Algebra Matricial y Teoría de Grafos

Unidad 3: Nociones de teoría de grafos

Luis M. Torres

Escuela Politécnica del Litoral

Quito, Enero 2008

Contenido

- Motivación
- Conceptos básicos
- Representaciones computacionales
- Problema del sendero euleriano
- Problema del árbol generador de peso mínimo
- Problemas de caminos más cortos

Contenido

- Motivación
- Conceptos básicos
- Representaciones computacionales
- Problema del sendero euleriano
- Problema del árbol generador de peso mínimo
- Problemas de caminos más cortos

Todo empezó en Königsberg...

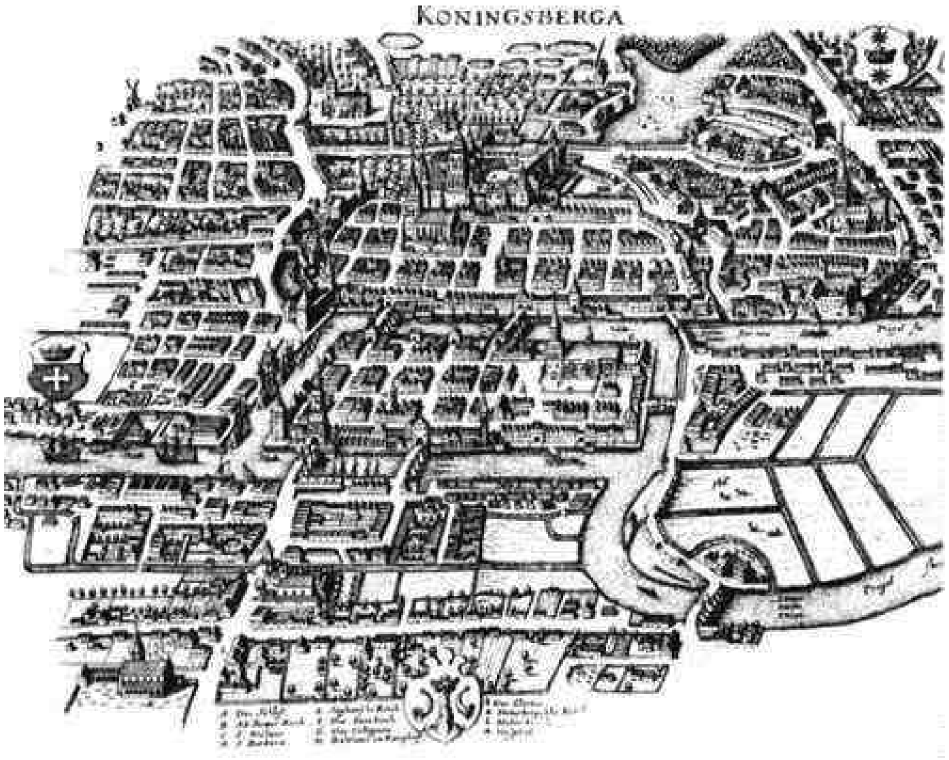
Kaliningrado (2005)
exclave ruso en el mar báltico...

Todo empezó en Königsberg...

Kaliningrado (2005)
exclave ruso en el mar báltico...

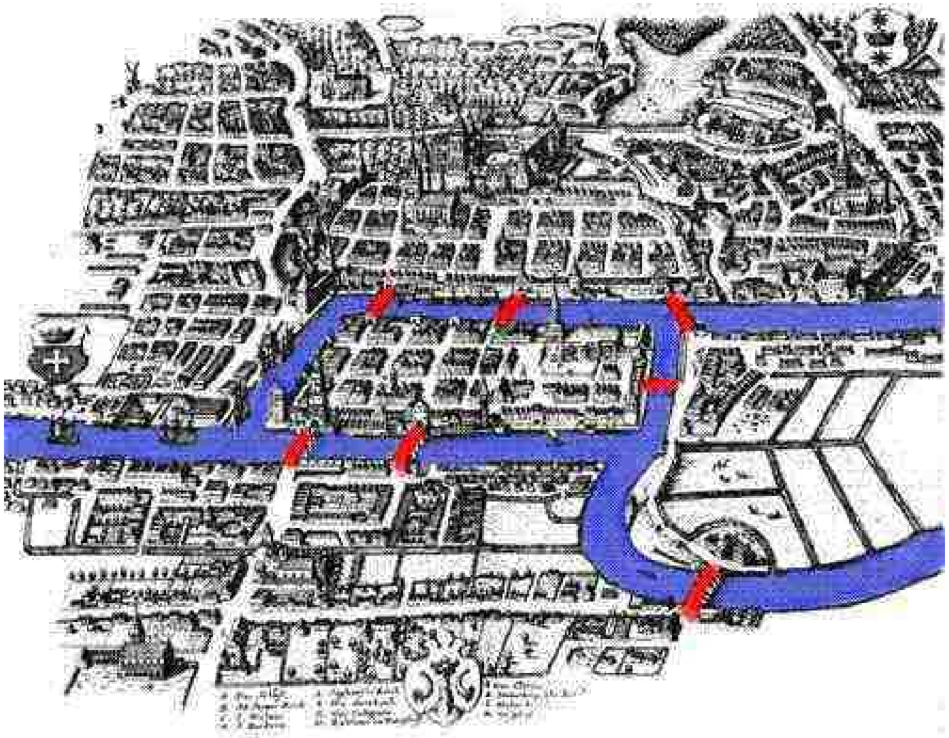
Königsberg ($\sim$ 1750)
... puerto del antiguo imperio prusiano

...con un problema de puentes



El Pregel atraviesa la ciudad formando dos islas

...con un problema de puentes



El Pregel atraviesa la ciudad formando dos islas

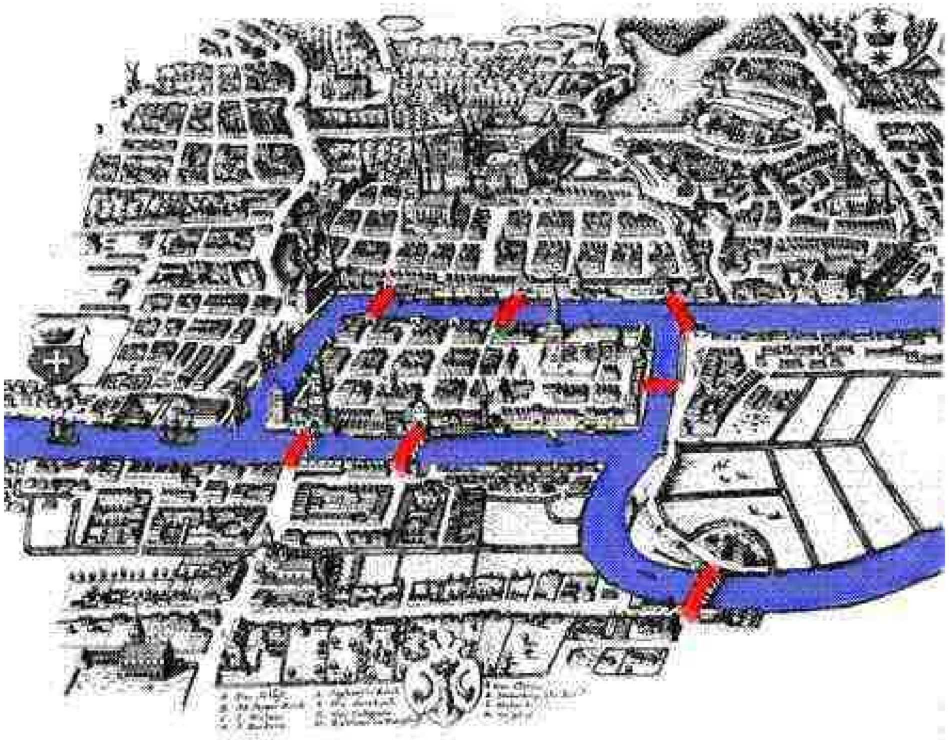
Problema:
Cruzar cada uno de los siete puentes exactamente una vez

Leonhard Euler

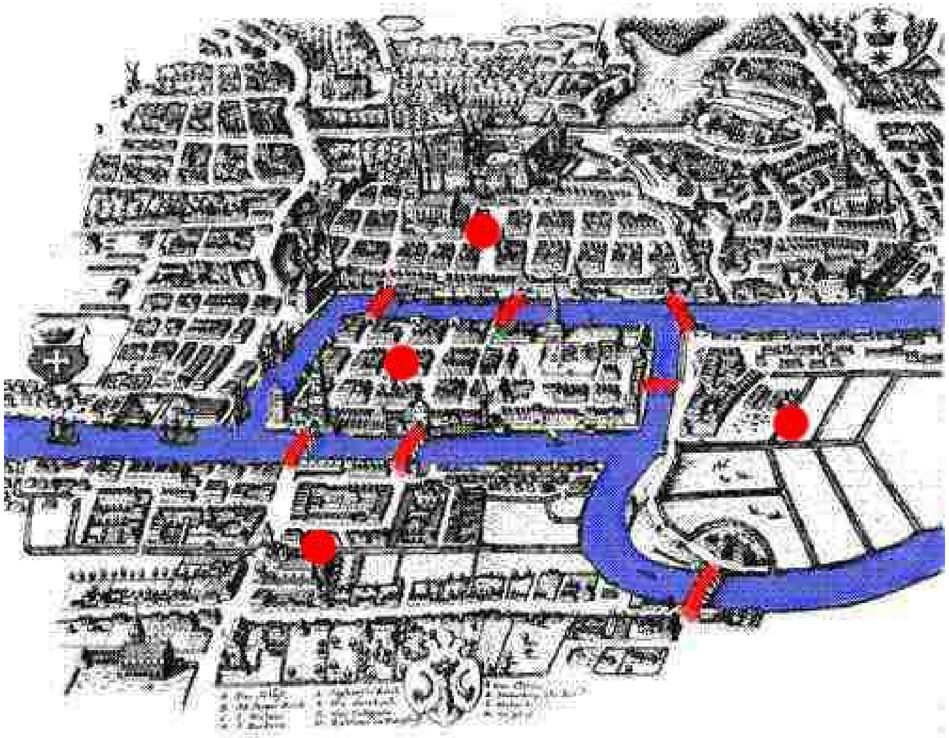
1707, Bâle
1783, St. Petersburgo

- trabajos en análisis
- consolidación de las matemáticas puras
- (1736) teoría de grafos

Volviendo a los puentes...

