Árboles generadores mínimos (MST)

Gregorio Hernández Peñalver

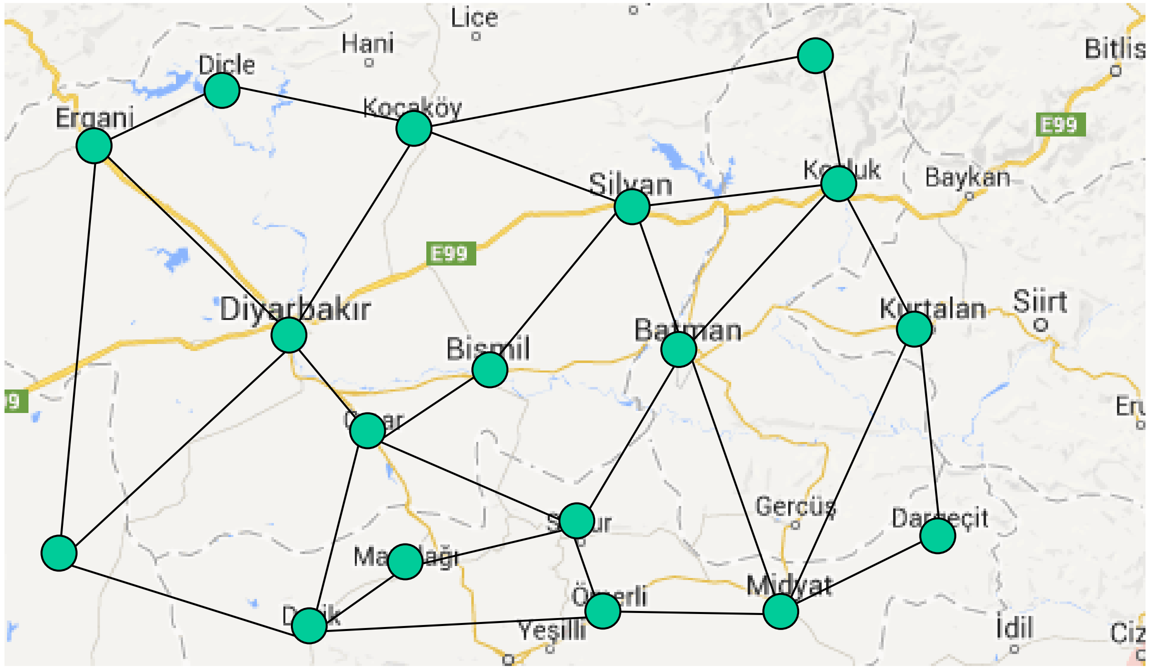
UPM

Matemática Discreta II

(MI)

Árboles

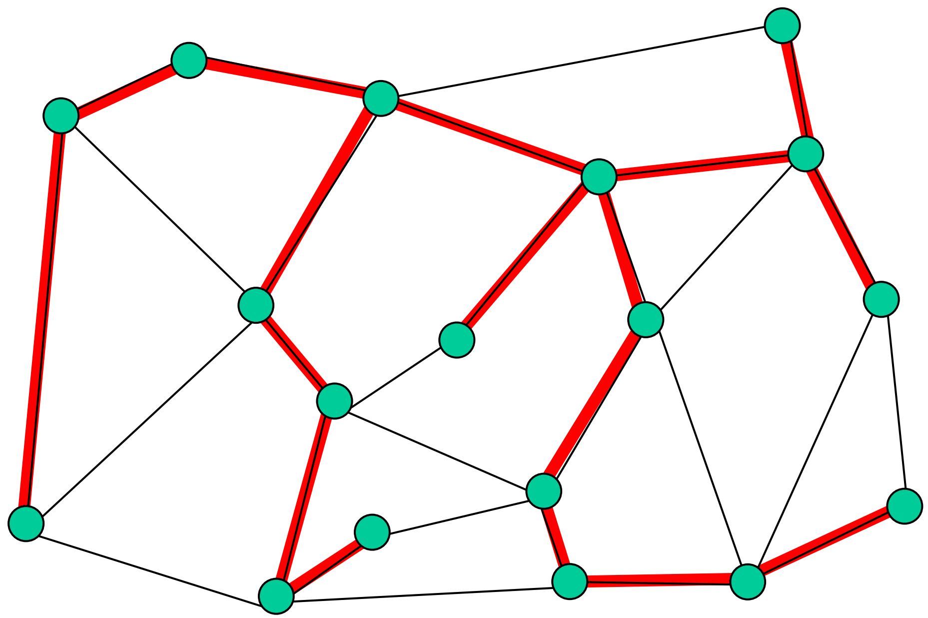
Un terremoto destruye las carreteras de una comarca. ¿Cuáles se deben reparar para conseguir rápidamente que todos los pueblos sigan conectados?



