

número primo

entero positivo divisible sola-
mente entre uno y si mismo

$$x \bmod 2 = 0$$

par

$$x \bmod 2 = 1$$

impar

a	b	$a \oplus b$
F	F	F
F	T	T
T	F	T
T	T	F

XOR

a	$\neg a$
F	T
F	T
T	F
T	F

NOT

a	b	$a \wedge b$
F	F	F
F	T	F
T	F	F
T	T	T

AND

a	b	$a \vee b$
F	F	F
F	T	T
T	F	T
T	T	T

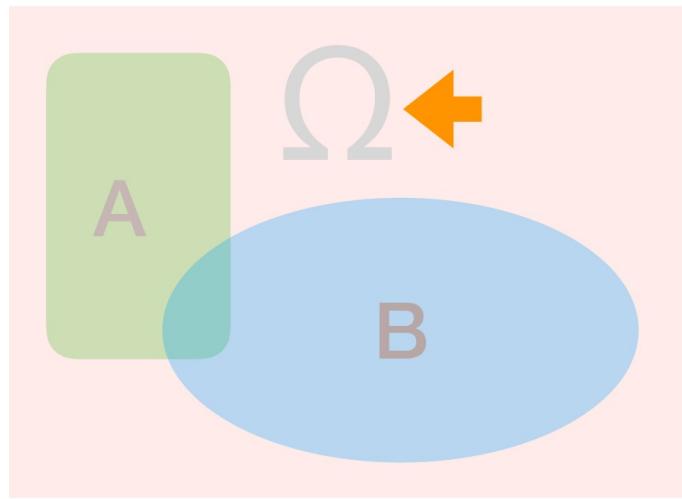
OR

a	b	$a \rightarrow b$
F	F	T
F	T	T
T	F	F
T	T	T

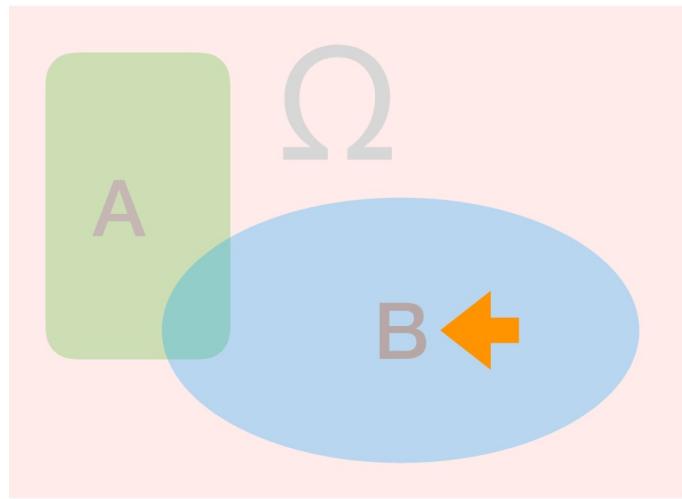
implicación

a	b	$a \leftrightarrow b$
F	F	T
F	T	F
T	F	F
T	T	T

equivalencia



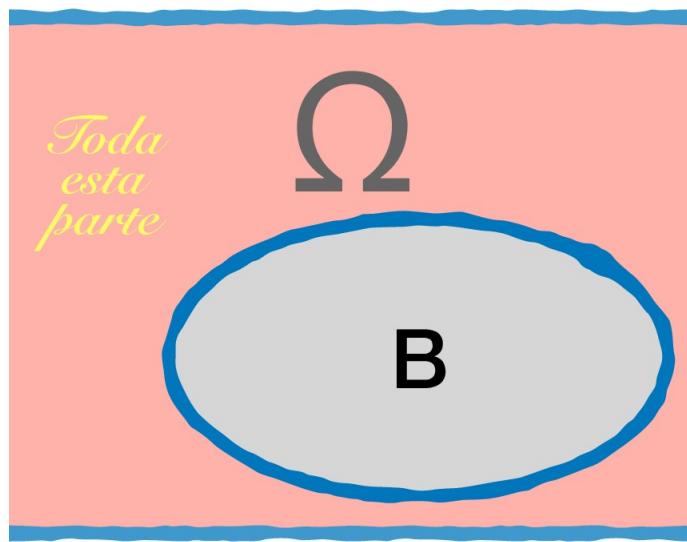
universo



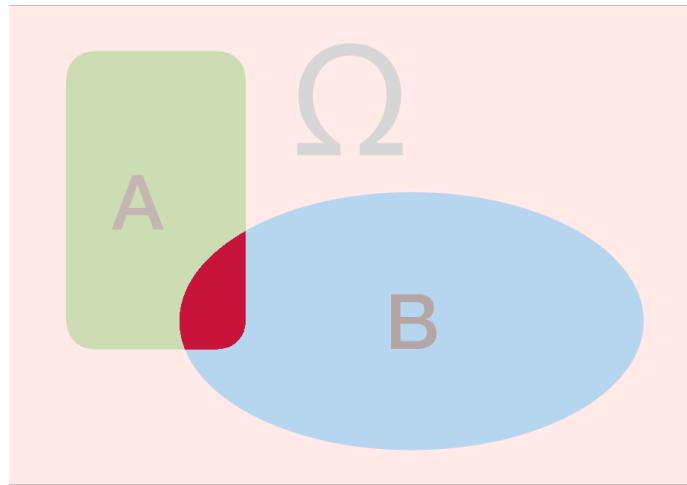
conjunto



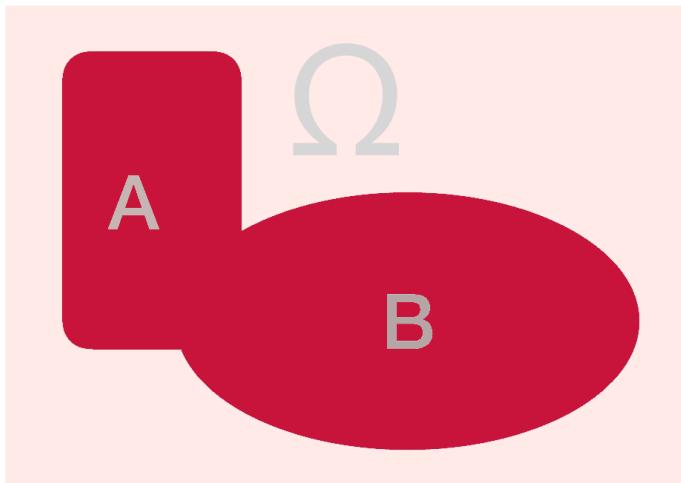
complemento de A



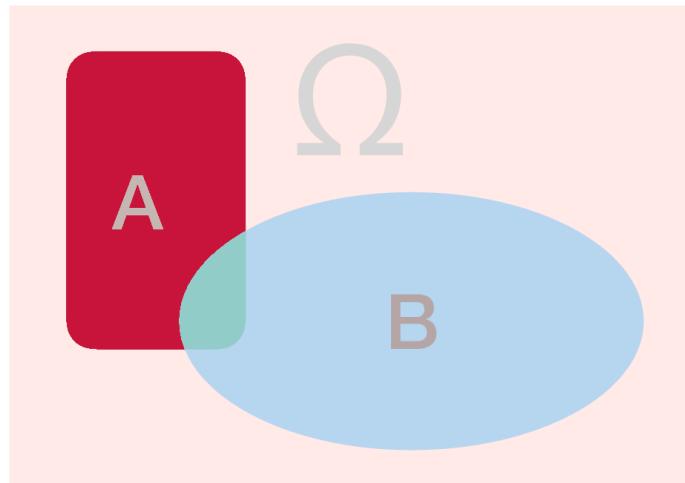
complemento de B

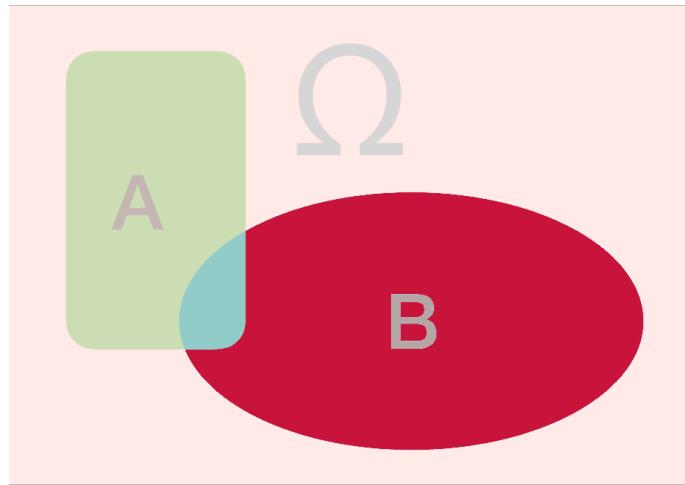


$$A \cap B$$



$$A \cup B$$


$$A \setminus B$$



$$B \setminus A$$

*i*ésimo

elemento en posición con
índice i

10₂

dos en binario

112

tres en binario

100₂

cuatro en binario

101₂

cinco en binario

110₂

seis en binario

111₂

siete en binario

1000_2

ocho en binario

1001_2

nueve en binario

1010₂

diez en binario

1011₂

once en binario

1100₂

doce en binario

1101₂

trece en binario

1110₂

catorce en binario

10000_2

dieciséis en binario

1111₂

quince en binario

10₈

ocho en octal

118

nueve en octal

128

diez en octal

*a*₁₆

diez en hexadecimal

*b*₁₆

once en hexadecimal

c₁₆

doce en hexadecimal

*d*₁₆

trece en hexadecimal

*e*₁₆

catorce en hexadecimal

*f*16

quince en hexadecimal

10_{16}

dieciséis en hexadecimal

1116

diecisiete en hexadecimal

12₁₆

dieciocho en hexadecimal

13₁₆

diecinueve en hexadecimal

1416

veinte en hexadecimal

*g*₂₀

dieciséis en base veinte

h₂O

diecisiete en base veinte

20

1 como potencia de dos

2¹

2 como potencia de dos

2²

4 como potencia de dos

2³

8 como potencia de dos

2⁴

16 como potencia de dos

25

32 como potencia de dos

2⁶

64 como potencia de dos

27

128 como potencia de dos

2⁸

256 como potencia de dos

29

512 como potencia de dos

2¹⁰

1024 como potencia de dos

2 11

2048 como potencia de dos

2¹²

4096 como potencia de dos

#ff0000

rojo en RGB

#00ff00

verde en RGB

#0000ff