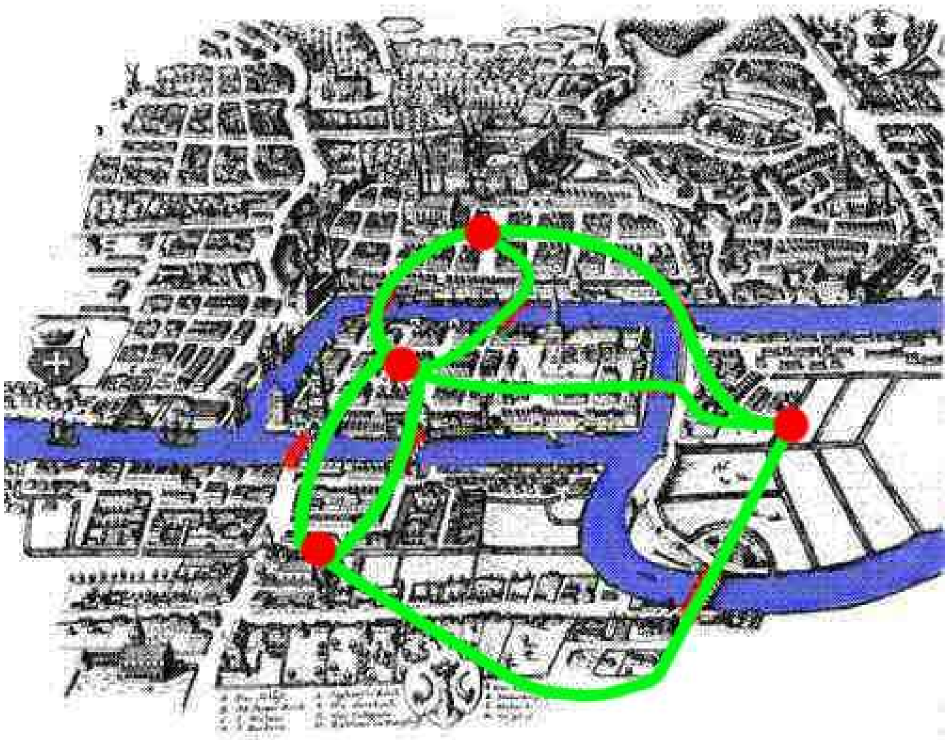


Volviendo a los puentes...



Cuatro orillas...

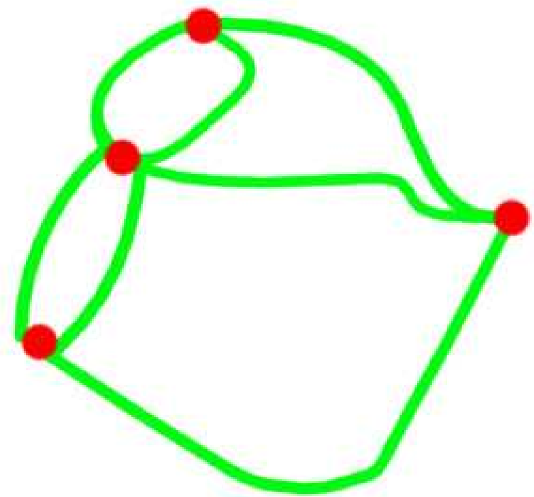
Volviendo a los puentes...



Cuatro orillas...

... unidas por siete puentes ...

Volviendo a los puentes...



Cuatro orillas...

... unidas por siete puentes ...

Idea:
Olvidar la ciudad, retener la estructura!

Matemáticas discretas

Problemas:

- Caminos eulerianos
- Caminos más cortos
- Arboles generadores
- Flujos máximos
- Programación lineal y entera
- ...

Matemáticas discretas

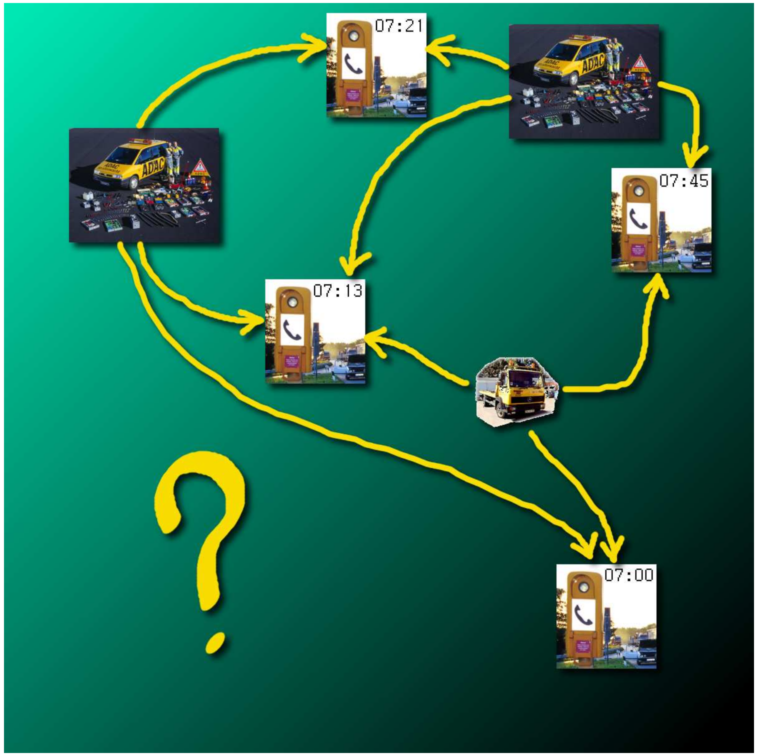
Problemas:

- Caminos eulerianos
- Caminos más cortos
- Arboles generadores
- Flujos máximos
- Programación lineal y entera
- ...

Aplicaciones:

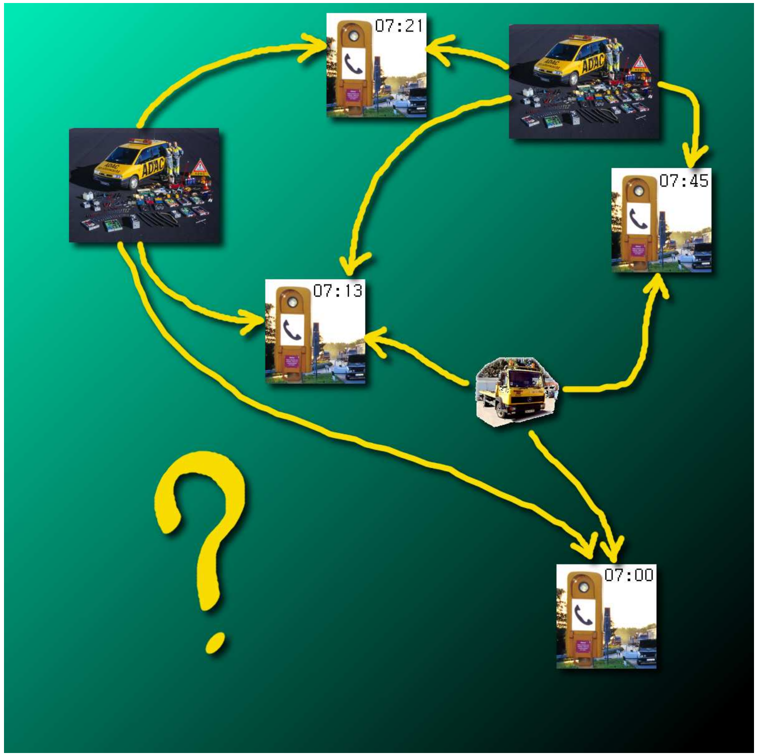
- Logística
- Transporte
- Telecomunicaciones
- Redes de distribución
- Secuenciamiento genético
- ...

El problema del ADAC



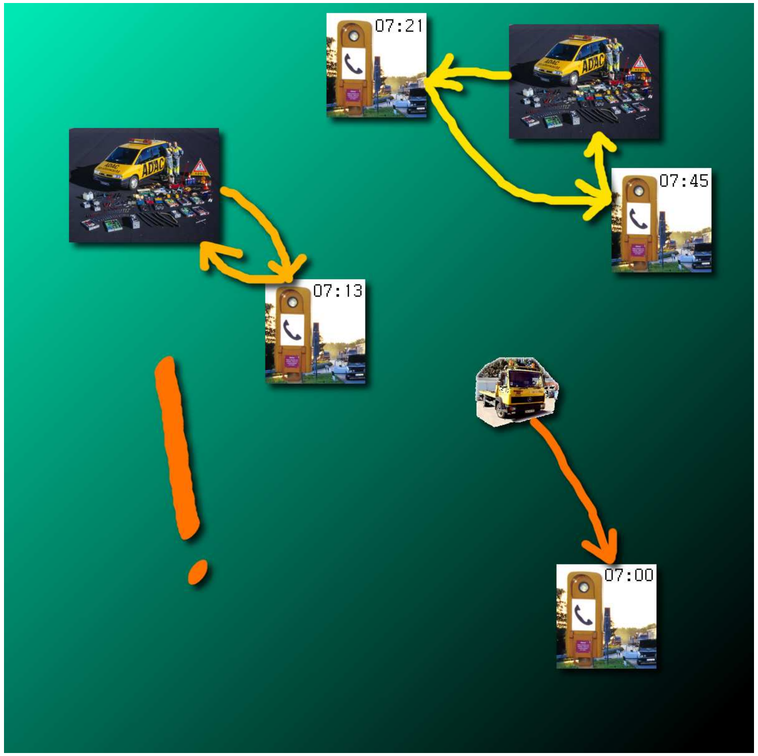
pedidos, unidades, contratistas...