Grafo conexo
Sin ciclos

Árboles

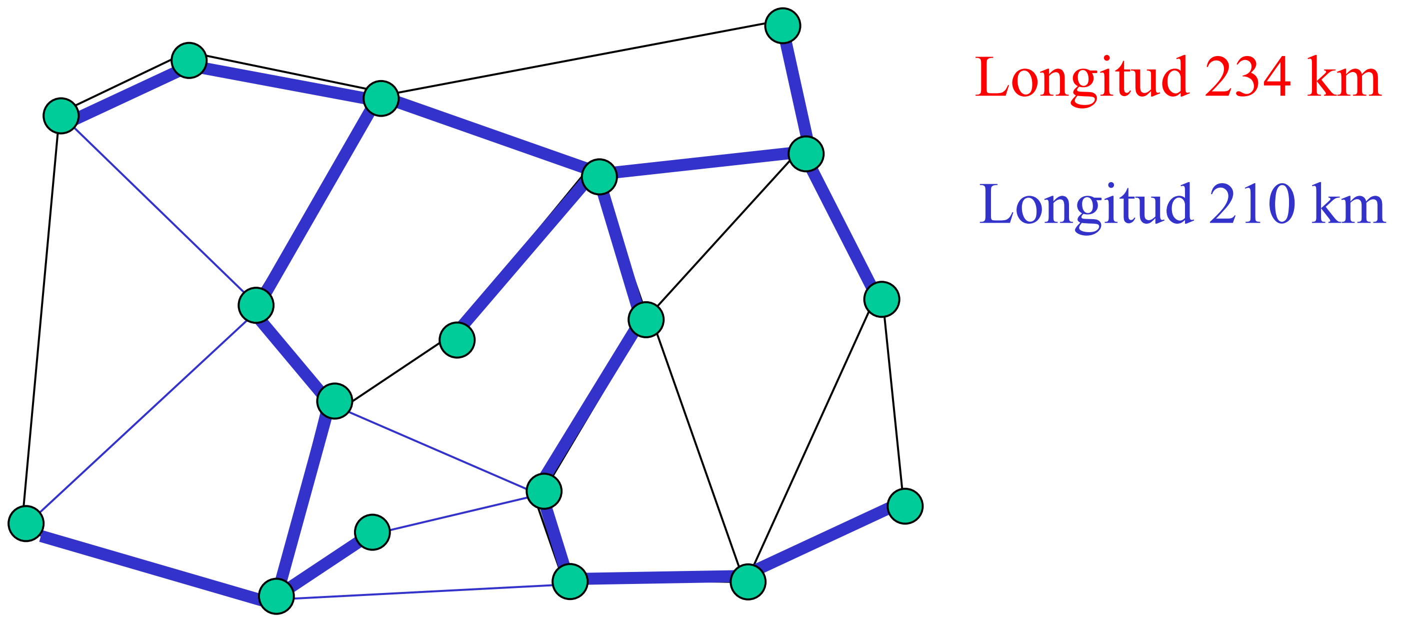
Un terremoto destruye las carreteras de una comarca. ¿Cuáles se deben reparar para conseguir rápidamente que todos los pueblos sigan conectados?