azul en RGB

#ffff00

amarillo en RGB

#ff00ff

magenta en RGB

#00ffff

cian en RGB

#000000

negro en RGB

#ffffff

blanco en RGB

#e0e0e0

gris claro en RGB

#606060

gris oscuro en RGB

#6600cc

morado en RGB

#ff9933

naranja en RGB

#663300

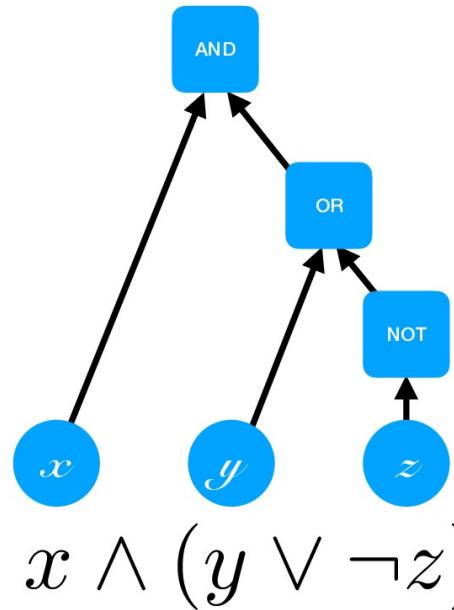
café en RGB

#ff99ff

rosa en RGB

precedencia

orden de evaluación de operadores



árbol de decisión

$$\log 1 = 0$$

logaritmo de uno es cero en
cualquier base

$$\log_2 8 = 3$$

dos al cúbico es ocho

$$\log_3 81 = 4$$

tres elevado a cuatro es 81

$$\log_{10} 100 = 2$$

diez al cuadrado es cien

máximo

no existe ninguno otro que
sea mayor

mínimo

no existe ninguno otro que
sea menor

maximal

no se le puede agregar elementos

minimal

no se le puede quitar elementos

fuente

vértice inicial de un flujo

sumidero

vértice destino de un flujo

bisección

división en dos partes

óptimo

mejor posible

factible

respeta restricciones

búsqueda

una serie de pasos para determinar la presencia de un elemento

recorrido

una serie de pasos para visitar a todos los elementos

distancia de edición

el costo total para convertir
una palabra en otra

problema de la mochila

elegir los objetos de mayor valor total respetando la capacidad

$$\Psi \geq \sum_{\varphi \in M} \omega(\varphi)$$

restricción de capacidad

$$\delta(G) = \frac{m}{m_{\max}}$$

densidad

$$\delta(G) = \frac{2m}{n(n-1)}$$

densidad

$$\text{diam}(G) = \max_{\substack{v \in V \\ w \in V}} \text{dist}(v, w)$$

diámetro

$$\frac{\sum_{v \in V} \deg(v)}{n} = \frac{2m}{n}$$

grado promedio



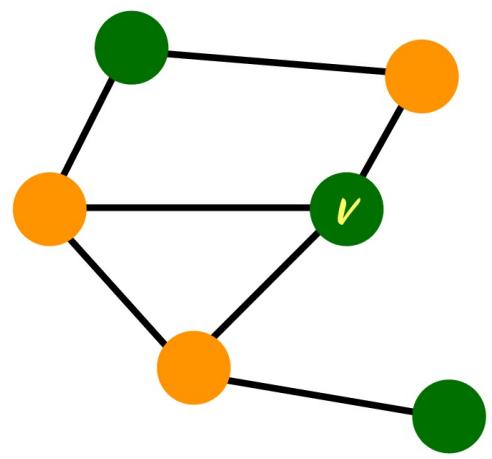
arista no dirigida



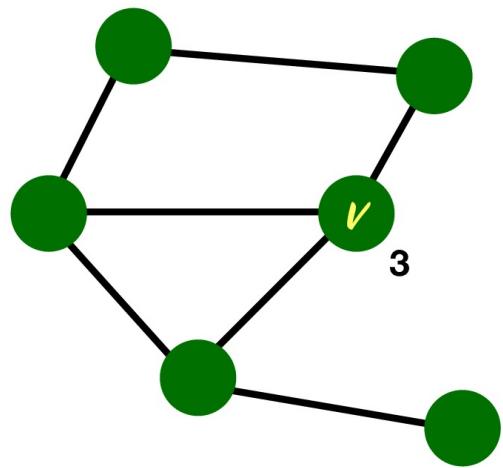
arista dirigida



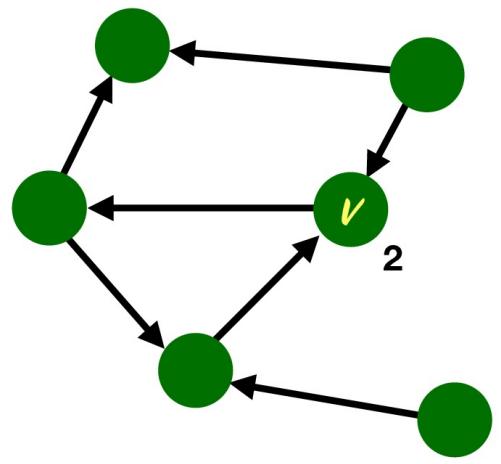
vértice



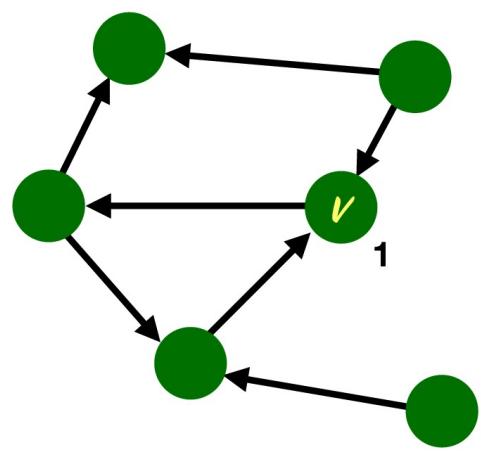
vecinos



grado

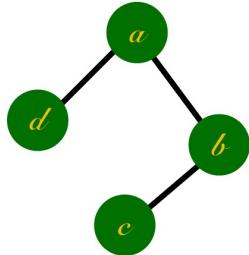


grado de entrada



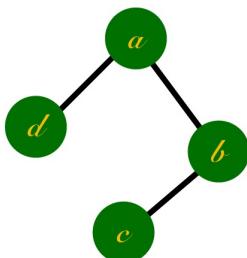
grado de salida

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



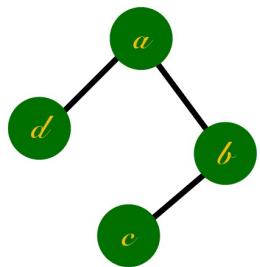
matriz de adyacencia

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>
<i>a</i>	0	1	2	1
<i>b</i>	1	0	1	2
<i>c</i>	2	1	0	3
<i>d</i>	1	2	3	0