El problema del ADAC



Central de despacho

El problema del ADAC



Plan de rutas

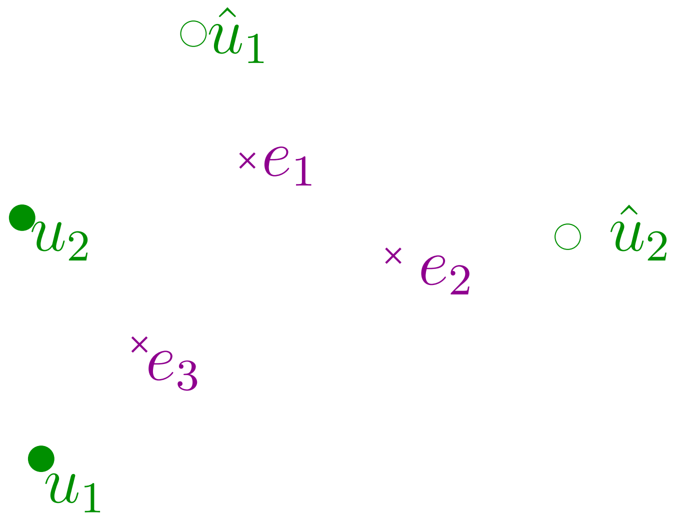
El problema del ADAC

Objetivo:
Diseñar un algoritmo de optimización para el enrutamiento de las unidades

Plan de rutas

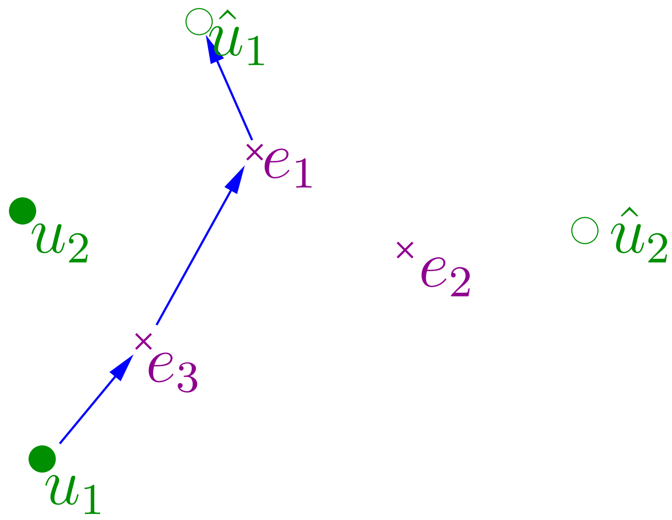
Formulando el IP

> **Idea:**
> Definir una variable binaria por cada **ruta** de servicio.



Formulando el IP

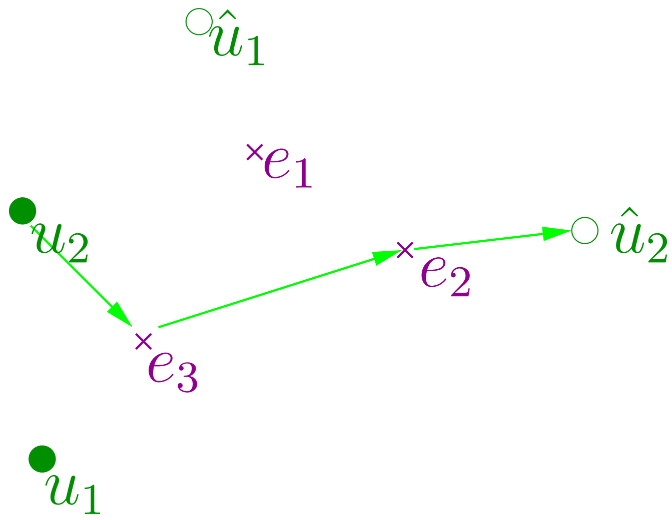
Idea:
Definir una variable binaria por cada **ruta** de servicio.



	T_1	
	1	r_1
	0	r_2
	1	r_3
	1	u_1
	0	u_2
	c_{T_1}	
	x_{T_1}	

Formulando el IP

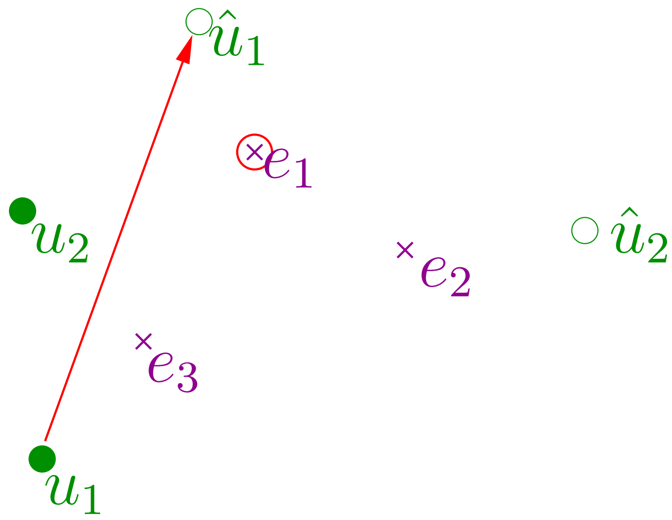
Idea:
Definir una variable binaria por cada **ruta** de servicio.



	T_1	T_2	
	1	0	r_1
	0	1	r_2
	1	1	r_3
	1	0	u_1
	0	1	u_2
	c_{T_1}	c_{T_2}	
	x_{T_1}	x_{T_2}	

Formulando el IP

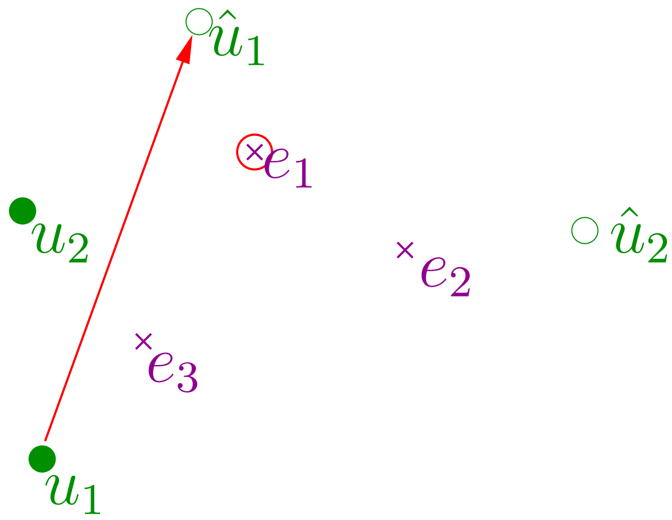
Idea:
Definir una variable binaria por cada **ruta** de servicio.



	T_1	T_2	T_3	T_4	
	1	0	0	1	r_1
	0	1	0	0	r_2
	1	1	0	0	r_3
	1	0	1	0	u_1
	0	1	0	0	u_2
	c_{T_1}	c_{T_2}	c_{T_3}	c_{T_4}	
	x_{T_1}	x_{T_2}	x_{T_3}	x_{T_4}	

Formulando el IP

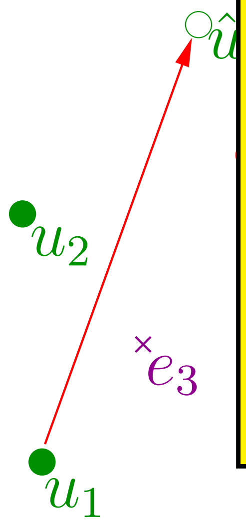
Idea:
Definir una variable binaria por cada **ruta** de servicio.



	T_1	T_2	T_3	T_4	$\ldots$	T_N	
	1	0	0	1	$\ldots$		r_1
	0	1	0	0	$\ldots$		r_2
A	1	1	0	0	$\ldots$		r_3
	1	0	1	0	$\ldots$		u_1
	0	1	0	0	$\ldots$		u_2
c^T	c_{T_1}	c_{T_2}	c_{T_3}	c_{T_4}	$\ldots$	c_{T_N}	
x^T	x_{T_1}	x_{T_2}	x_{T_3}	x_{T_4}	$\ldots$	x_{T_N}	

Formulando el IP

Idea:
Definir una variable binaria por cada **ruta** de servicio.



Formular un **enorme** Problema de Particionamiento (IP):

$$\min c^T x$$

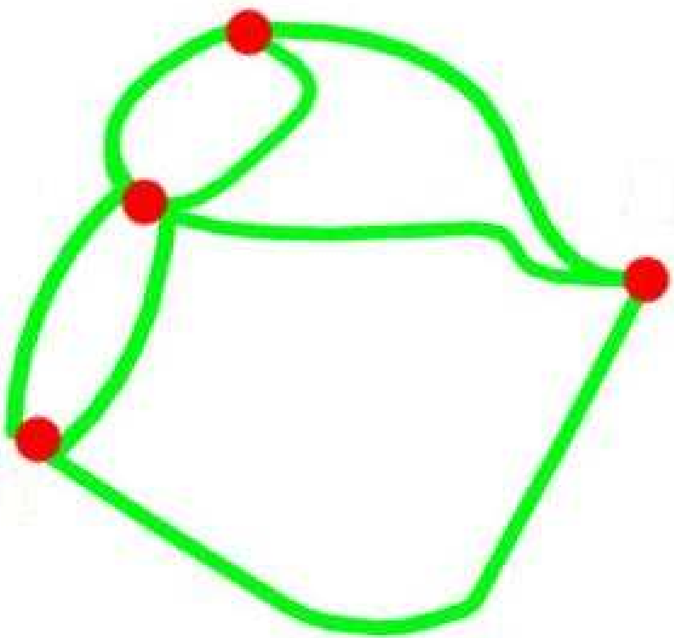
s.t.

$$Ax = 1$$

$$x \in \{0,1\}^N$$

							r_1
							r_2
							r_3
							u_1
							u_2
c^T	c_{T_1}	c_{T_2}	c_{T_3}	c_{T_4}	$\dots$	c_{T_N}	
x^T	x_{T_1}	x_{T_2}	x_{T_3}	x_{T_4}	$\dots$	x_{T_N}	

Y los puentes?