Longitud 234 km

Árboles

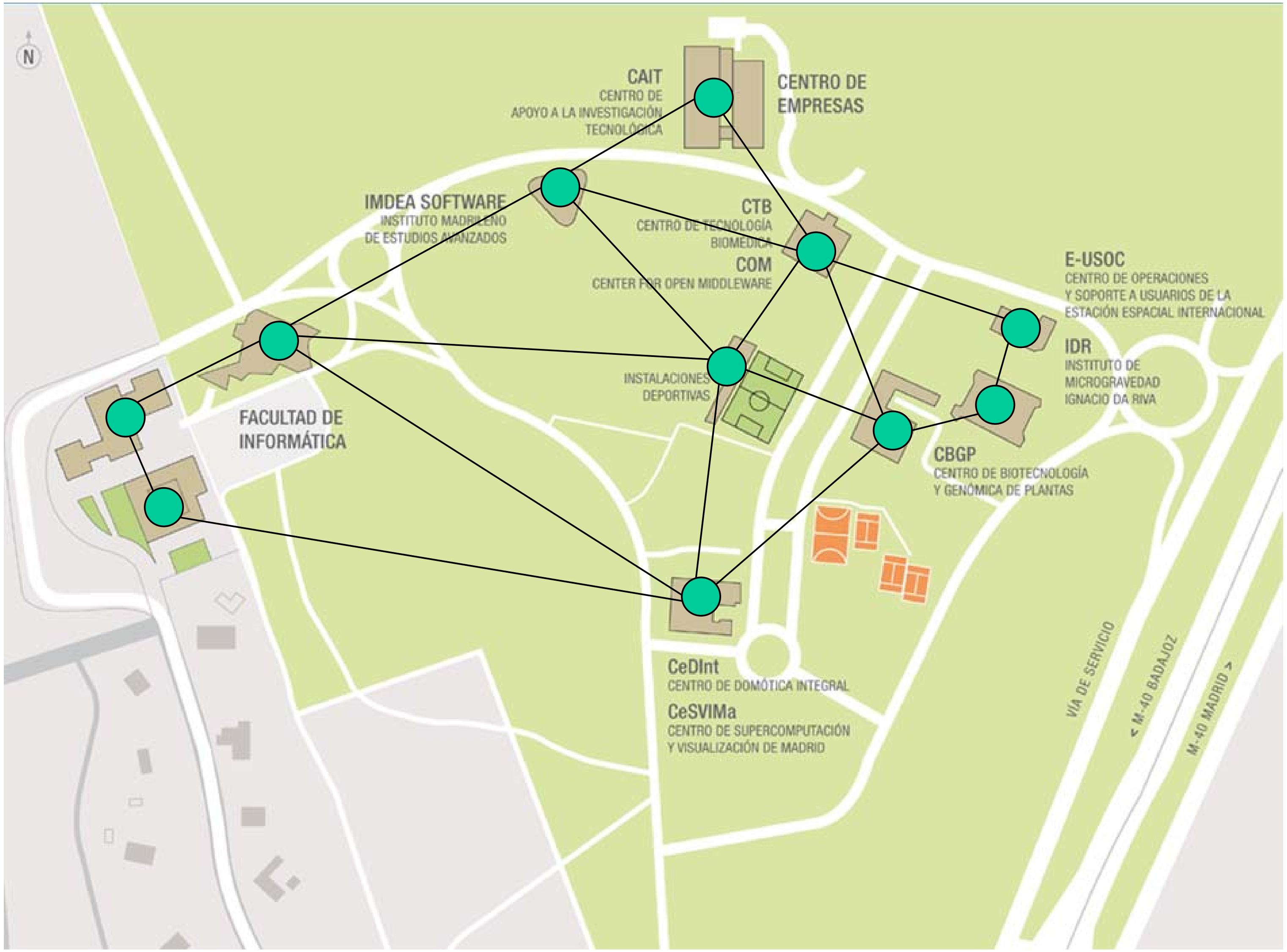
Un terremoto destruye las carreteras de una comarca. ¿Cuáles se deben reparar para conseguir rápidamente que todos los pueblos sigan conectados?