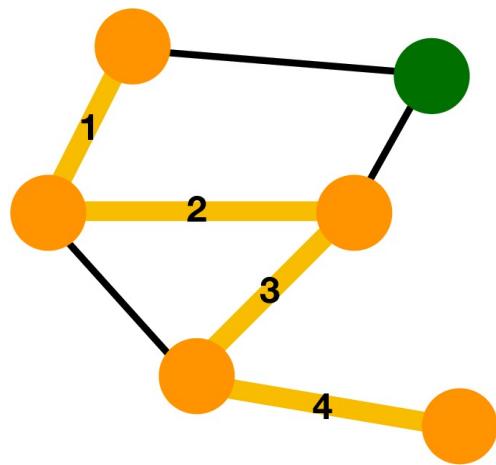


matriz de distancias

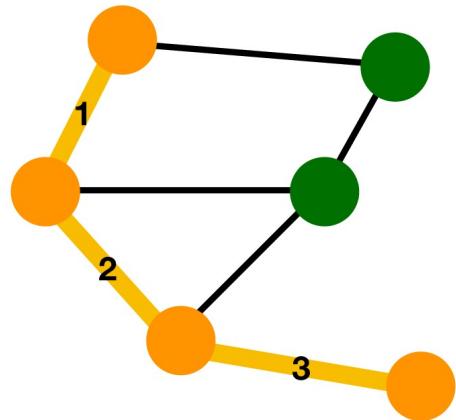
	a	b	c	d
a	0	1	2	1
b	1	0	1	2
c	2	1	0	3
d	1	2	3	0



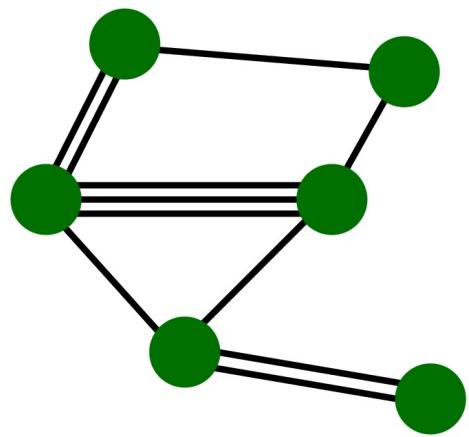
diámetro



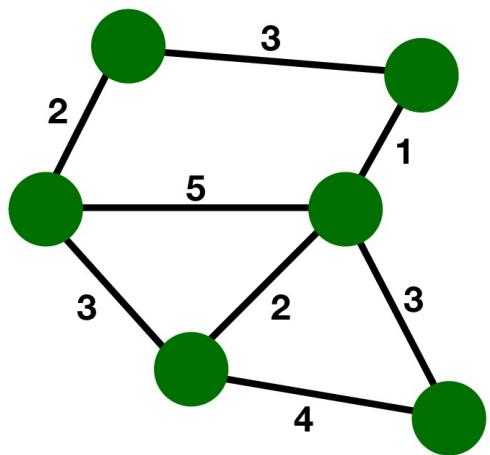
camino entre dos vértices



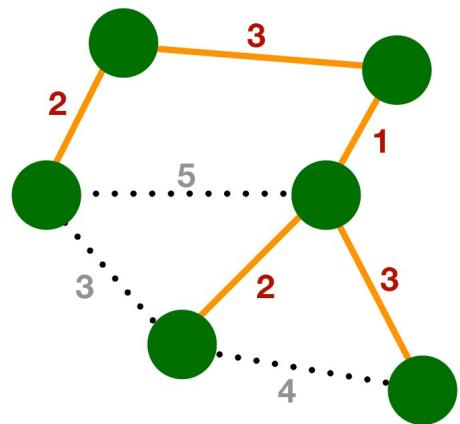
camino más corto entre dos vértices



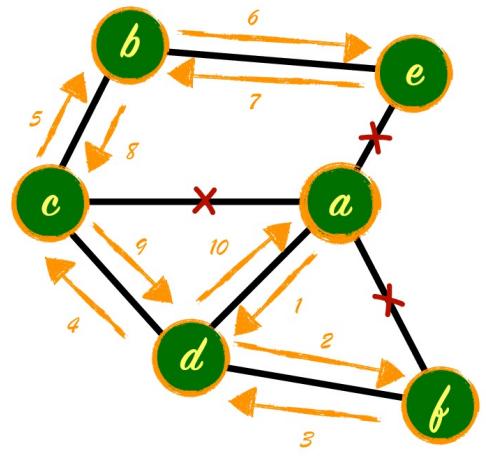
multigrafo



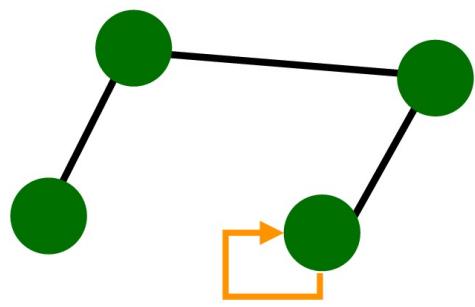
grafo ponderado



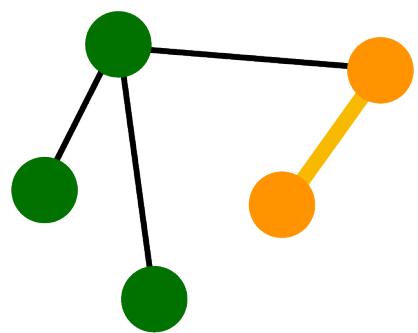
árbol de expansión mínima
(MST)



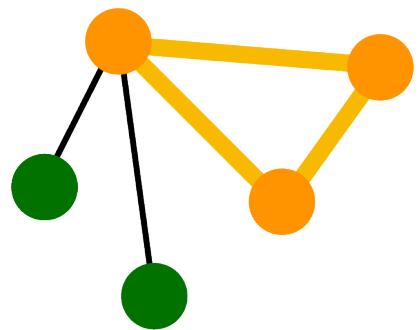
recorrido por profundidad
(DFS)



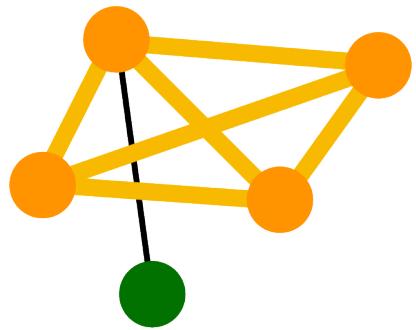
bucle



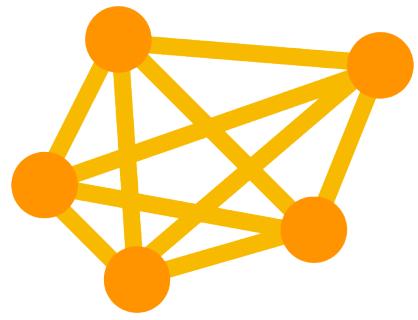
camarilla de dos vértices



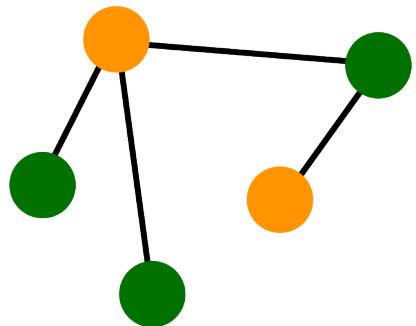
camarilla de tres vértices



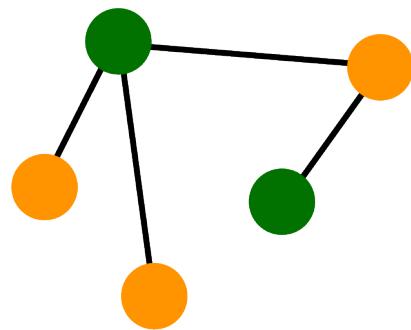
camarilla de cuatro vértices



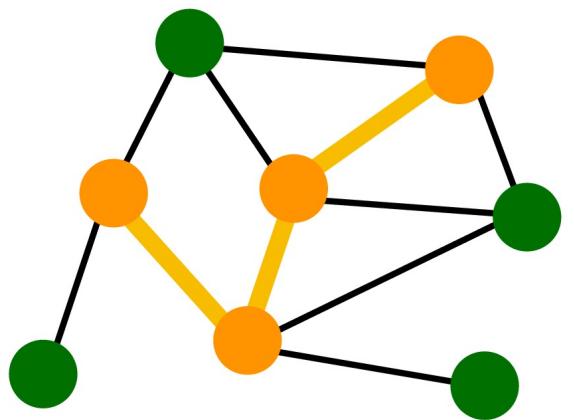
camarilla de cinco vértices



conjunto independiente de
dos vértices



conjunto independiente de tres vértices



subgrafo inducido



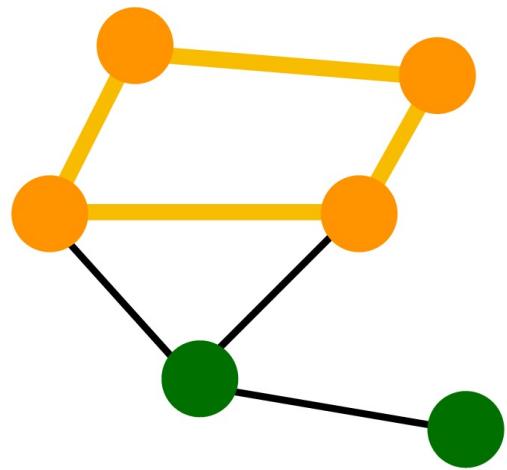
algoritmo



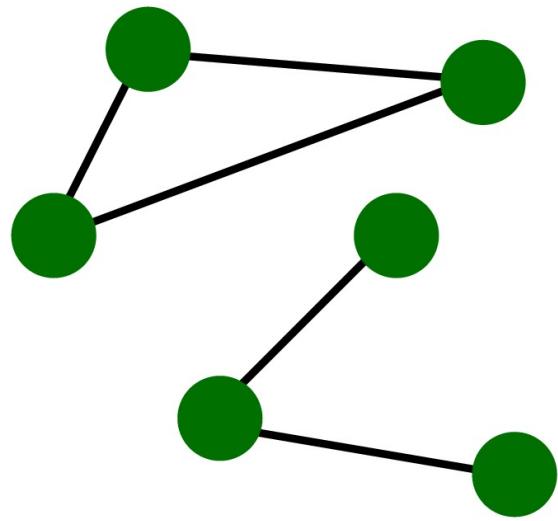
entrada



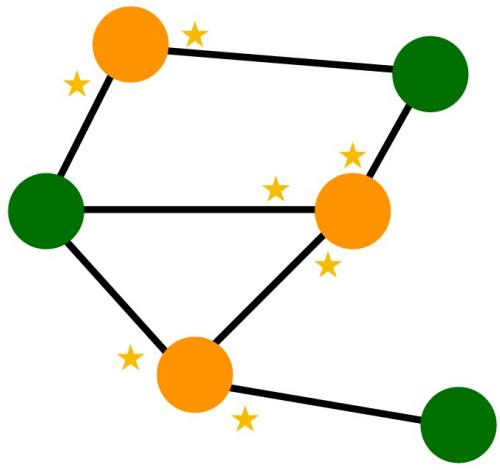
salida



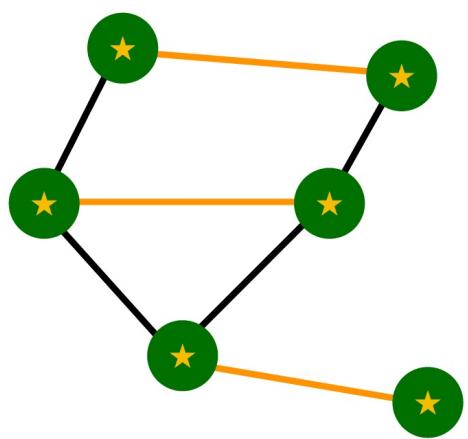
ciclo de largo cuatro



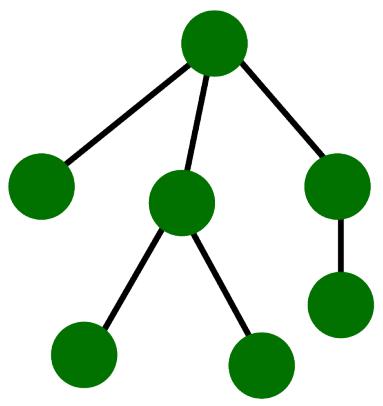
desconexo



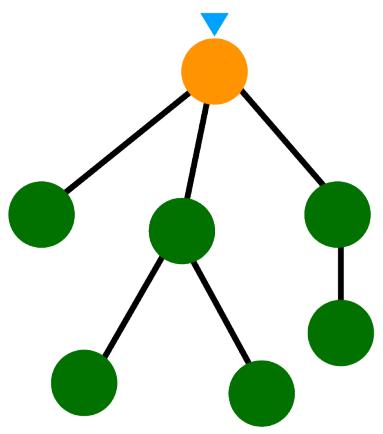
cubierta de vértices



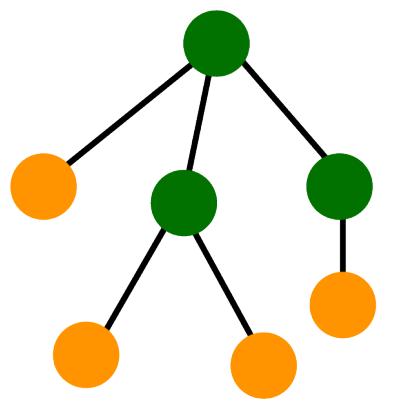
cubierta de aristas



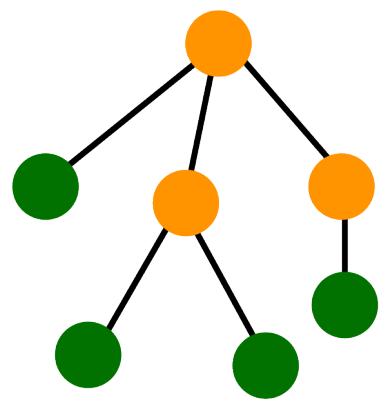
árbol



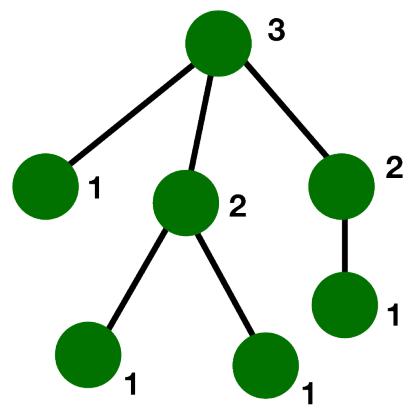
raíz



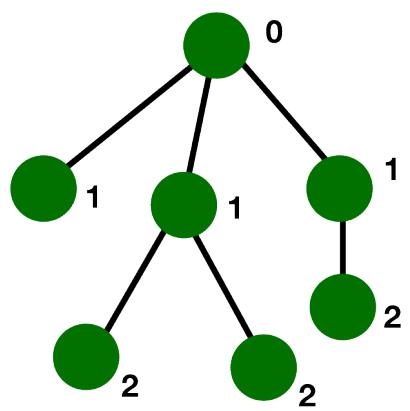
hojas



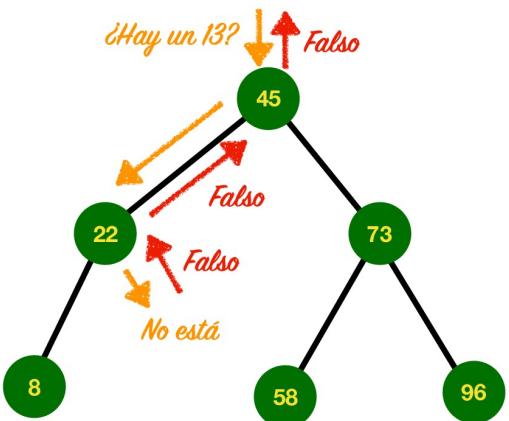
nodos internos



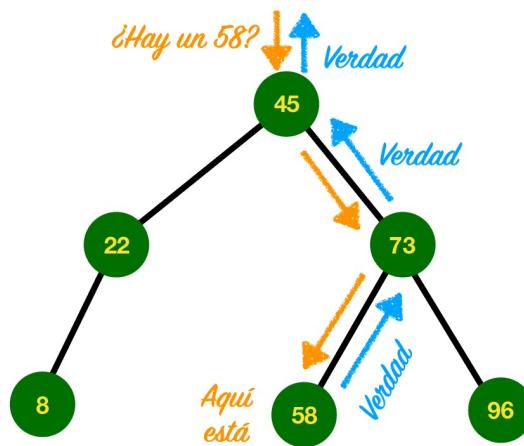
altura de cada nodo



profundidad de cada nodo



búsqueda fracasada



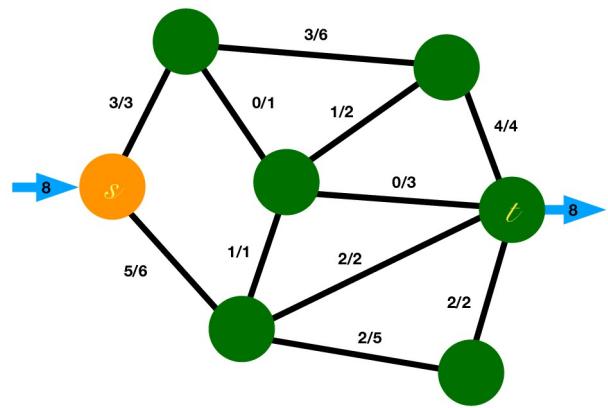
búsqueda exitosa

		D	I	F	I	C	I	L
	0	1	2	3	4	5	6	7
F	1	1	2	2	3	4	5	6
A	2	2	2	3	3	4	5	6
C	3	3	3	3	4	3	4	5
I	4	4	3	4	3	4	3	4
L	5	5	4	4	4	4	4	3

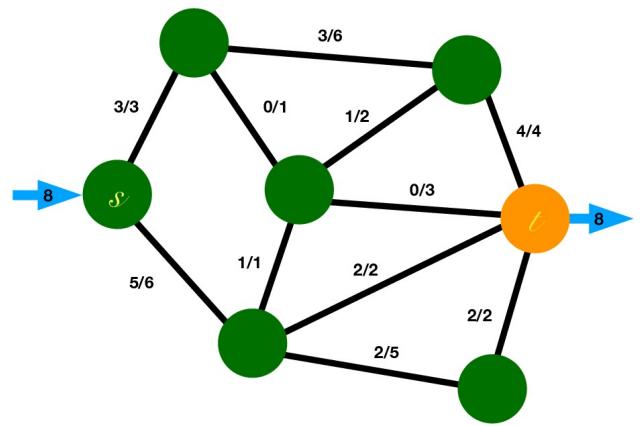
distancia de edición



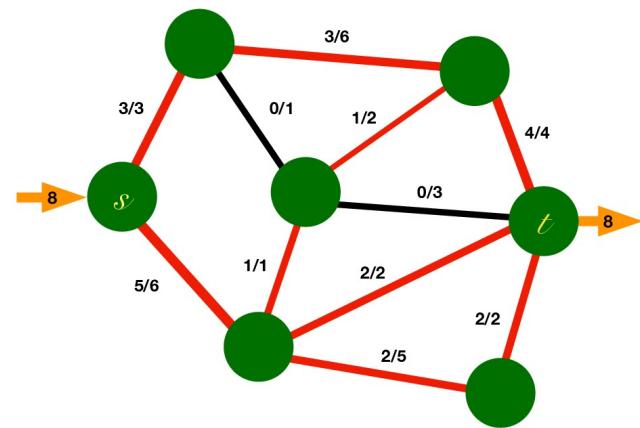
problema de la mochila



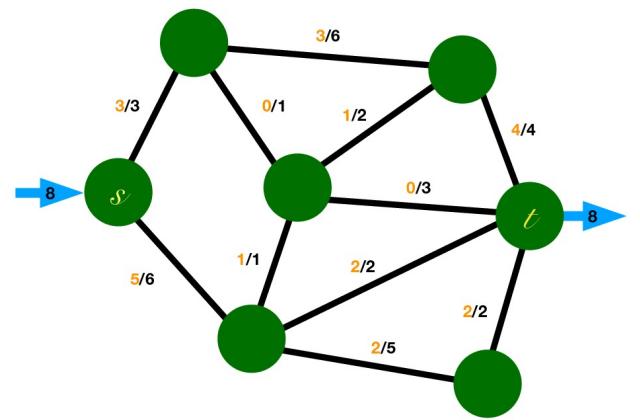
fuente



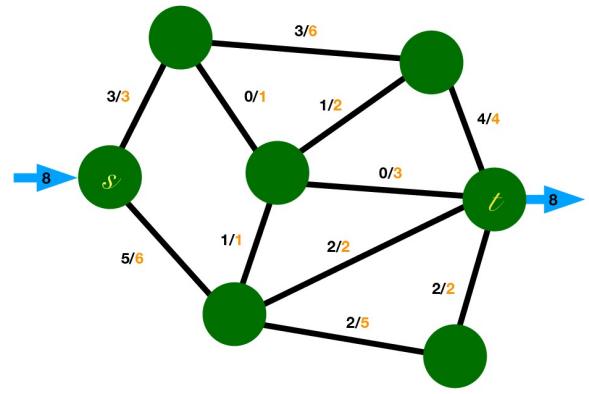
sumidero



flujo factible



niveles de flujo de las aristas



capacidades de las aristas

árbol

grafo simple conexo y acíclico

bósque

grafo simple desconexo y
acíclico

Floyd-Warshall

algoritmo para calcular distancias

distancia

largo del camino más corto

diámetro

distancia máxima

camino

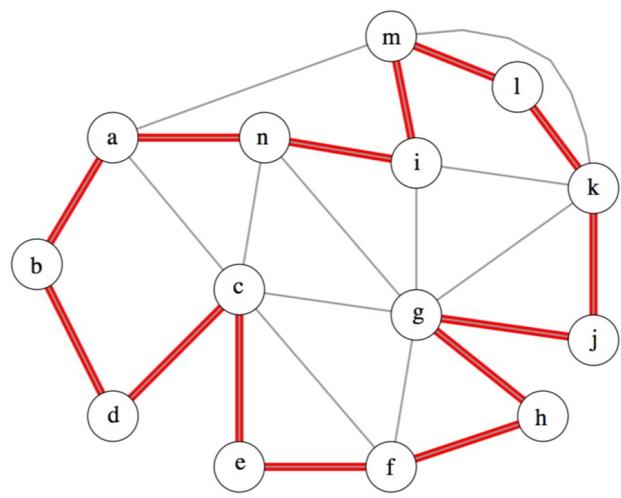
secuencia de aristas consecutivas entre vértices

camino de Hamilton

un camino que visita cada vértice exactamente una vez

ciclo de Hamilton

un ciclo que visita cada vértice exactamente una vez



ciclo (de Hamilton)

$L \in \text{TIME}(f(n))$

lenguaje decidido con TM determinista

$L \in \text{NTIME}(f(n))$

lenguaje decidido con TM no determinista

$$P = \bigcup_{k>0} \text{TIME}(n^k)$$

lenguajes decididos en tiempo polinomial con TM determinista

$$\text{NP} = \bigcup_{k>0} \text{NTIME}(n^k)$$

lenguajes decididos en tiempo polinomial con TM no determinista

p

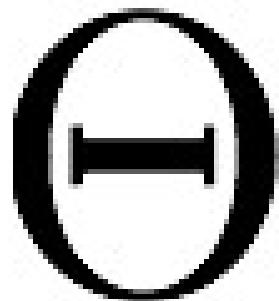
ro

Z+

enteros positivos

$\Theta(\dots)$

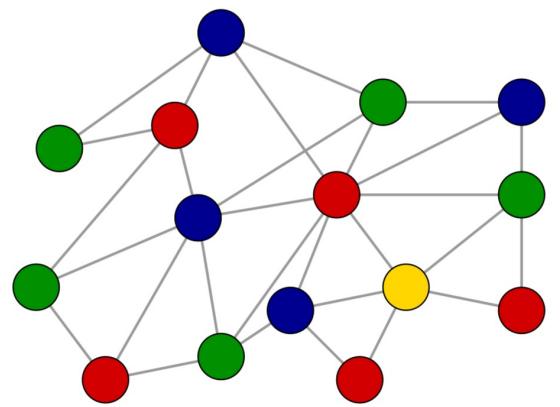
asintóticamente iguales



theta

TSP

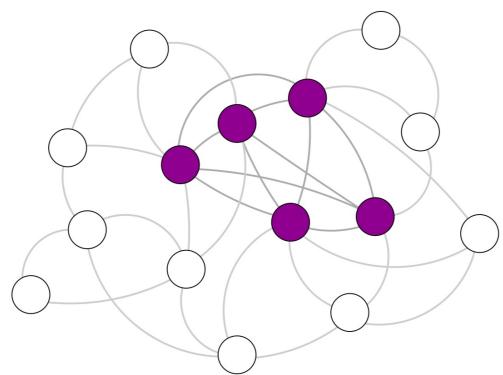
problema del viajante (traveling salesman problem)



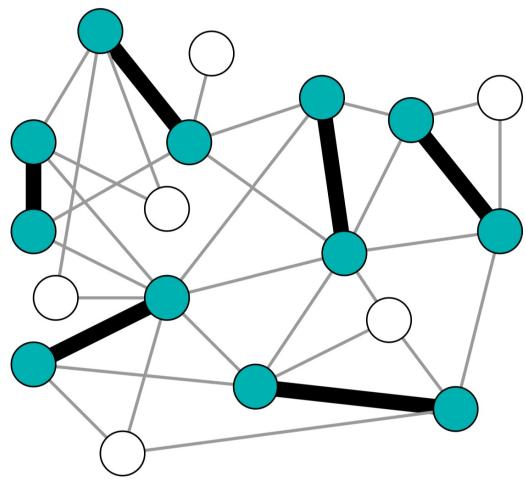
coloreo

KNAPSACK

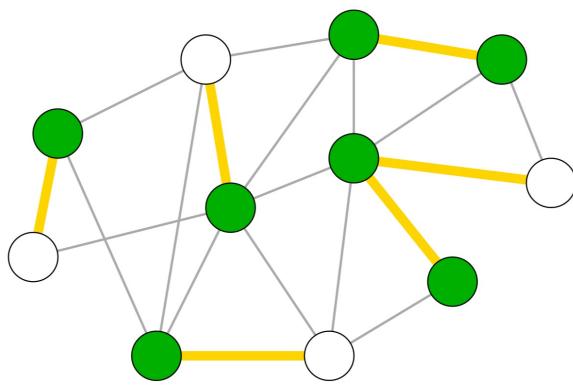
problema de la mochila



camarilla



acoplamiento



cubierta

Ford-Fulkerson

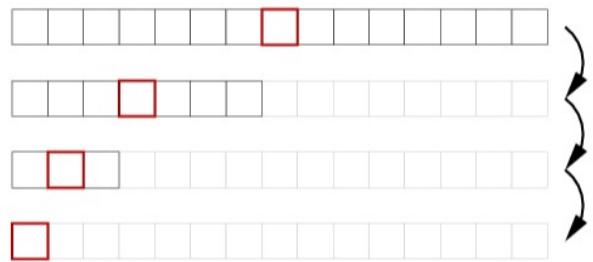
algoritmo para flujo máximo

Prim & Kruskal

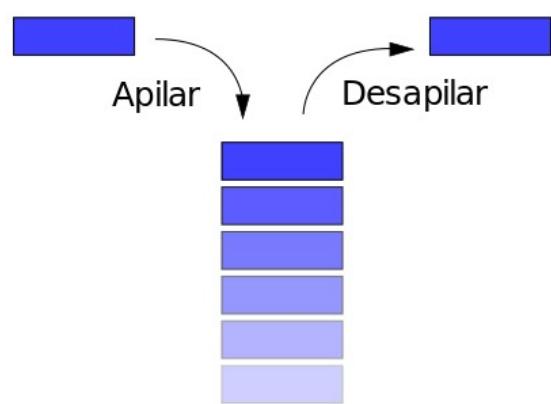
algoritmos para MST

MST

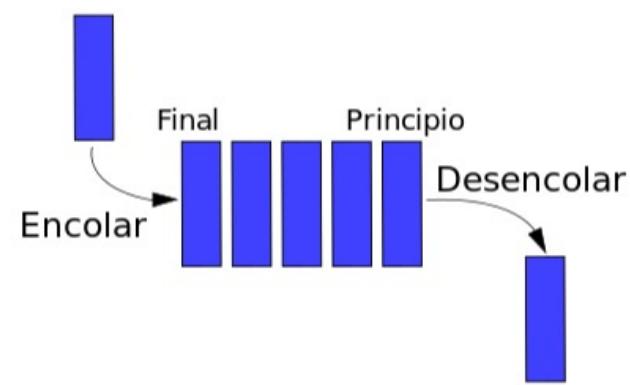
árbol de expansión mínima
(minimum spanning tree)



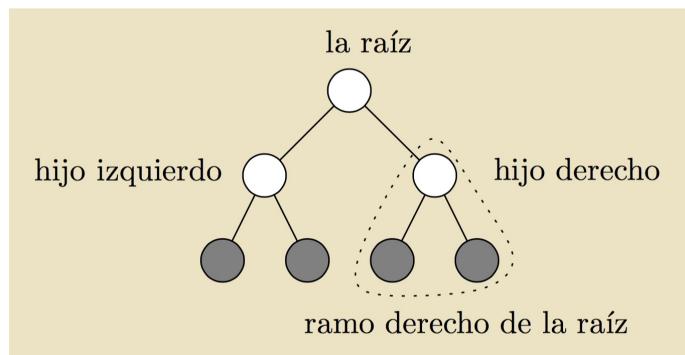
búsqueda binaria



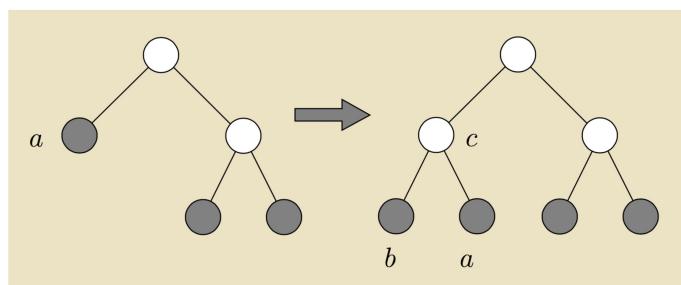
pila



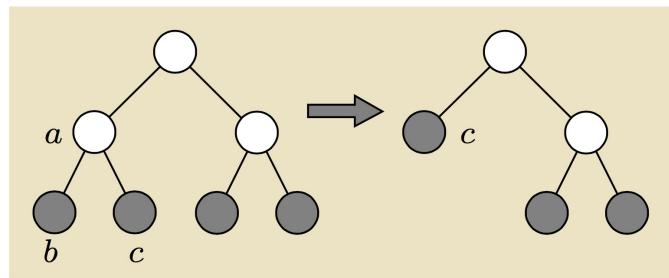
cola



árbol binario



inserción



eliminación

DFS

búsqueda por profundidad
(depth-first search)

BFS

búsqueda por anchura
(breadth-first search)

hoja

nodo que no tiene hijos

raíz

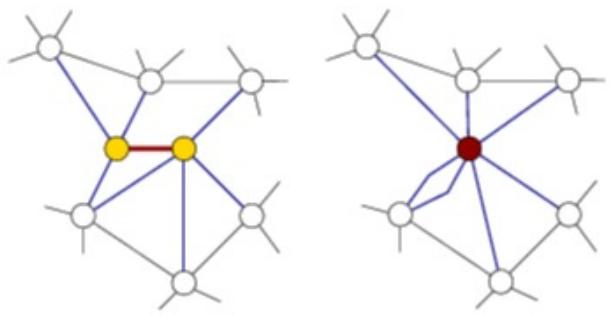
nodo acceso de un árbol

heurística

algoritmo no exacto

óptimo

mejor solución posible



contracción

ab**

potencia

a // b

división entera

a % b

módulo



igual

!=

no igual

from... import...

cargar rutina

$$A = \{1, 2\}$$

conjunto

print()

imprimir

`len(A)`

cardinalidad

3 in A

pertenece

not

negación

and

y

or

O

A & B

intersección

A | B

unión

A - B

diferencia

`A.issubset(B)`

subconjunto

`A.issuperset(B)`

superconjunto

all()

para todos

any()

existe

a & b

y por bit

a | b

o por bit

a ^ b

xor por bit

$\sim a$

negación por bit

bin(n)

conversión de decimal a binario (base 2)

`oct(n)`

conversión de decimal a octal
(base 8)

hex(n)

conversión de decimal a hexadecimal (base 16)

<=

menor o igual

\geq

mayor o igual

def

subrutina

return

regresa

$$(a \wedge (a \rightarrow b)) \Rightarrow b$$

modus ponens

$$((a \rightarrow b) \wedge \neg b) \Rightarrow \neg a$$

modus tollens

$$\begin{aligned} & ((a \rightarrow b) \wedge (b \rightarrow c)) \\ & \Rightarrow (a \rightarrow c) \end{aligned}$$

silogismo

$$\models \phi$$

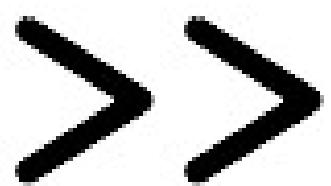
tautología

$$f_\phi(\tau) = \begin{cases} \top, & \text{si } \tau \models \phi, \\ \perp, & \text{si } \tau \not\models \phi \end{cases}$$

función booleana

<<

desplazar bits a la izquierda



desplazar bits a la derecha

for

ciclo sobre secuencia

while

ciclo con condición

range()

secuencia de enteros

set()

conjunto vacío

`list()`

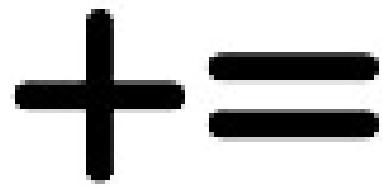
lista vacía

[3, 7, 1]

lista inicializada

*** =**

asignar múltiple del valor anterior



incremento

sqrt()

raíz cuadrada

a[0]

primer elemento

a[-1]

último elemento

assert

asegurar condición

if

ejecución condicional

a.append(3)

anexar a una lista

a.add(3)

agregar a un conjunto

elif

condición secundaria

else

condición de otro caso

sum

sumatoria

`quit()`

salir

M = {1: 3, 4: 5}

mapeo inicializado

`dict()`

mapeo vacío

$$|x| = \begin{cases} x, & \text{si } x \geq 0 \\ -x, & \text{si } x < 0 \end{cases}$$

valor absoluto

abs()

valor absoluto

`exp()`

función exponencial

log()

logaritmo natural

$\log(x, 2)$

logaritmo base dos

floor()

piso

ceil()

techo

round()

redondeo

pi

pi

$\exp(1)$

e

True

verdad

False

falso

```
def gcd(a, b):
    if b == 0:
        return a
    return gcd(b, a % b)
```

recursión

$p \in K$	$\sigma \in \Sigma$	$\delta(p, \sigma)$
s	0	$(s, 0, \rightarrow)$
s	1	$(s, 1, \rightarrow)$
s	\sqcup	(q, \sqcup, \leftarrow)
s	\triangleright	$(s, \triangleright, \rightarrow)$
q	0	(alto, 1, -)
q	1	$(q, 0, \leftarrow)$
q	\triangleright	(alto, \triangleright , \rightarrow)

función de transición

class

clase

self

autoreferencia

$$\Psi \geq \sum_{\psi \in M} \omega(\psi)$$

restricción de KNAPSACK

$$\max_{M \subseteq N} \sum_{\psi \in M} \nu(\psi)$$

función objetivo de KNAPSACK

NP-completo

problema al cual no se conocen soluciones eficientes en todos los casos

fuente

vértice inicial en un problema de flujo máximo

sumidero

vértice meta en un problema
de flujo máximo

$\mathcal{O}(n)$

complejidad asintótica lineal

$\mathcal{O}(\log n)$

complejidad asintótica logarítmica

$\mathcal{O}(n^k)$

complejidad asintótica polinomial

$\mathcal{O}(2^n)$

complejidad asintótica exponencial a base dos

$\mathcal{O}(1)$

complejidad asintótica constante

altura

uno para hojas, el máximo de
las alturas de los hijos más
uno para los demás

profundidad

cero para la raíz, uno mayor
a la del padre para los demás

hijo

un nodo dependiente

padre

un nodo del cual se depende

árbol balanceado

diferencia limitada de alturas
en los ramos

árbol binario lleno

dos o cero hijos; se permite
un nodo con un hijo

hoja

nodo sin hijos

raíz

nodo sin padre

montículo

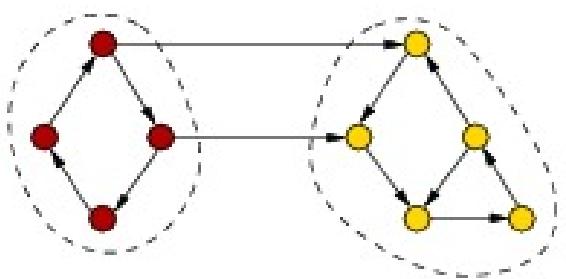
lista de árboles

recorrido

método que visita a cada vértice

búsqueda

método que encuentra un vértice



componentes fuertemente
conexos

componente conexo

subconjunto de vértices con
caminos entre todos

conexo

tiene un sólo componente conexo

desconexo

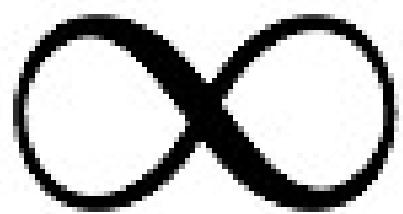
tiene dos o más componentes
conexos

ciclo

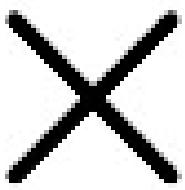
un camino que regresa a su
inicio

bucle

arista reflexiva



infinito



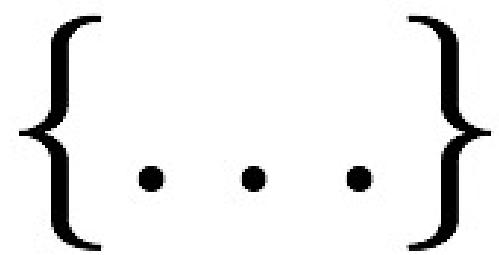
multiplicación

$$\{\dots\}^*$$

cero o más

$$\{\dots\}^+$$

uno o más



conjunto (elementos entre llaves)

X

vector (*minúsculas en negritas*)

M

matríg (mayúsculas en negritas)

a^b

potencia

2A

conjunto potencia

Z_k

conjunto residual

1 kB

kilobyte 1024 bytes

1 kb

kilobit 1024 bits

1 MB

megabyte 1024 kilobytes

1 Mb

megabit 1024 kilobits

1 GB

gigabyte 1024 megabytes

1 Gb

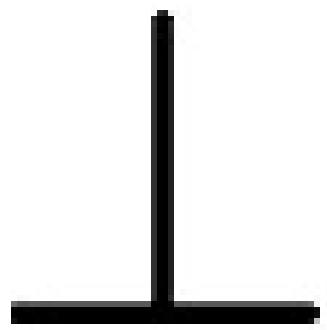
gigabit 1024 megabits

1 B

1 byte 8 bits

1 b

1 bit (vale cero o uno)



falso

T

verdad

*n*₂

binario (base 2)

*n*₈

octal (base 8)

*n*₁₆

hexadecimal (base 16)

*n*₁₀

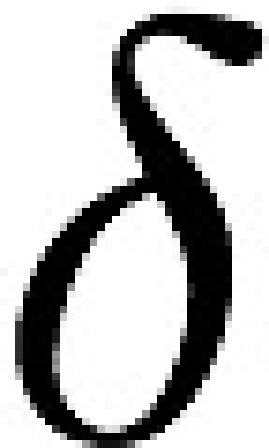
decimal (base 10)

n_k

base arbitraria

A faint, handwritten-style symbol resembling a stylized 'g' or 'y'. It consists of a curved loop on the left and a straight vertical stroke with a small hook on the right.

gamma minúscula

A large, stylized letter 'δ' (delta) in black ink, centered on the page. The letter has a thick, solid black outline and a white interior. It features a small loop on the left side and a larger, more prominent loop on the right side.

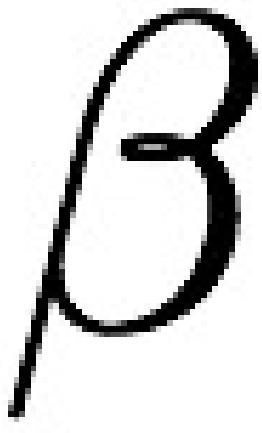
delta minúscula



delta mayúscula

α

alfa

A handwritten-style Greek letter beta (β) is shown. It features a large, oval-shaped loop on the left side, a short vertical stroke on the right side, and a horizontal bar across the middle of the loop.

beta

μ

mu

w

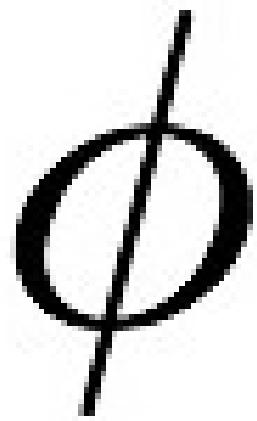
omega

$m \cdot b^\gamma$

punto flotante



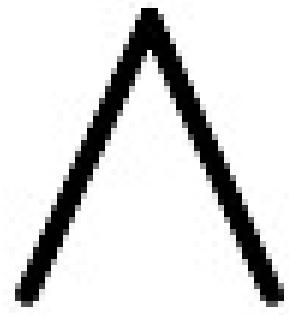
negación



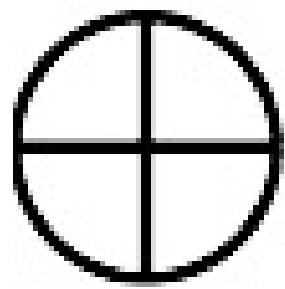
fi

V

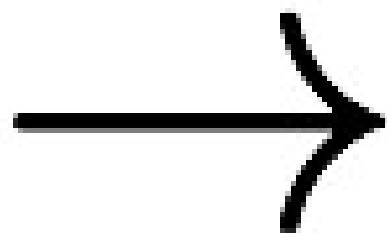
o (disyunción)



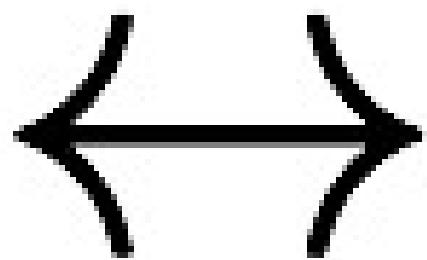
y (conjunción)



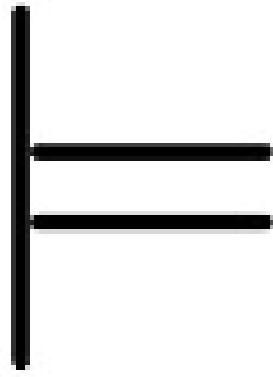
xor (o exclusivo)



implicación



equivalencia



satisface

tautología

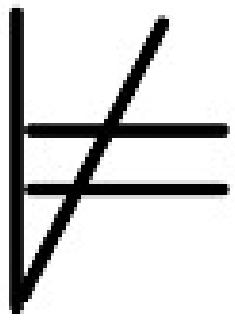
válida bajo cada asignación



equivalencia lógica

mod

módulo (residuo)



no satisfies

CNF

forma normal conjuntiva

DNF

forma normal disyuntiva

SAT

problema de satisfiabilidad

$\mathcal{O}(\dots)$

cota superior (complejidad
asintótica de peor caso)

$\Omega(\dots)$

cota inferior (complejidad
asintótica de mejor caso)

BDD

árbol de decisión binaria

k!

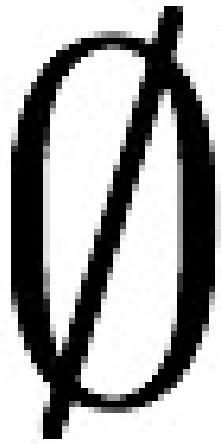
factorial

$$\binom{n}{k}$$

coeficiente binomial

1				
1	1			
1	2	1		
1	3	3	1	
1	4	6	4	1

triángulo de Pascal



conjunto vacío

G

grupo

π

pi 3.14159

e

número de Euler 2.71828

$$F_{k-1} + F_{k-2}$$

sucesión de Fibonacci (comienza con cero y uno)

$$\frac{1 + \sqrt{5}}{2}$$

tasa dorada 1.61803

$$\sum_k f_k \cdot g_{n-k}$$

convolución

$$x_{i+1} - x_i = d$$

serie aritmética

$$x_{i+1} = d \cdot x_i$$

serie geométrica

$$A \times B$$

producto cartesiano

$$\mathcal{R} \subseteq A \times B$$

relación

$$\forall a : a \mathcal{R} a$$

reflexiva

$$a \mathcal{R} b \Leftrightarrow b \mathcal{R} a$$

simétrica

$$aRb \wedge bRc$$

\Leftrightarrow

$$aRc$$

transitiva

$$g : A \rightarrow B$$

mapeo (de dominio a rango)

$$\forall b \exists a : g(a) = b$$

epiyectivo

$$\begin{aligned}\forall a_1, a_2 : \\ g(a_1) = g(a_2) \\ \Rightarrow a_1 = a_2\end{aligned}$$

inyectivo

biyectivo

inyectivo y epiyectivo

$$e^x = \exp(x)$$

función exponencial

$\log(x)$

logaritmo

$\ln(x)$

logaritmo natural

$$\log_k(x)$$

logaritmo en base arbitraria

$$\lfloor x \rfloor$$

piso (mayor entero menor o igual)

$$\lceil x \rceil$$

techo (menor entero mayor o igual)

primo

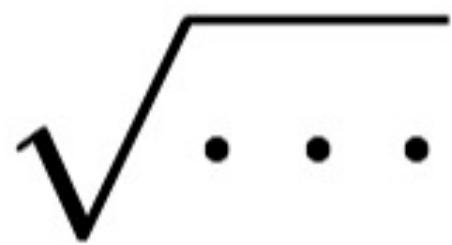
sin divisores además de uno
y sí mismo

impar

residuo uno al dividir entre
dos

par

residuo cero al dividir entre
dos



raíz cuadrada

$$[a, b]$$

intervalo cerrado

$[a, b)$

intervalo semicerrado (cierre izquierdo)

$$(a, b]$$

intervalo semicerrado (cierre derecho)

$$(a, b)$$

intervalo abierto

GCD

mayor divisor común (greatest common divisor)

$$a \perp b$$

relativamente primos

O

operador binario (operador de un grupo)

$$\begin{aligned}\forall g, h \in G : \\ i = g \circ h \in G\end{aligned}$$

clausura

$$(g \circ h) \circ i$$

=

$$g \circ (h \circ i)$$

asociatividad

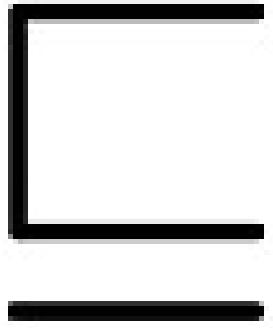
$$\exists e \in G :$$
$$\forall g \in G$$

$$e \circ g = g \circ e = e$$

identidad

$$\begin{aligned}\forall g \in G \\ \exists h \in G : \\ h \circ g = g \circ h = e\end{aligned}$$

inversión



subgrupo

$a | b$

divisible

$$\phi(n)$$

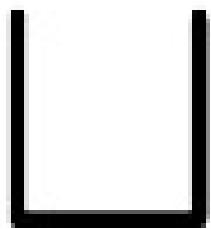
función de Euler

$$a^{p-1} \equiv 1 \pmod{p}$$

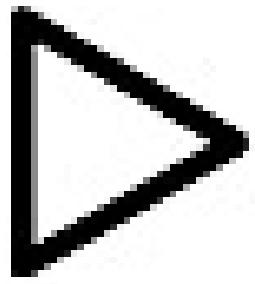
teorema de Fermat (para p primo)

$$M = (K, \Sigma, \delta, s)$$

máquina Turing



espacio blanco



inicio de cinta

$M(x) = \nearrow$

no detiene

$$\mathcal{G} = (V, E)$$

grafo

$v \in V$

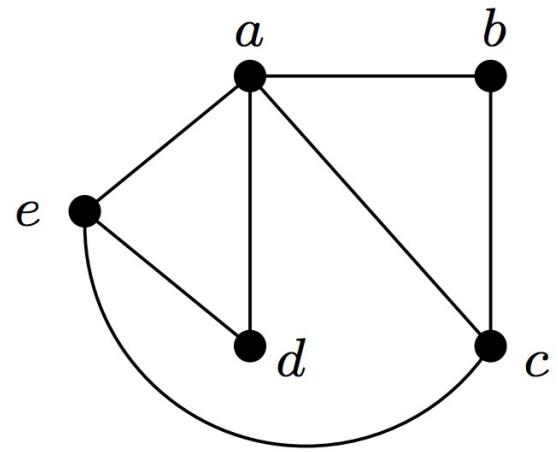
vértice

$$(u, v) \in E$$

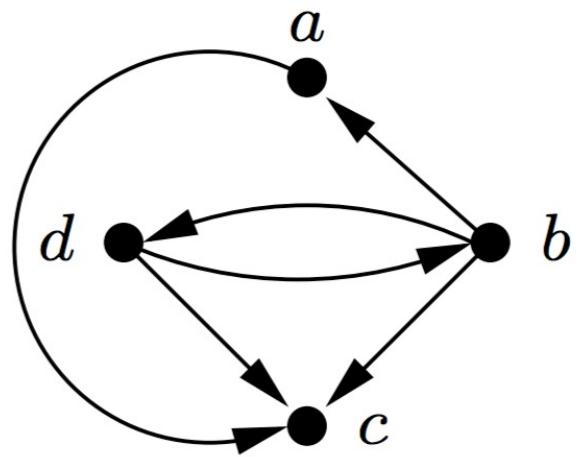
arista (no dirigida)

$$\langle u, v \rangle \in E$$

arista dirigida



grafo (no dirigido)



grafo dirigido

A

matriz de adyacencia

ponderado

cuenta con un peso

vecino

conectado por arista

$\deg(v)$

grado (número de vecinos)

$\overrightarrow{\deg}(v)$

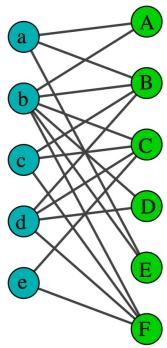
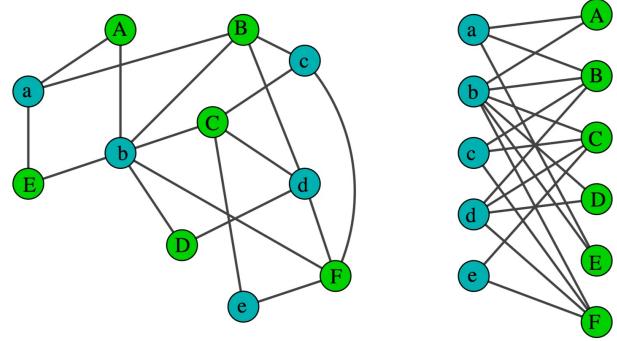
grado de salida

$\overleftarrow{\deg}(v)$

grado de entrada

k -regular

todos los grados valen k



bipartito

$$\Gamma(v)$$

vecindad

Г

gamma mayúscula

ψ

psi