Contenido

- Motivación
- Conceptos básicos
- Representaciones computacionales
- Problema del sendero euleriano
- Problema del árbol generador de peso mínimo
- Problemas de caminos más cortos

Conceptos básicos

Definición:

Un grafo es un par ordenado $G = (V, E)$ donde:

- V es un conjunto **finito**
- E es un **multiconjunto** de la forma $E \subseteq \{\{i, j\} \; : \; i \in V, \; j \in V\}$

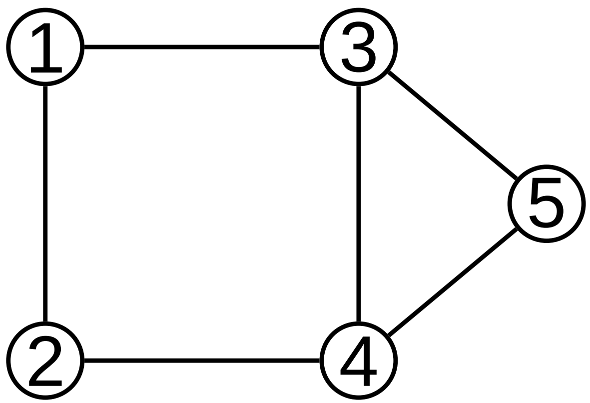
Conceptos básicos

Definición:

Un grafo es un par ordenado $G = (V, E)$ donde:

- V es un conjunto **finito**
- E es un **multiconjunto** de la forma
  $E \subseteq \{\{i, j\} \, : \, i \in V, \, j \in V\}$

Ejemplo:



$$V = \{1, 2, 3, 4, 5\}$$

$$E = \{\{1, 2\}, \{1, 3\}, \{2, 4\}, \{3, 4\}, \{3, 5\}, \{4, 5\}\}$$

Conceptos básicos

Definición:

Un grafo es un par ordenado $G = (V, E)$ donde:

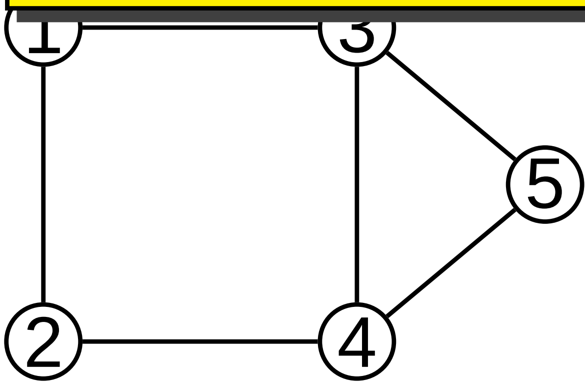
- V
- E

  E

Ejemp

Los elementos de V se llaman **nodos** de G, los elementos de E son las **aristas** de G.
Designaremos en adelante:

$$n := |V| \qquad m := |E|$$



$$V = \{1, 2, 3, 4, 5\}$$

$$E = \{\{1,2\}, \{1,3\}, \{2,4\}, \{3,4\}, \{3,5\}, \{4,5\}\}$$

Conceptos básicos

- una arista $e \in E$ es **incidente** a un nodo $i \in V$ si $i \in e$

Conceptos básicos

- una arista $e \in E$ es **incidente** a un nodo $i \in V$ si $i \in e$
- dos nodos $i, j \in V$ son **adyacentes** si $\{i, j\} \in E$

Conceptos básicos

- una arista $e \in E$ es **incidente** a un nodo $i \in V$ si $i \in e$
- dos nodos $i, j \in V$ son **adyacentes** si $\{i, j\} \in E$
- el conjunto de nodos adyacentes a un cierto nodo $i \in V$ es la **vecindad por nodos** de i:

$$N_V(i) := \{j \in V \ : \ ij \in E\}$$

Conceptos básicos

- una arista $e \in E$ es **incidente** a un nodo $i \in V$ si $i \in e$
- dos nodos $i, j \in V$ son **adyacentes** si $\{i, j\} \in E$
- el conjunto de nodos adyacentes a un cierto nodo $i \in V$ es la **vecindad por nodos** de i:

$$N_V(i) := \{j \in V \ : \ ij \in E\}$$

- el conjunto de aristas incidentes a un cierto nodo $i \in V$ es la **vecindad por aristas** de i:

$$N_E(i) := \{ij \in E\}$$

Conceptos básicos

- el **grado por nodos** de un nodo i es la cardinalidad de su vecindad por nodos ($\leadsto$ cantidad de nodos adyacentes):

$$d_V(i) := |N_V(i)|$$

- el **grado por aristas** de un nodo i es la cardinalidad de su vecindad por aristas ($\leadsto$ cantidad de aristas incidentes):

$$d_E(i) := |N_E(i)|$$

Usaremos en adelante el término grado para referirnos al grado por aristas:

$$d(i) := d_E(i)$$

Conceptos básicos

Cadenas y senderos

- Una cadena P es una sucesión de aristas donde dos aristas consecutivas tienen siempre un nodo en común $\leadsto$ trayectoria sobre el grafo

$$P := \langle \{i_0, i_1\}, \{i_1, i_2\}, \ldots, \{i_{k-1}, i_k\} \rangle$$

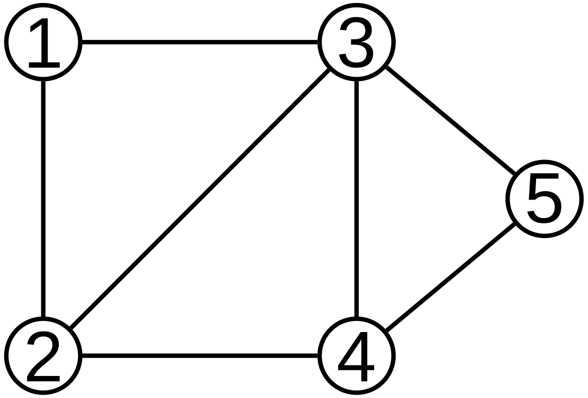
- Una cadena que no repite aristas se llama sendero

- Un sendero es elemental si no repite nodos

- Un sendero es cerrado si termina donde empezó ($i_0 = i_k$)

Conceptos básicos

Ejemplo:

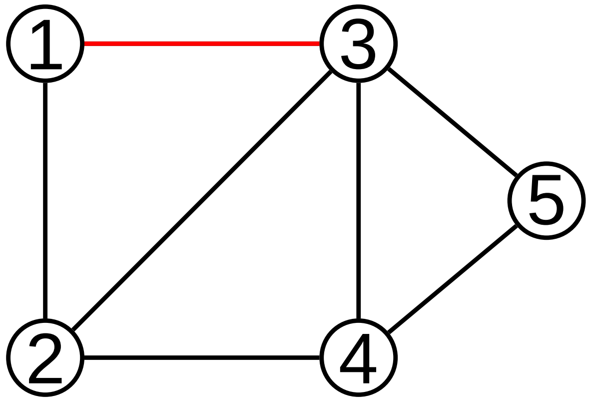


Una cadena en G:

$$P := \langle \rangle$$

Conceptos básicos

Ejemplo:

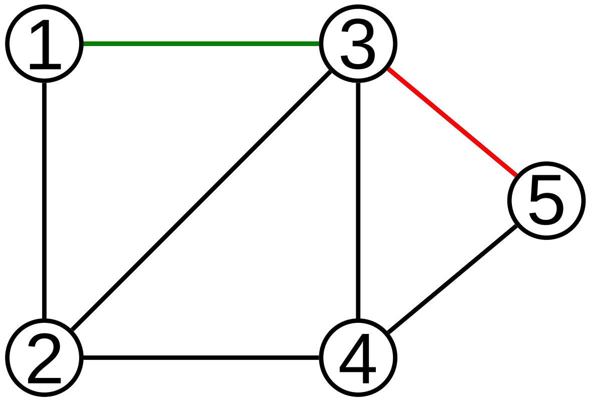


Una cadena en G:

$$P := \langle \{1, 3\}, \rangle$$

Conceptos básicos

Ejemplo:

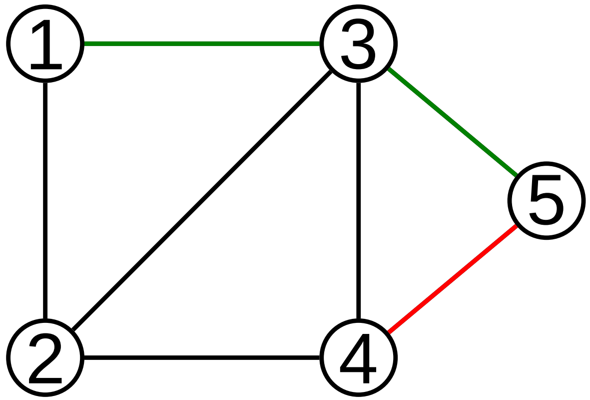


Una cadena en G:

$$P := \langle \{1, 3\}, \{3, 5\}, \rangle$$

Conceptos básicos

Ejemplo:

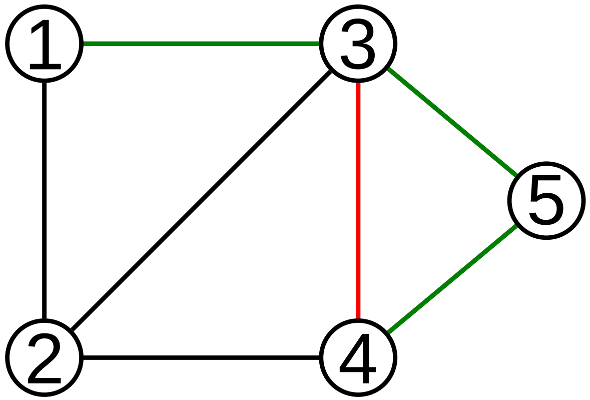


Una cadena en G:

$$P := \langle \{1,3\}, \{3,5\}, \{4,5\}, \rangle$$

Conceptos básicos

Ejemplo:

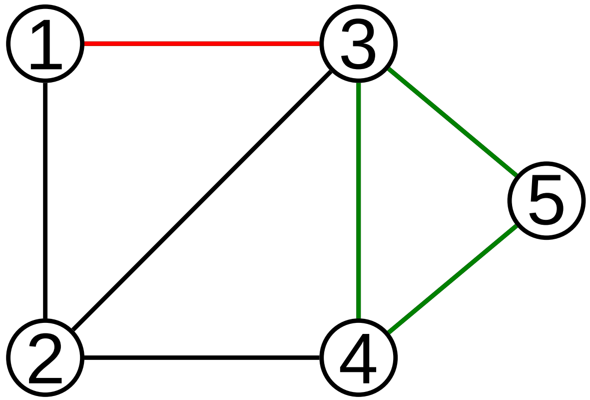


Una cadena en G:

$$P := \langle \{1,3\}, \{3,5\}, \{4,5\}, \{3,4\}, \rangle$$

Conceptos básicos

Ejemplo:

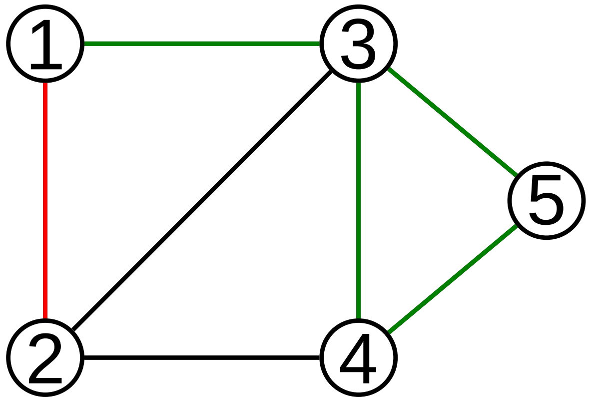


Una cadena en G:

$$P := \langle \{1,3\}, \{3,5\}, \{4,5\}, \{3,4\}, \{1,3\}, \rangle$$

Conceptos básicos

Ejemplo:

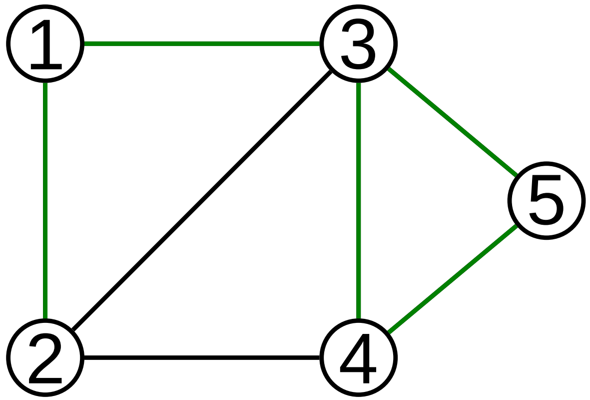


Una cadena en G:

$$P := \langle \{1,3\}, \{3,5\}, \{4,5\}, \{3,4\}, \{1,3\}, \{1,2\} \rangle$$

Conceptos básicos

Ejemplo:

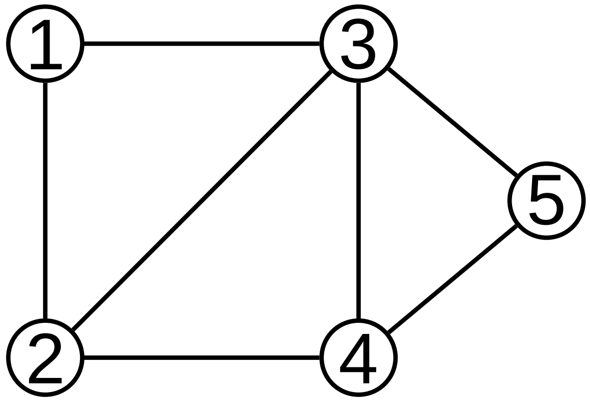


Una cadena en G:

$$P := \langle \{1,3\}, \{3,5\}, \{4,5\}, \{3,4\}, \{1,3\}, \{1,2\} \rangle$$

Conceptos básicos

Ejemplo:

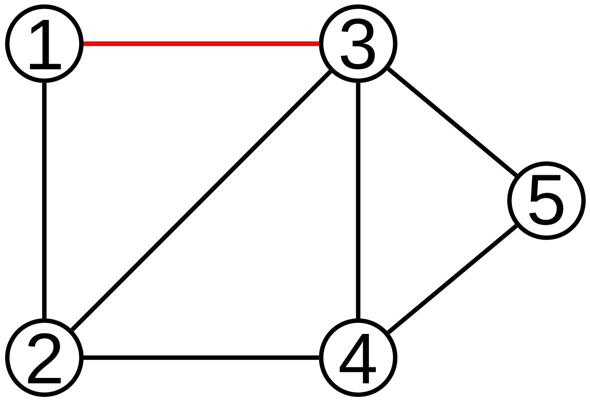


Un sendero en G:

$$P := \langle\rangle$$

Conceptos básicos

Ejemplo:

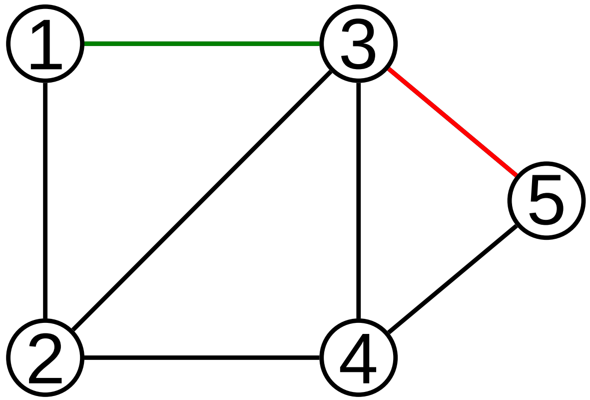


Un sendero en G:

$$P := \langle \{1, 3\}, \rangle$$

Conceptos básicos

Ejemplo:

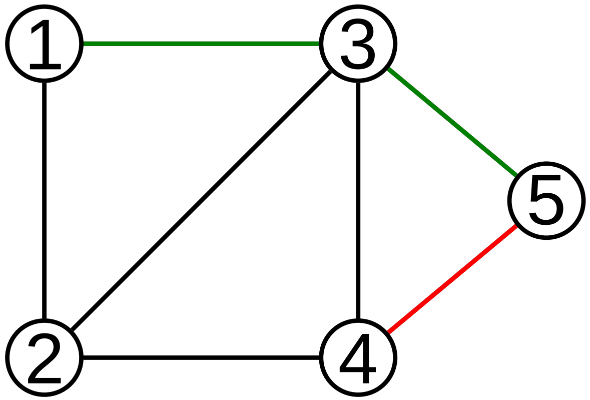


Un sendero en G:

$$P := \langle \{1,3\}, \{3,5\}, \rangle$$

Conceptos básicos

Ejemplo:

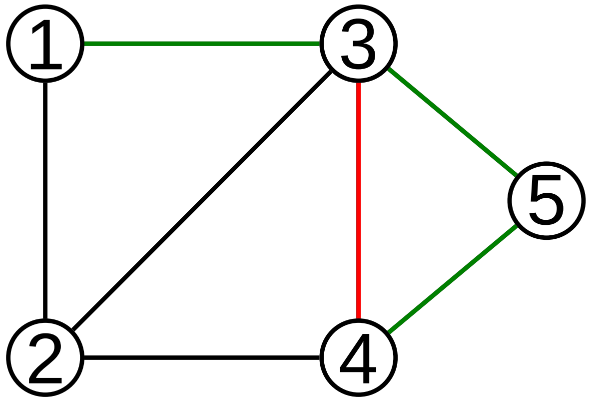


Un sendero en G:

$$P := \langle \{1,3\}, \{3,5\}, \{4,5\}, \rangle$$

Conceptos básicos

Ejemplo:

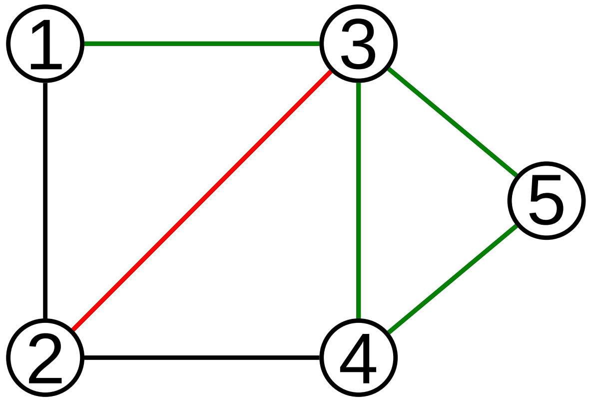


Un sendero en G:

$$P := \langle \{1,3\}, \{3,5\}, \{4,5\}, \{3,4\}, \rangle$$

Conceptos básicos

Ejemplo:

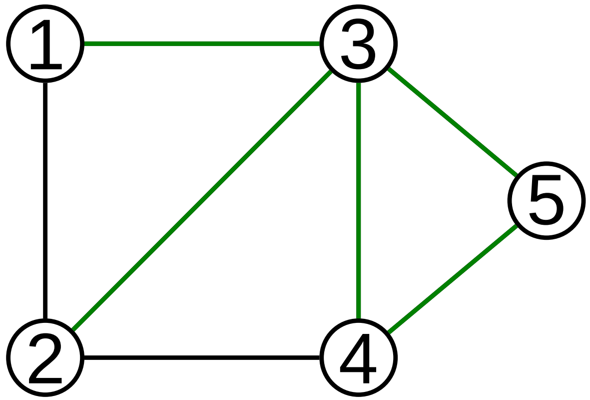


Un sendero en G:

$$P := \langle \{1,3\}, \{3,5\}, \{4,5\}, \{3,4\}, \{2,3\} \rangle$$

Conceptos básicos

Ejemplo:

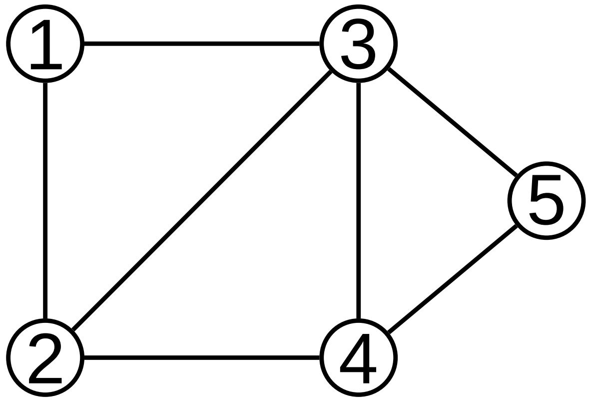


Un sendero en G:

$$P := \langle \{1,3\}, \{3,5\}, \{4,5\}, \{3,4\}, \{2,3\} \rangle$$

Conceptos básicos

Ejemplo:

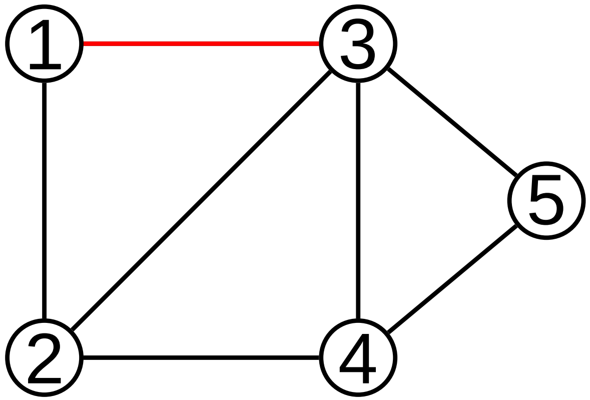


Un sendero elemental en G:

$$P := \langle \rangle$$

Conceptos básicos

Ejemplo:

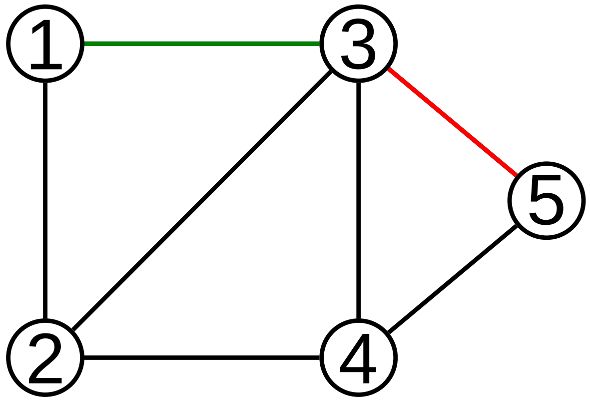


Un sendero elemental en G:

$$P := \langle \{1, 3\}, \rangle$$

Conceptos básicos

Ejemplo:

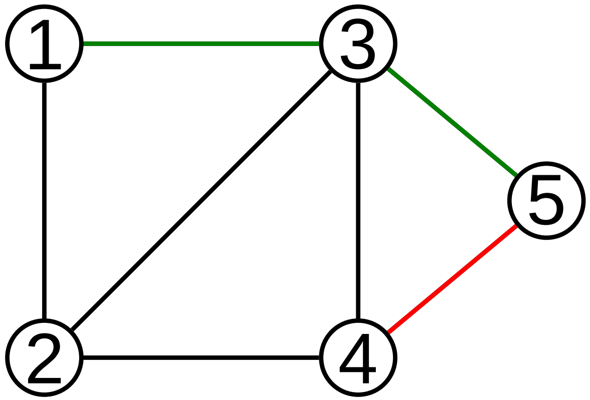


Un sendero elemental en G:

$$P := \langle \{1,3\}, \{3,5\}, \rangle$$

Conceptos básicos

Ejemplo:

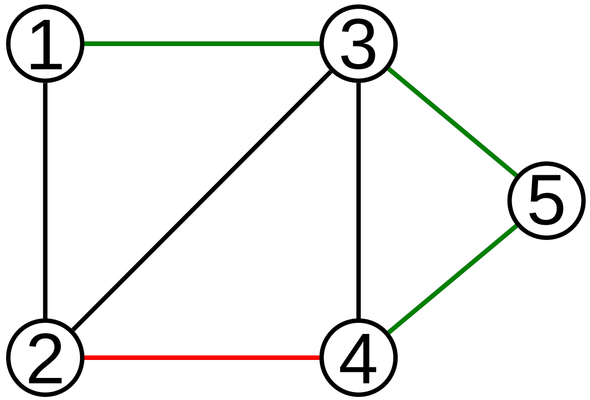


Un sendero elemental en G:

$$P := \langle \{1,3\}, \{3,5\}, \{4,5\}, \rangle$$

Conceptos básicos

Ejemplo:

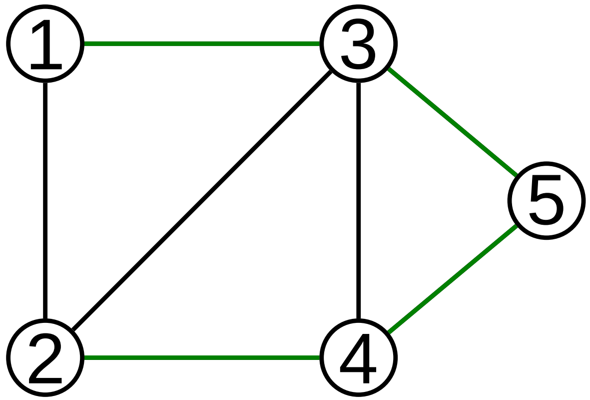


Un sendero elemental en G:

$$P := \langle \{1,3\}, \{3,5\}, \{4,5\}, \{2,4\} \rangle$$

Conceptos básicos

Ejemplo:

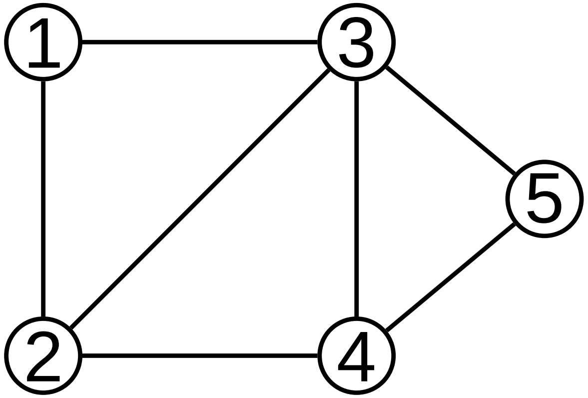


Un sendero elemental en G:

$$P := \langle \{1,3\}, \{3,5\}, \{4,5\}, \{2,4\} \rangle$$

Conceptos básicos

Ejemplo:

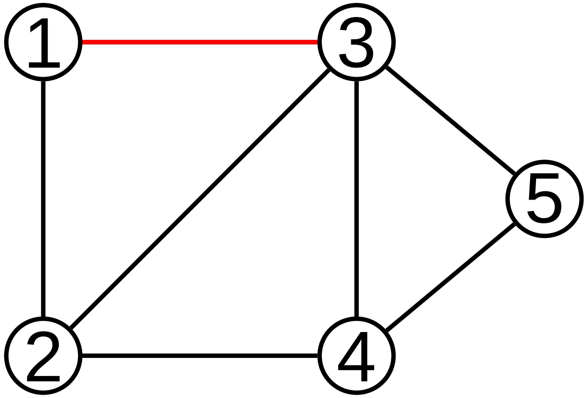


Un sendero cerrado en G:

$$P := \langle \rangle$$

Conceptos básicos

Ejemplo:

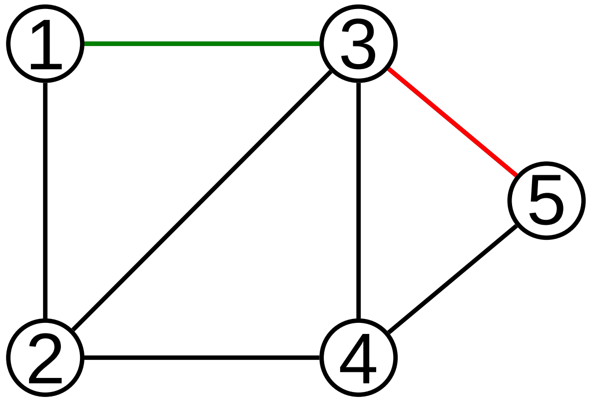


Un sendero cerrado en G:

$$P := \langle \{1, 3\}, \rangle$$

Conceptos básicos

Ejemplo:

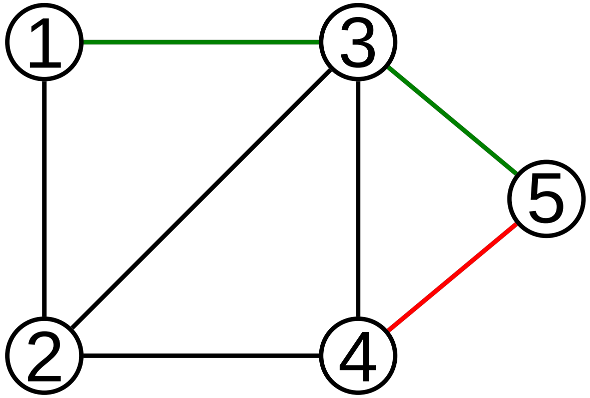


Un sendero cerrado en G:

$$P := \langle \{1,3\}, \{3,5\}, \rangle$$

Conceptos básicos

Ejemplo:

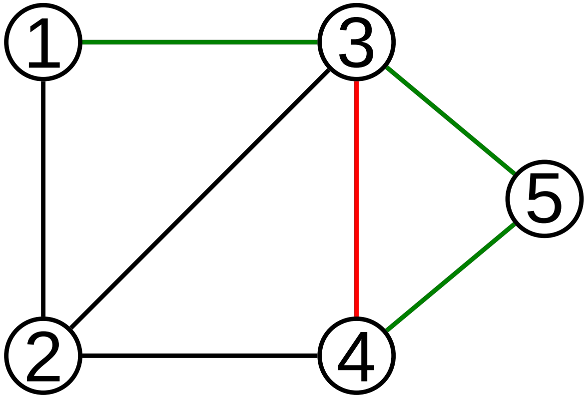


Un sendero cerrado en G:

$$P := \langle \{1, 3\}, \{3, 5\}, \{4, 5\}, \rangle$$

Conceptos básicos

Ejemplo:

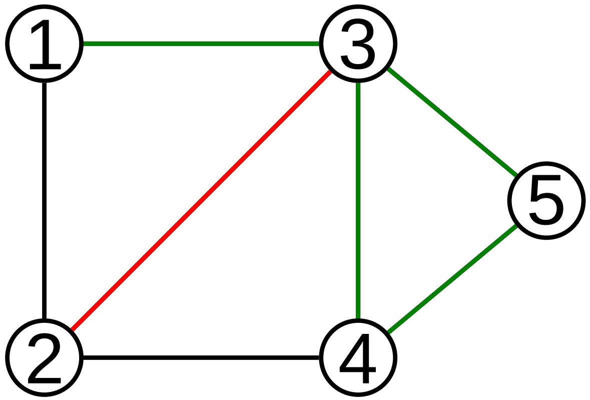


Un sendero cerrado en G:

$$P := \langle \{1, 3\}, \{3, 5\}, \{4, 5\}, \{3, 4\}, \rangle$$

Conceptos básicos

Ejemplo:

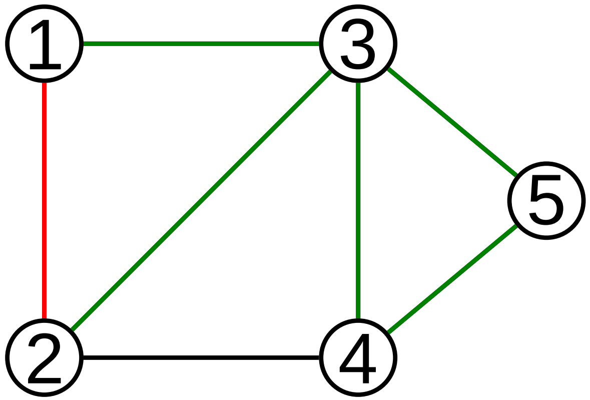


Un sendero cerrado en G:

$$P := \langle \{1,3\}, \{3,5\}, \{4,5\}, \{3,4\}, \{2,3\}, \rangle$$

Conceptos básicos

Ejemplo:

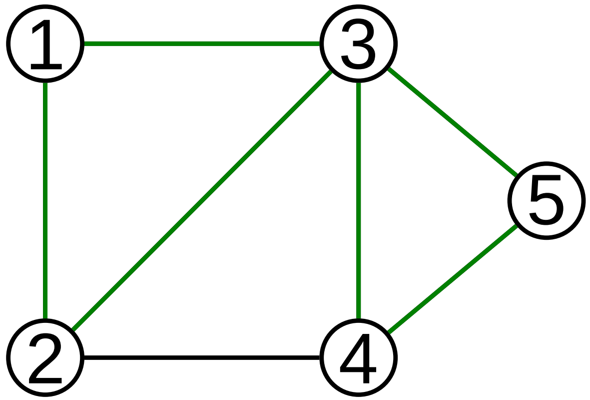


Un sendero cerrado en G:

$$P := \langle \{1,3\}, \{3,5\}, \{4,5\}, \{3,4\}, \{2,3\}, \{1,2\} \rangle$$

Conceptos básicos

Ejemplo:



Un sendero cerrado en G:

$$P := \langle \{1,3\}, \{3,5\}, \{4,5\}, \{3,4\}, \{2,3\}, \{1,2\} \rangle$$

Conceptos básicos

Cadenas orientadas y caminos

- Una cadena orientada P es una sucesión de arcos donde el extremo final de cada arco es el extremo inicial de su sucesor

$$P := \langle (i_0, i_1), (i_1, i_2), \ldots, (i_{k-1}, i_k) \rangle$$

- Una cadena orientada que no repite arcos se llama camino
- Un camino es elemental si no repite nodos
- Un camino es cerrado si termina donde empezó ($i_0 = i_k$)

Contenido

- Motivación
- Conceptos básicos
- Representaciones computacionales
- Problema del sendero euleriano
- Problema del árbol generador de peso mínimo
- Problemas de caminos más cortos

Representación computacional

Matriz de adyacencia:

Dado un grafo $G = (V, E)$, definimos su matriz de adyacencia $M(G) \in \mathcal{M}_{V \times V}$ por medio de:

$$m_{ij} := \begin{cases} 1, & \text{si } i \text{ y } j \text{ con nodos adyacentes} \\ 0, & \text{caso contrario} \end{cases}$$

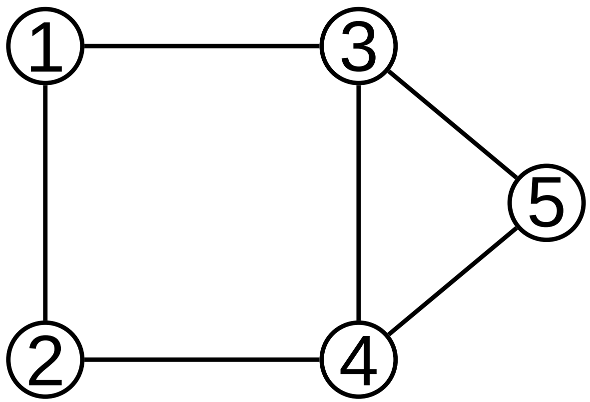
Representación computacional

Matriz de adyacencia:

Dado un grafo $G = (V, E)$, definimos su matriz de adyacencia $M(G) \in \mathcal{M}_{V \times V}$ por medio de:

$$m_{ij} := \begin{cases} 1, & \text{si } i \text{ y } j \text{ con nodos adyacentes} \\ 0, & \text{caso contrario} \end{cases}$$

Ejemplo:



$$M(G) = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Representación computacional

Matriz de incidencia:

Dado un grafo $G = (V, E)$, definimos su matriz de incidencia $H(G) \in \mathcal{M}_{V \times E}$ por medio de:

$$h_{ij} := \begin{cases} 1, & \text{si la arista } j \text{ es incidente al nodo } i \\ 0, & \text{caso contrario} \end{cases}$$

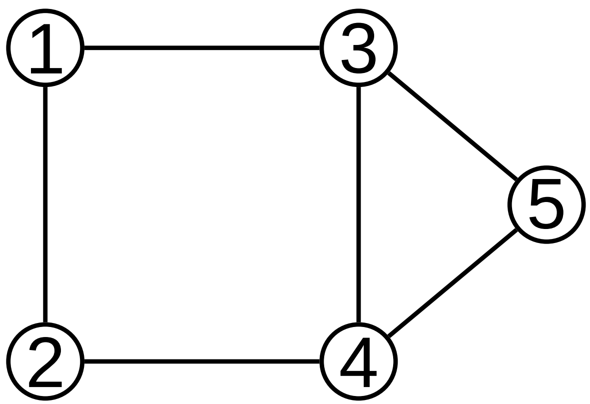
Representación computacional

Matriz de incidencia:

Dado un grafo $G = (V, E)$, definimos su matriz de incidencia $H(G) \in \mathcal{M}_{V\times E}$ por medio de:

$$h_{ij} := \begin{cases} 1, & \text{si la arista } j \text{ es incidente al nodo } i \\ 0, & \text{caso contrario} \end{cases}$$

Ejemplo:



$$H(G) = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Representación computacional

Listas de adyacencia:

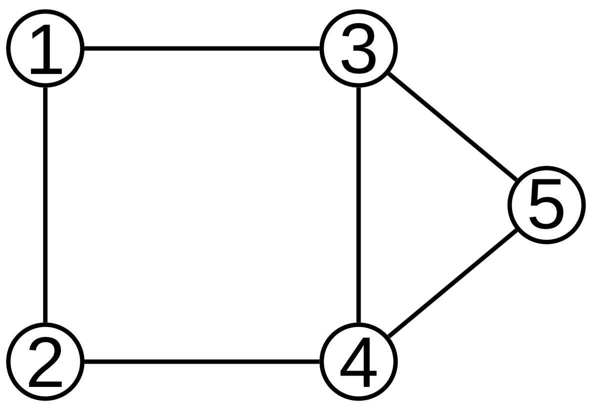
Dado un grafo $G = (V, E)$, almacenamos para cada nodo $i \in V$ una lista con su vecindad.

Representación computacional

Listas de adyacencia:

Dado un grafo $G = (V, E)$, almacenamos para cada nodo $i \in V$ una lista con su vecindad.

Ejemplo:



$$L[1] = \langle 2, 3 \rangle$$
$$L[2] = \langle 1, 4 \rangle$$
$$L[3] = \langle 1, 4, 5 \rangle$$
$$L[4] = \langle 2, 3, 5 \rangle$$
$$L[5] = \langle 3, 4 \rangle$$

Contenido

- Motivación
- Conceptos básicos
- Representaciones computacionales
- Problema del sendero euleriano
- Problema del árbol generador de peso mínimo
- Problemas de caminos más cortos

Dos problemas clásicos

Problema del sendero euleriano:

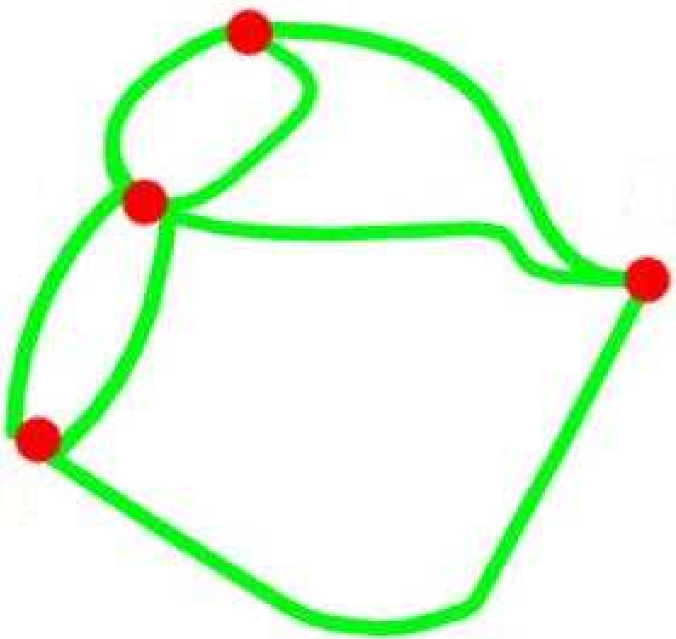
Dado un grafo $G = (V, E)$, determinar si existe un **sendero cerrado** que visite todas las **aristas** de G.