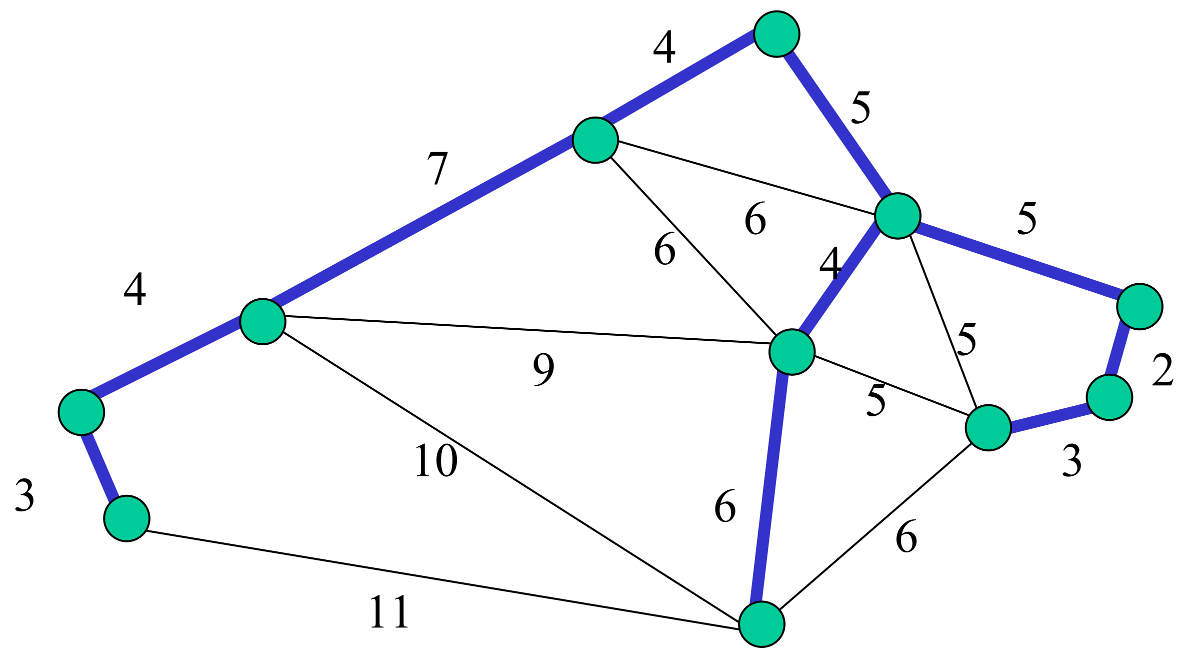
Árbol generador de longitud total mínima

Electrificación de Moravia (Borůvka, 1926)

Red de fibra óptica (Campus de Montegancedo)

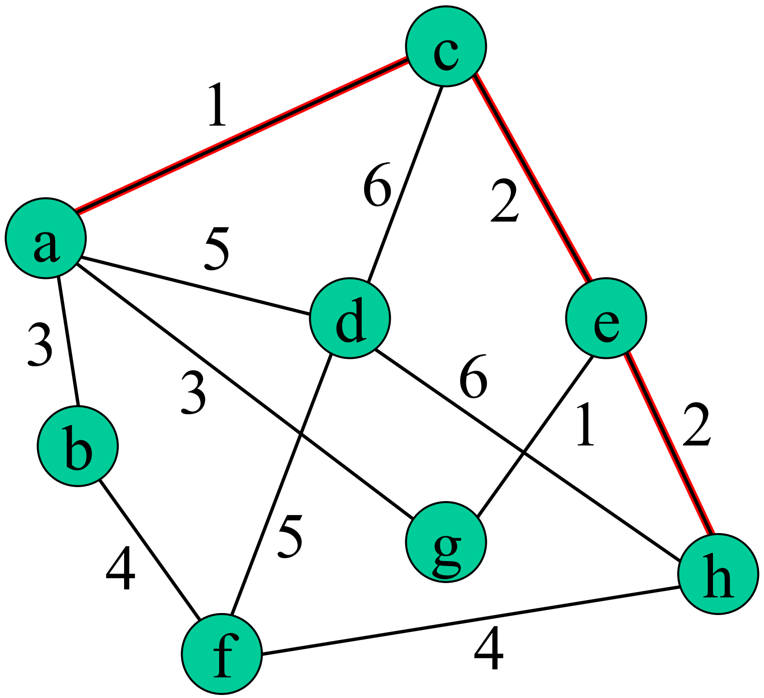


Red de fibra óptica (Campus de Montegancedo)



Grafos ponderados

$(G,w) \qquad w : A \longrightarrow R^{+}$



$w(G) = 42$

$\boxed{d(a,h) = 5}$

Peso de un subgrafo H=(V_1, A_1)

$$w(H) = \sum_{e \in A_1} w(e)$$

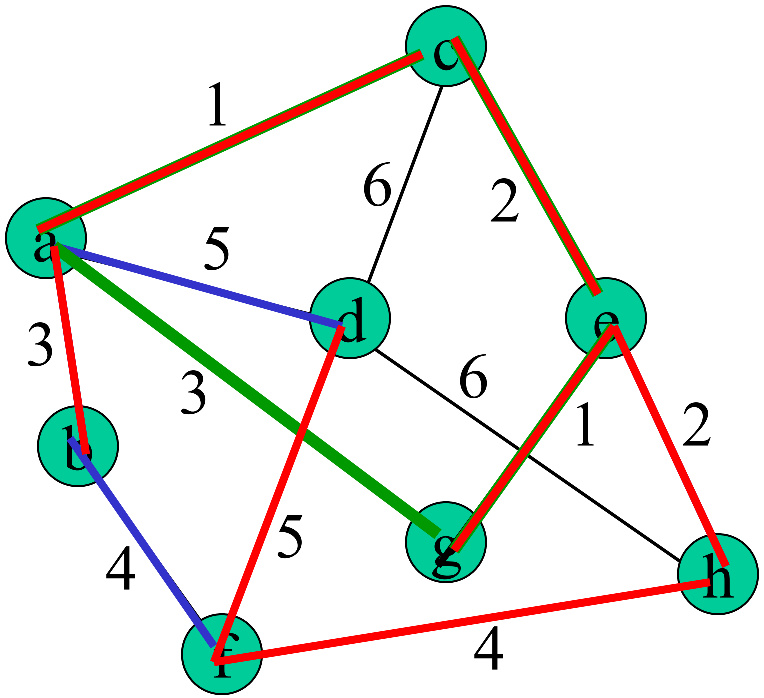
Distancia en un grafo ponderado

$$\boxed{d: V \times V \longrightarrow R^{+}}$$

d(u,v)=min{w(C) | C camino de u a v}

Árboles generadores mínimos

Un árbol generador mínimo de G, MST(G), es un árbol generador con el menor peso posible.

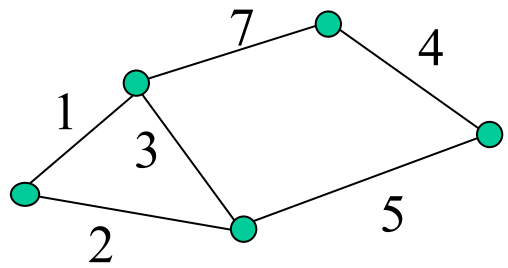


(G,w)

1. En general no es único

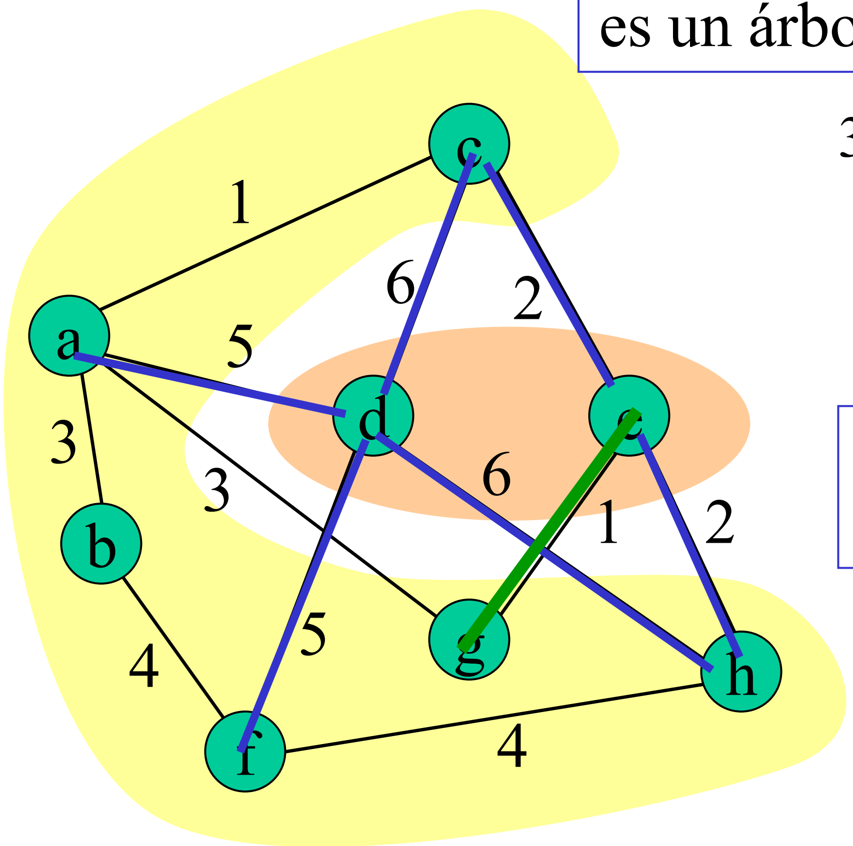
2. La arista de mayor peso de un **ciclo** no pertenece a MST(G).

¿Y la arista de menor peso de un ciclo?



Árboles generadores mínimos

Un árbol generador mínimo de G, MST(G), es un árbol generador con el menor peso posible.

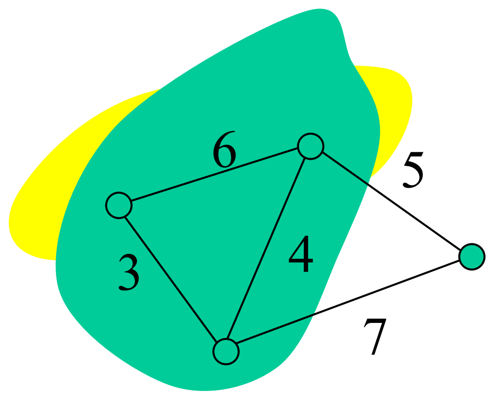


(G,w)

3. Un conjunto C de aristas de G es un **corte** si existe una partición (V_1, V_2) de V tal que C contiene todas las aristas con un extremo en V_1 y otro en V_2 .

4. La arista de menor peso de un corte pertenece a MST(G).

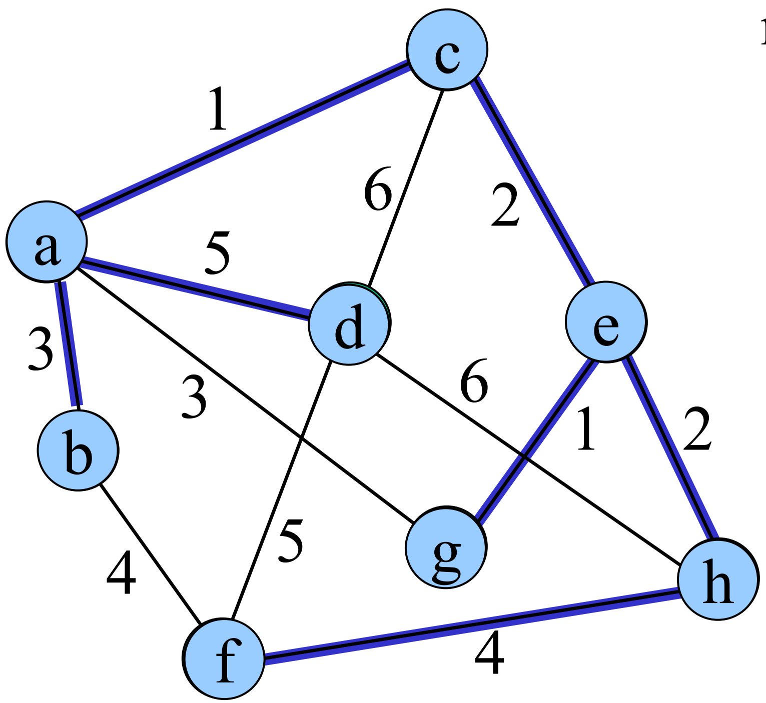
¿Y la arista de mayor peso de un corte?



