Grafos

- Partes de un grafo: vértices y aristas
- Tipos de grafo:
  - Grafo simple: Sin bucles ni aristas múltiples No No
  - Multigrafo: Con aristas múltiples Sí No
  - Pseudografo: Con bucles Sí Sí
  - Grafo dirigido: Grafo simple con dirección Sí No
  - Multigrafo dirigido: Multigrafo con dirección Sí

- Familias distinguidas de grafos simples
  - Grafos completos: grafo simple donde cada par de vértices está conectado por una arista, aristas = $n(n-1)/2$
  - Ciclos: grafo que se asemeja a un polígono de n lados y n vértices
  - Rueda: grafo ciclo + un vértice que se une con todos los demás del grafo, aristas = $2(n-1)$

- Grafo bipartito: Aquel cuyos vértices se pueden separar en dos conjuntos en donde las aristas siempre van de un conjunto a otro

- Grafos definidos a partir de otro grafo
  - Subgrafo: grafo cuyo conjunto de vértices y aristas es un subconjunto de vértices y aristas de otro grafo
  - Unión de grafos: grafos con todos los vértices y aristas de los dos grafos a unir

Árboles
→ para cualquier par de vértices existe un camino

- Grafos conexos sin bucles con un nodo raíz
- Bosques: conjunto de árboles
- Un árbol m-ario es aquel en el que cada nodo puede tener hasta m hijos como máximo

Teoremas

- El número de vértices de grado impar en un grafo no dirigido es par
- Apretón de manos: la suma de los grados es el doble que el número de aristas

Isomorfismo (misma forma)

Se dice que dos grafos son isomorfos si existe una función biyectiva que permita transformar los vértices de uno en los vértices de otro

En un isomorfismo debe mantenerse
- Nº de aristas y nodos
- Adyacencias
- Grado de los vértices

(Esto no demuestra que sean isomorfos, es condición necesaria pero no es condición suficiente)

* Puede ser interesante mirar los caminos y circuitos ya que es invariante el nº de circuitos de la misma long

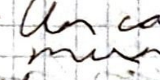
Camino

: Secuencia de aristas que comienza en un vértice del grafo y recorre ciertas aristas conectando vértices adyacentes

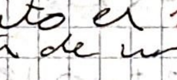
- La longitud de un camino es el nº de aristas que conecta
- Un camino es circuito si comienza y termina en el mismo vértice
- Un camino o circuito es simple si no contiene la misma arista más de una vez

Un grafo es conexo si existe un camino entre cada par de vértices del grafo

Si el camino es en ambos sentidos será Fuertemente conexo



fuertemente conexo



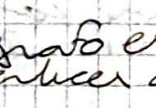
Debilmente conexo

Caminos / circuitos de Euler

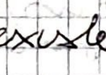
→ Circuito euleriano: circuito simple que contiene todas las aristas

→ Camino euleriano: camino simple que contiene todas las aristas del grafo

• Un multigrafo convexo tiene circuito euleriano si todos sus vértices tienen grado par



Grado impar no puede ser inicio ni final



Tampoco puede ser un nodo del medio

• Un multigrafo convexo tiene camino si tiene exactamente dos vértices de grado impar

Caminos / Circuitos de Hamilton

→ Circuito hamiltoniano: circuito simple que contiene todos los vértices del grafo sin repeticiones

→ Camino hamiltoniano: camino simple que contiene todos los vértices del grafo sin repetición

Un grafo con vértices de grado 1 no puede tenerlo

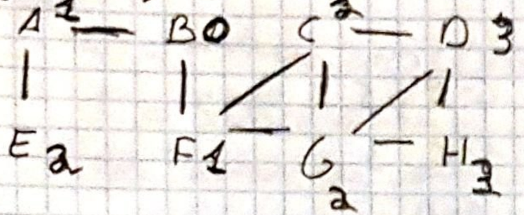
Busqueda en anchura / profundidad

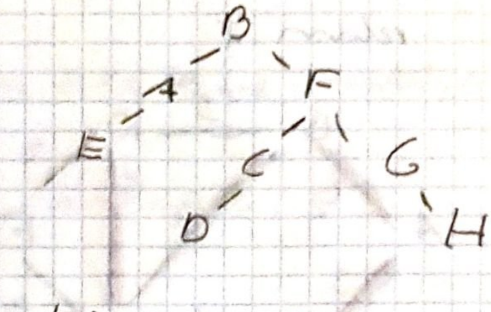
- Primero se visitan todos los hermanos antes de visitar a los hijos

Meter nodo raíz en cola y a partir de ahí

Sacar nodo
meter hijos en cola

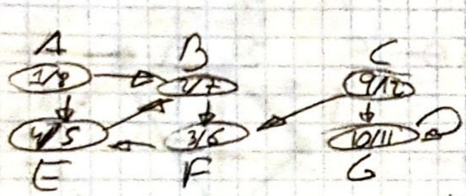
Ejemplo $S = B$





- Primero los hijos luego los hermanos

Explorar raíz
Explorar hijo izquierdo
Explorar hijo derecho



Algoritmos Prim / Kruskal

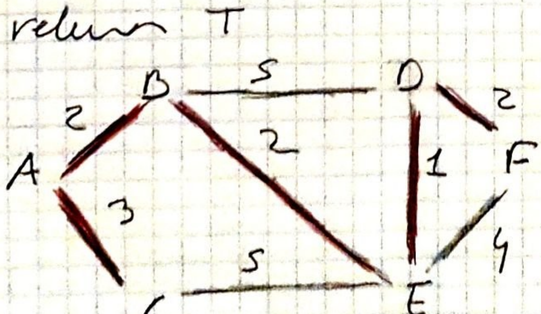
Árbol abarcador mínimo

↓ que accede a todos los nodos

↓ mínima suma posible

→ Kruskal

T = empty graph
for $i = 1$ to $n-2$
  e = any edge in G with smallest weight que no forme circuito
  T = T with e added
return T



Arista	Coste	Selección
DE	1	✓
AD	2	✓
BE	2	✓
DF	2	✓
AC	3	✓
EF	4	X
B D	5	X
C E	5	X

} Forman circ

→ Prim

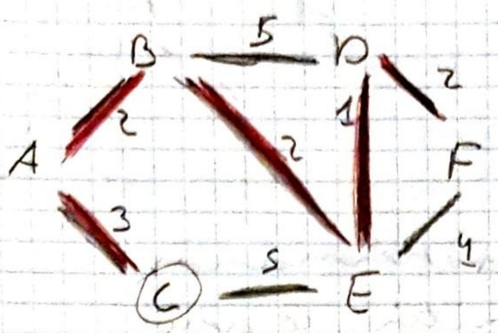
T = a minimum-weight edge

For i := 1 to n-2

e = an edge of minimum weight incident to a vertex T and not forming a simple circuit

T = T with e added

return T



C-E 5

A-C 3

A-B 2

B-D 5

B-E 2

D-E 1

E-F 4

D F 2

Dijkstra

For $i := 1$ to n

$L(v_i) := \infty$

$L(a) := 0$

$S = \emptyset$

while z NO pertenece a S

begin

u = vértice con $L(u)$ mínima entre los que no están en S

$S = S$ UNION $\{u\}$

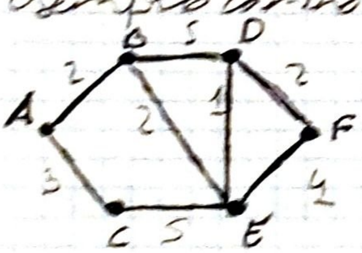
For todos los vértices v que no están en S

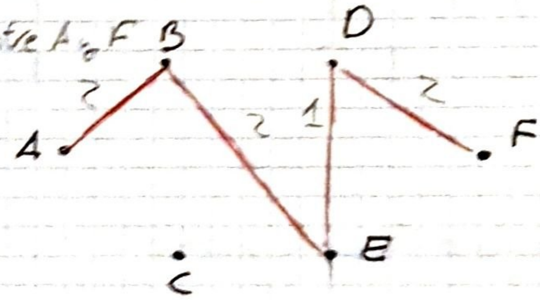
IF $(L(u) + w(u,v) < L(v))$

$L(v) = L(u) + w(u,v)$

end $L(z)$ = long del camino mínimo entre a y z

Ejemplo camino de menor coste entre A y F



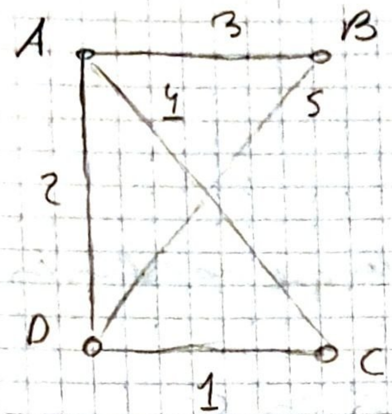


El camino de menor coste es de peso 6 y se obtiene recorriendo en orden inverso desde F

A	0^*	–	–	–	–	–	–
B	∞	2_a	–	–	–	–	–
C	∞	3_a	3^*_a	–	–	–	–
D	∞	∞	7_B	7_B	5^*_E	–	–
E	∞	∞	4_B	4^*_B	–	–	–
F	∞	∞	∞	∞	8_E	7^*_D	–

Algoritmo Warshall

Crear matrices por cada nodo en el que guardar el coste para llegar utilizando x nodo como pivote



Es simétrica por ser no dirigida

	A	B	C	D
A	0	3	4	2
B	3	0	∞	5
C	4	∞	0	1
D	2	5	1	0

Usando A como pivote

	A	B	C	D
A	0	3_{AB}	4_{AC}	2_{AD}
B	3_{BA}	0	7_{BAC}	5_{BD}
C	4_{CA}	7_{CAB}	0	1_{CD}
D	2_{DA}	5_{DB}	1_{DC}	0

$\infty > w(BA) + w(AC)$

Usando B como pivote

	A	B	C	D
A	0	3_{AB}	4_{AC}	2_{AD}
B	3_{BA}	0	7_{BAC}	5_{BD}
C	4_{CA}	7_{CAB}	0	1_{CD}
D	2_{DA}	5_{DB}	1_{DC}	0

Usando C como pivote

	A	B	C	D
A	0	3_{AB}	4_{AC}	2_{AD}
B	3_{BA}	0	7_{BAC}	5_{BD}
C	4_{CA}	7_{CAB}	0	1_{CD}
D	2_{DA}	5_{DB}	1_{DC}	0

Usando D como pivote

	A	B	C	D
A	0	3_{AB}	3_{ADC}	2_{AD}
B	3_{BA}	0	6_{BDC}	5_{BD}
C	3_{CDA}	6_{CDB}	0	1_{CD}
D	2_{DA}	5_{DB}	1_{DC}	0