Dos problemas clásicos

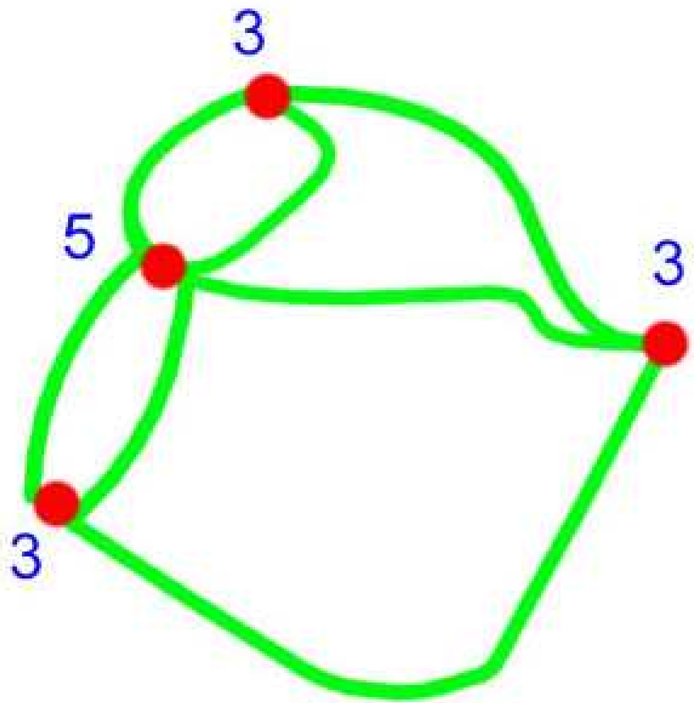
Problema del sendero euleriano:
Dado un grafo $G = (V, E)$, determinar si existe un **sendero cerrado** que visite todas las **aristas** de G.

Problema del ciclo hamiltoniano:
Dado un grafo $G = (V, E)$, determinar si existe un **sendero cerrado elemental** que visite todos los **nodos** de G.

Y los puentes?

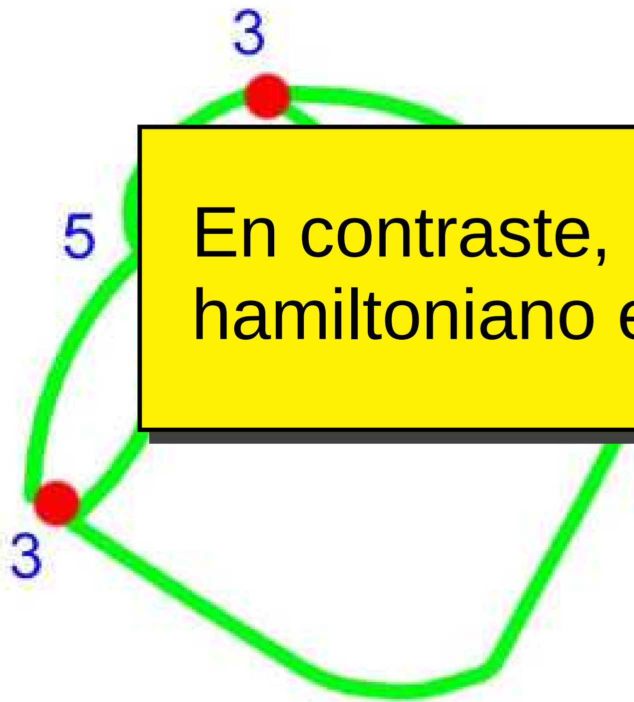


Y los puentes?



Teorema:
Un grafo admite un sendero euleriano cerrado si y sólo si todos sus nodos tienen grado par.

Y los puentes?



En contraste, el problema del ciclo hamiltoniano es muy **dificil**...

rrado

nodos tienen grado par.

Construyendo un circuito euleriano

Fase I: Orientación euleriana

- Seleccionar un nodo i con aristas incidentes sin orientar
- Iniciar un recorrido en i, empleando únicamente aristas sin orientar
- Terminar al llegar nuevamente a i
- Orientar las aristas del recorrido
- Repetir el proceso anterior mientras existan aristas sin orientar

Construyendo un circuito euleriano

Fase II: Búsqueda primero en profundidad (DFS)

- seleccionar un nodo $i \in V$
- `procesar(`i`)`

`procesar(`i`)`

- marcar los arcos entrantes a i
- para todo arco saliente no marcado (i, j): `procesar(`j`)`

Construyendo un circuito euleriano

Fase II: Búsqueda primero en profundidad (DFS)

- seleccionar un nodo $i \in V$
- `procesar(`i`)`

El orden de procesamiento de los nodos refleja un sendero euleriano

`procesar(`i`)`

- marcar los arcos entrantes a i
- para todo arco saliente no marcado (i, j): `procesar(`j`)`

Contenido

- Motivación
- Conceptos básicos
- Representaciones computacionales
- Problema del sendero euleriano
- Problema del árbol generador de peso mínimo
- Problemas de caminos más cortos

Árbol generador de peso mínimo

Ciclos, circuitos y conexidad

- Un ciclo es un sendero elemental cerrado
- Un circuito es un camino elemental cerrado
- Un grafo se llama conexo si para cada par de vértices $i, j \in V$ existe un sendero que tiene i y j por extremos
- Un grafo se llama fuertemente conexo si para cada par de vértices $i, j \in V$ existen un camino desde i hacia j y un camino desde j hacia i

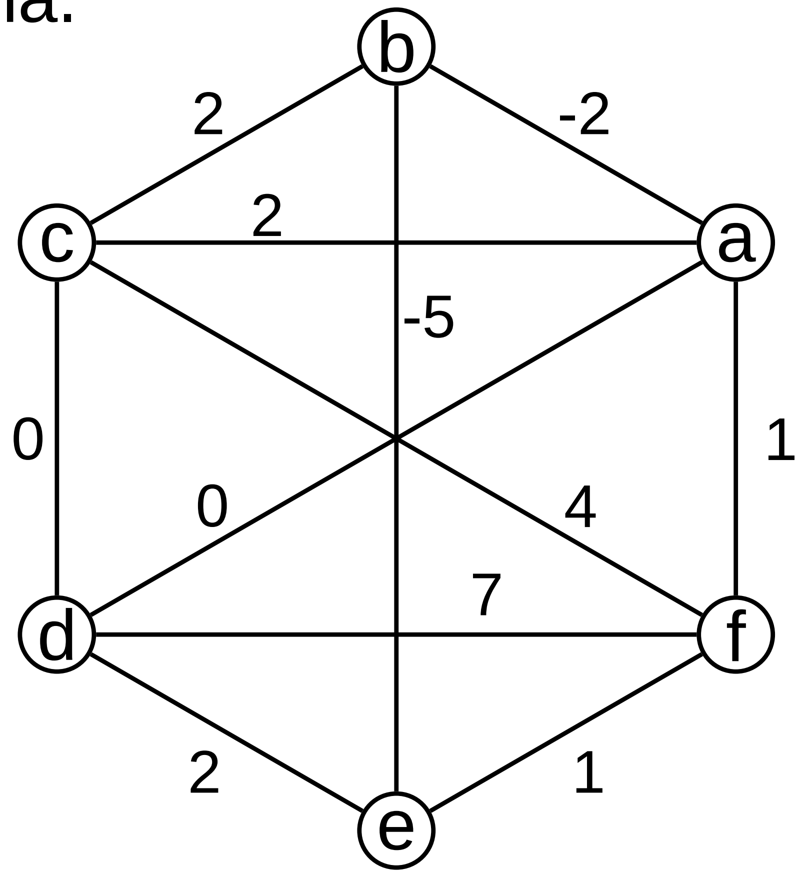
Árbol generador de peso mínimo

Árboles

- Un bosque es un grafo sin ciclos
- Un árbol es un grafo conexo sin ciclos
- Dado un grafo $G = (V, E)$, un **árbol generador** para G es un subgrafo de G que es un árbol y contiene todos los nodos de V

Árbol generador de peso mínimo

Problema del árbol generador de peso mínimo
[Minimum Spanning Tree, MST]
Dado un grafo G con pesos sobre las aristas, encontrar un árbol generador tal que la suma de los pesos de sus aristas sea mínima.



Árbol generador de peso mínimo

Algoritmo de Kruskal

1. Ordenar las aristas por sus pesos (ascendentemente)
2. $T := \emptyset$
3. Para $i := 1, \ldots, m$
4. $\quad$ $e := i$-ésima arista de la lista
5. $\quad$ Añadir e a T si no forma ciclos con las demás aristas en T

Contenido

- Motivación
- Conceptos básicos
- Representaciones computacionales
- Problema del sendero euleriano
- Problema del árbol generador de peso mínimo
- Problemas de caminos más cortos

Caminos más cortos

Problema de caminos más cortos
[Shortest Path Problem, SPP]
Dados un grafo dirigido D con pesos (longitudes) sobre los arcos y un nodo de salida $s \in V$, encontrar caminos de longitud mínima desde s hacia los demás nodos.

La longitud de un camino es la suma de las longitudes de los arcos que lo componen.

Caminos más cortos

Ejemplo: