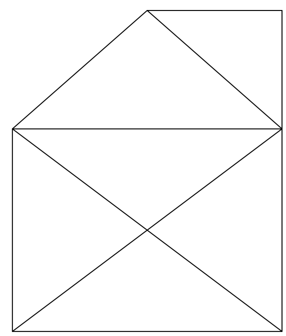
¿Puede dibujar la siguiente figura sin despegar el lápiz y sin remarcar (pasar más de una vez por una línea ?



Análisis y Diseño de Algoritmos *Graph*-2

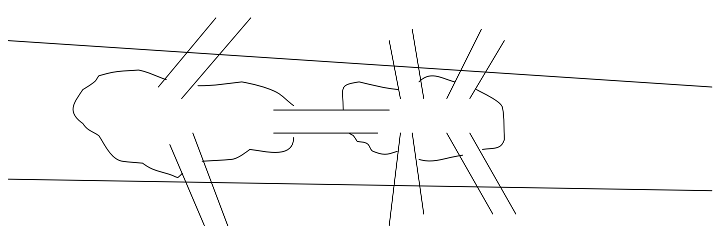
Otro Problema

¿Qué tal con esta?



Análisis y Diseño de Algoritmos *Graph*-3

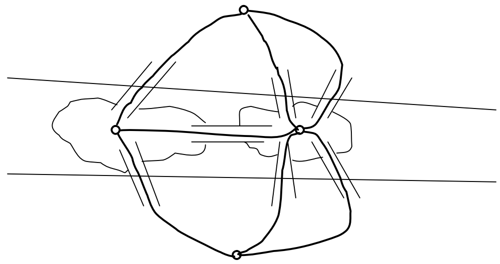
Nuevo Problema



- Los residentes de Köniesberg, Alemania, se preguntaban si era posible hacer un recorrido de su villa cruzando cada uno de los puentes sobre el río Presel solo una vez

Análisis y Diseño de Algoritmos *Graph*-4

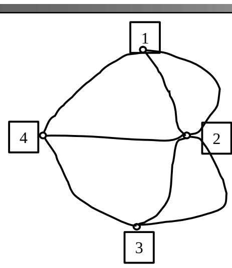
Otro Problema

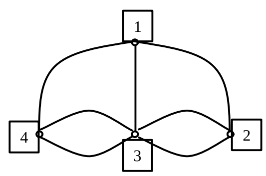


- ☞ Lenhardt Euler (1736). Padre de la teoría de gráficas
  - ← Se imaginó solo lo que es esencial para el problema
  - ← ¿Es posible inicial en algún nodo, hacer un recorrido pasando por cada puente una sola vez y terminar en el nodo inicial ?

Análisis y Diseño de Algoritmos *Graph*-5

Otro Problema



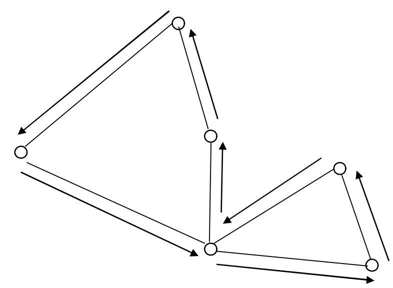


- ☞Lenhardt Euler (1736). Padre de la teoría de gráficas
  - ← Se puede redibujar la gráfica siempre y cuando para cada arco entre un par de vértices *i,j* se coloca un arco en la gráfica redibujada sobre los vértices correspondientes
  - ← Euler demostró que en la gráfica de arriba no se puede encontrar el recorrido buscado. ¿Puede ver por qué?

Análisis y Diseño de Algoritmos *Graph*-6

Recorridos de Euler

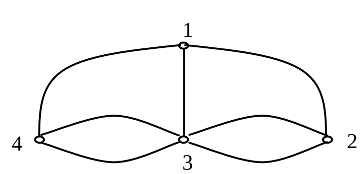
- Un recorrido en una gráfica es llamado un ***recorrido de Euler*** si inicia y termina en el mismo lugar y utiliza cada arco exactamente una sola vez
- Un recorrido en una gráfica es llamado un ***camino de Euler*** si usa cada arco exactamente una sola vez



Análisis y Diseño de Algoritmos *Graph-*7

Recorridos de Euler

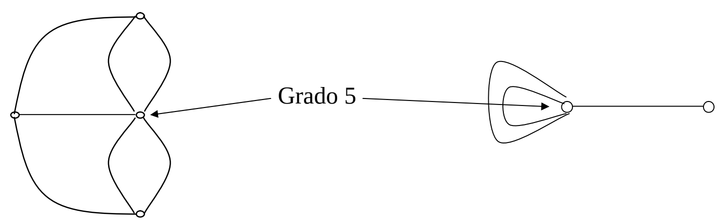
- Si una gráfica tiene un recorrido de Euler se dice que es una ***gráfica Euleriana***
- Para nuestro ejemplo, se tiene que mostrar que la gráfica no es Euleriana



Análisis y Diseño de Algoritmos *Graph-*8

Definiciones

- Una gráfica es ***conectada*** si es posible ir de cualquier vértice a cualquier otro vértice
- Dos nodos (vértices) son ***adyacentes*** si hay al menos un arco entre ellos
- El ***grado*** de un vértice es el número de arcos que salen de él.



Gráfica conectada

Cada ciclo cuenta por 2

Análisis y Diseño de Algoritmos *Graph*-9

Teorema de Euler

- ✶ Una gráfica es Euleriana **si y solo** si es conectada y no tiene vértices de grado impar.
- ✸ Una gráfica tiene un camino de Euler de un nodo a a algún otro nodo b **si y solo** si $a \neq b$ y a y b son los únicos nodos de grado impar.

Análisis y Diseño de Algoritmos *Graph*-10

Prueba del Teorema de Euler

☞ ⇒ ✶, ✹

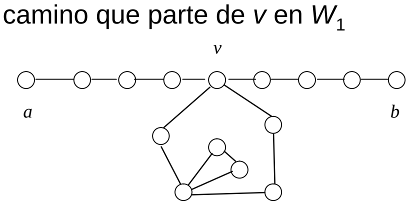
- Suponga que una gráfica *G* Euleriana tiene un camino *C* de un nodo *a* a un nodo *b*, no necesariamente distintos.
- *G* debe ser conectada ya que el camino (recorrido) de Euler alcanza cada nodo de *G*
- Para cada nodo de *G* diferente de *a* y de *b*, *C* usa un arco para llegar y un arco para salir de un nodo. Así el grado de cada nodo, diferente a *a* y *b*, debe ser par.
- Si $a = b$, entonces, tiene también un grado par.
- Si $a \neq b$, entonces, ellos tienen grado impar.

Análisis y Diseño de Algoritmos *Graph*-11

Prueba del Teorema de Euler

☞ ⇐ ✹

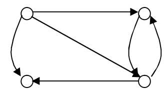
- Suponga que *G* es conectada. Si no hay vértices de grado impar, escoja $a = b$
- Si existen dos vértices de grado impar, llámelos *a* y *b*
- Inicie en *a*, tome un camino W_1 hasta que ya no pueda continuar más; debe estar en *b*. Si no hay un vértice en W_1 que tenga un arco sin visitar entonces hemos terminado.
- Si existe tal vértice, llámelo *v* y camine desde *v* hasta que no pueda continuar más; debe llegar nuevamente a *v.*
- Incorpore el camino que parte de *v* en W_1



Análisis y Diseño de Algoritmos *Graph*-12

Gráficas Dirigidas

- Una **gráfica dirigida** es una gráfica en donde los arcos tienen una dirección particular



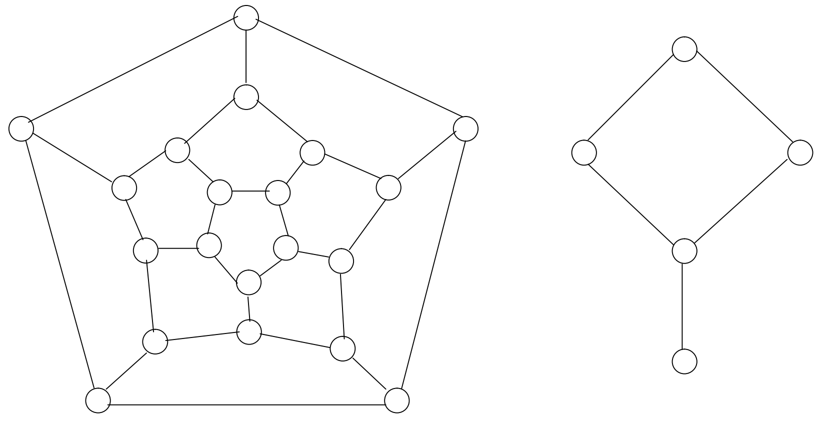
El teorema de Euler es válido para cualquier gráfica dirigida conectada en donde el número de arcos que llegan a un vértice es igual al número de arcos que salen de él

Una gráfica tiene un ***ciclo hamiltoniano*** si existe un recorrido que visita cada vértice exactamente una sola vez (y regresa al vértice inicial). A una gráfica de tal tipo se le llama una ***gráfica hamiltoniana***

Una gráfica tiene un ***camino hamiltoniano*** si existe un recorrido que visita cada vértice exactamente una sola vez

Análisis y Diseño de Algoritmos *Graph*-13

Ciclos y Caminos Hamiltonianos



Análisis y Diseño de Algoritmos *Graph*-14

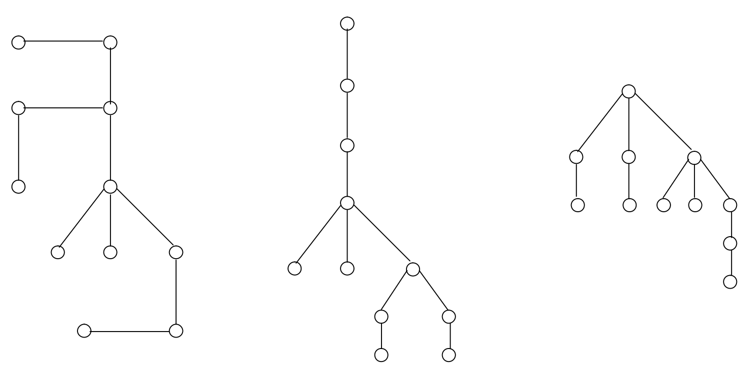
Ciclos y Caminos Hamiltonianos

- Dada una gráfica *G*, ¿existe una forma fácil para determinar si es hamiltoniana?
  - Nadie los sabe; decidir si una gráfica es hamiltoniana es un problema NP-completo
  - Probablemente no existe una caracterización simple de gráficas hamiltonianas como lo hay para las gráficas eulerianas

Análisis y Diseño de Algoritmos — *Graph*-15

Arboles

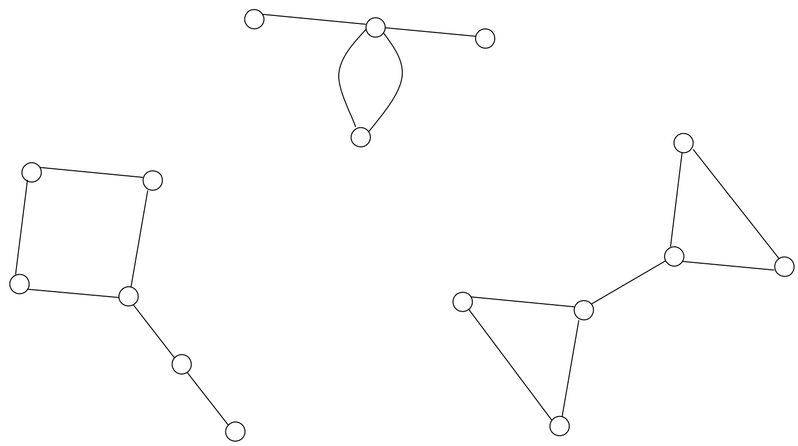
- Un *árbol* es una gráfica conectada sin ciclos



Análisis y Diseño de Algoritmos — *Graph*-16

Arboles

☞ No árboles



Análisis y Diseño de Algoritmos *Graph*-17

Teorema

☞ Sea *G* una gráfica con *n* nodos y *e* arcos

← Las siguientes proposiciones son equivalentes

1. *G* es un árbol
2. Cualesquiera dos vértices de *G* están unidos por un camino único
3. *G* es conectada y $n = e + 1$
4. *G* es acíclica y $n = e + 1$

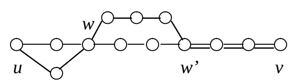
← Prueba

1 Þ 2 Þ 3 Þ 4 Þ 5

Análisis y Diseño de Algoritmos *Graph*-18

$1 \Rightarrow 2$

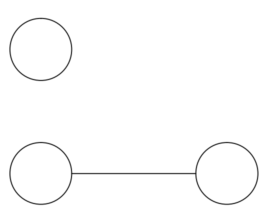
- *G* es un árbol $\Rightarrow \forall\, u \neq v$, *u* y *v* están conectadas por un camino único
  - Prueba por contraposición: not 1 $\Rightarrow$ not 2
  - Sean *u* y *v* fijos. Si ellos no tienen un camino, entonces, *G* es no conectada y de aquí no es un árbol
  - Sean P_1 y P_2 dos caminos diferentes uniendo a *u* y *v*.
  - Sea *w* el primer nodo a lo largo de P_1 cuyo sucesor es diferente a lo largo de P_2.
  - Sea *w'* el primer nodo a lo largo de P_1 después de *w* el cual está también en P_2.
  - El camino *w*-seg[P_1]-*w'*-seg[P_2] forma un ciclo en *G*, donde seg[P_i] es el segmento de P_i que está entre *w* y *w'*



Análisis y Diseño de Algoritmos *Graph*-19

$2 \Rightarrow 3$

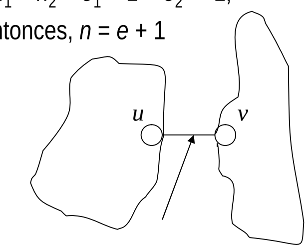
- Cualesquiera dos vértices de *G* están conectados por un camino único, entonces, *G* es conectada y $n = e + 1$
  - Prueba. Claramente *G* es conectada. Mostraremos que $n = e + 1$ por inducción sobre *n*
  - Casos base
    - $n = 1$
    - $n = 2$



Análisis y Diseño de Algoritmos *Graph*-20

2 $\Rightarrow$3

- Paso inductivo
  - Suponga que la proposición es válida para todos las gráficas con menos de *n* vértices. Consideremos una gráfica conectada con *n* vértices. Si se quita un arco de *G*, entonces, se formarán dos componentes conectadas.
  - Dado que cada componente tiene menos de *n* vértices, entonces, el teorema es válido para cada componente.
    - $n_1 = e_1 + 1$, $n_2 = e_2 + 1$, $n = n_1 + n_2 = e_1 + 1 + e_2 + 1$,
    - Dado que $e = e_1 + e_2 + 1$, entonces, $n = e + 1$

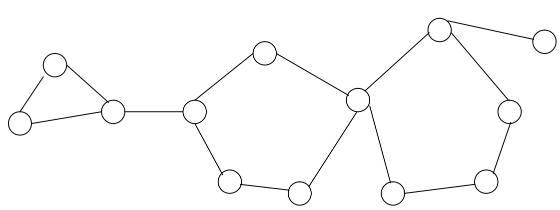


Camino único de *u* a *v*

Análisis y Diseño de Algoritmos Graph-21

3 $\Rightarrow$ 4

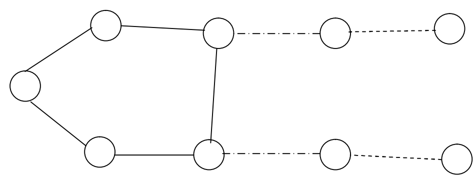
- *G* es conectada y $n = e + 1$, entonces, *G* es acíclica *y* $n = e + 1$.
  - Prueba. Supongamos que *G* tiene un ciclo de longitud *r*. El ciclo tiene *r* vértices y *r* arcos.



Análisis y Diseño de Algoritmos Graph-22

$3 \Rightarrow 4$

- Existen *n-r* vértices que no están en el ciclo. Para cada vértice que no está en el ciclo, asociamos uno de sus arcos lo que nos acerca al ciclo. Cada vértice, no en el ciclo, está asociado con un arco único, no en el ciclo.
- Así, con *n-r+r* vértices y *n-r+r* arcos creamos una gráfica conectada por lo que, se contradice el hecho de $n=e+1$ (hay un arco menos que nodos).



Análisis y Diseño de Algoritmos *Graph*-23

$4 \Rightarrow 1$

- *G* es acíclica *y n = e + 1*, entonces, *G* es acíclica y conectada
  - Prueba. Si *G* es acíclica y no conectada entonces tiene $k > 1$ componentes conectadas
    - $G_1, G_2, \ldots, G_k$,
  - Cada una de ellas es acíclica y conectada y, por lo tanto, forman un árbol
  - Ya se sabe, $1 \Rightarrow 4$, entonces,
    - $n_1 = e_1 + 1$
    - $n_2 = e_2 + 1$
    - ...
    - $n_k = e_k + 1$

$$\sum_{i=1}^{k} n_i = n = \sum_{i=1}^{k} e_i + 1 = e + k \neq e + 1$$

Lo que implica una contradicción con la hipótesis

Análisis y Diseño de Algoritmos *Graph*-24