Algoritmos para calcular MST

Algoritmo de Prim

Estrategia: Se parte de un vértice y se van alcanzando los demás, de uno en uno, del modo más económico posible



(G,w)

1. Seleccionar un vértice arbitrario u.
2. Hacer S={u} y T=∅ .
3. Mientras S ≠ V:
   a) Elegir v ∈ V–S vecino de un z ∈ S con w(vz) mínimo
   b) Insertar v en S.
   c) Insertar vz en T.

Algoritmo de Prim

Complejidad

Primera aproximación

En el paso 3, si S tiene k vértices hay n-k vértices en V-S. Por tanto, necesitamos hallar la arista de mínimo peso entre $k(n-k)$ aristas.

Como $k(n-k)<(n-1)^2$, el coste resulta $O(n^2)$.

Pero el bucle del paso 3 se repite $n-1$ veces luego la complejidad es $O(n^3)$

Algoritmo de Prim

Complejidad

Segunda aproximación

Una buena estructura de datos mejora la complejidad

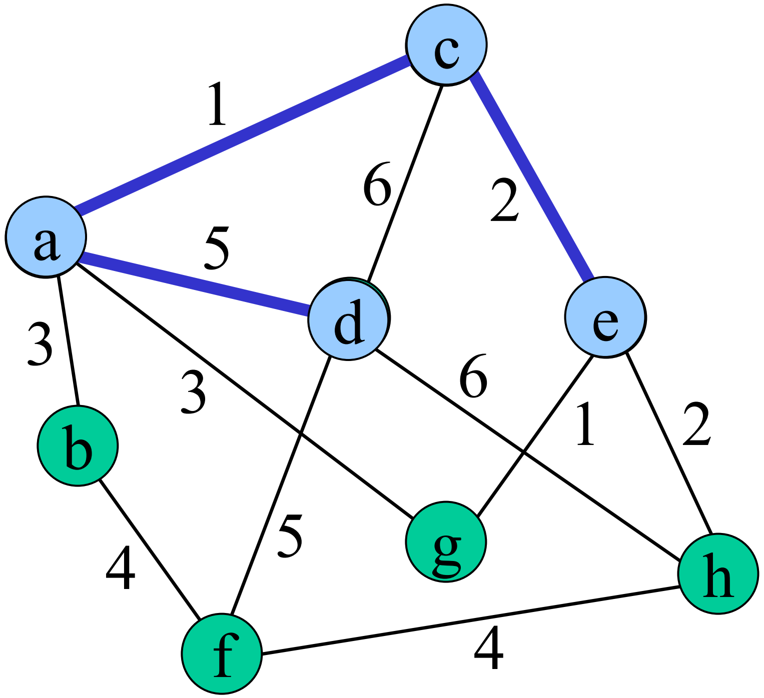
Consideremos las listas S y V – S .

A cada vértice z de V – S le etiquetamos inicialmente así:

$t(z)=w(uz)$ si existe la arista uz, $t(z)=\infty$ si no existe

Ahora en el paso 3, se elige el vértice z de V – S con etiqueta mínima, se halla $v \in S$ tal que $t(z)= w(vz)$, se añade z a S y se actualizan las etiquetas:

$$\text{Si } x \in Z \text{ hacemos } t(x):=\min\{t(x), w(vx)\}$$

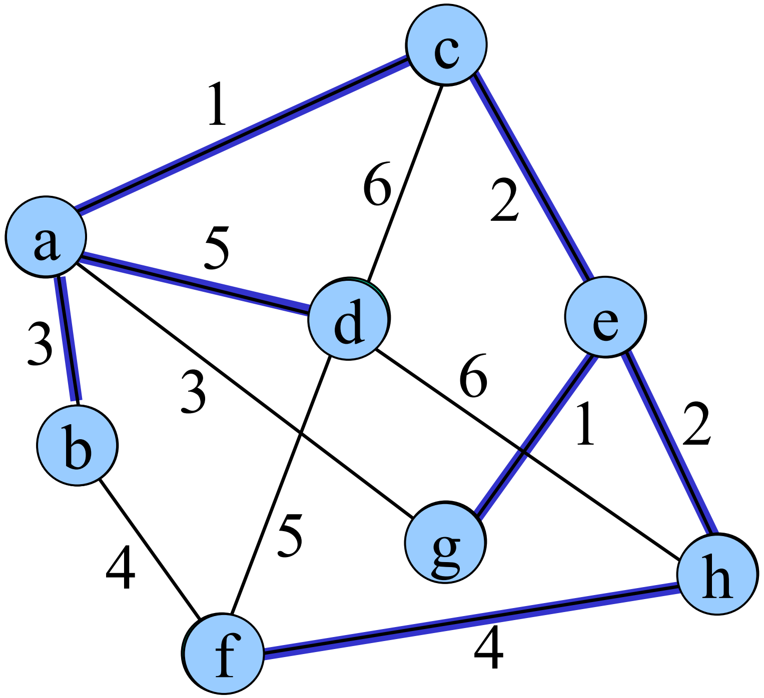


S:	d						
V-S:	a	b	c	e	f	g	h
	5	∞	6	∞	5	∞	6

S:	d	a					
V-S:		b	c	e	f	g	h
		3	1	∞	5	3	6

S:	d	a	c				
V-S:		b		e	f	g	h
		3		2	5	3	6

S:	d	a	c	e			
V-S:		b			f	g	h
		3			5	1	2



S:	**d**	**a**	**c**	**e**	**g**		
V-S:		**b**			**f**		**h**
		3			**4**		**2**

S:	**d**	**a**	**c**	**e**	**g**	**h**	
V-S:		**b**			**f**		
		3			**4**		

S:	**d**	**a**	**c**	**e**	**g**	**h**	**b**
V-S:					**f**		
					4		

Algoritmo de Prim

Complejidad

Ahora, en cada paso el mínimo se calcula entre n-k etiquetas.
Así el coste del cálculo de todos los mínimos es

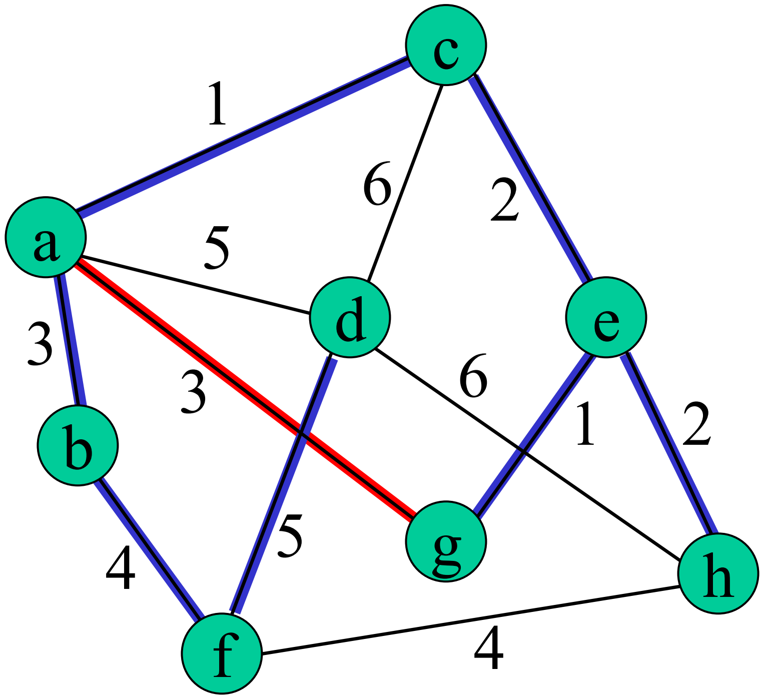
$$(n-1)+(n-2)+\ldots+1=\frac{n(n-1)}{2}\in O(n^2)$$

El nº total de actualizaciones de etiquetas es q, pues cada arista se considera sólo una vez.

Por tanto el coste total es $O(n^2) + O(q) = O(n^2)$

Algoritmos para calcular MST

Algoritmo de Kruskal



(G,w)

Estrategia: Se eligen aristas de la forma más barata, sin formar ciclos

Preproceso:
Ordenar las aristas de G por su peso

1. Hacer T=∅ .
2. Mientras T no sea generador:
   a) Elegir una arista uv tal que
      T ∪ uv no tenga ciclos
      y w(uv) sea mínimo.
   b) Insertar uv en T.

Algoritmo de Kruskal

Complejidad

Preproceso. Ordenar las aristas por su peso *O(q logq)*

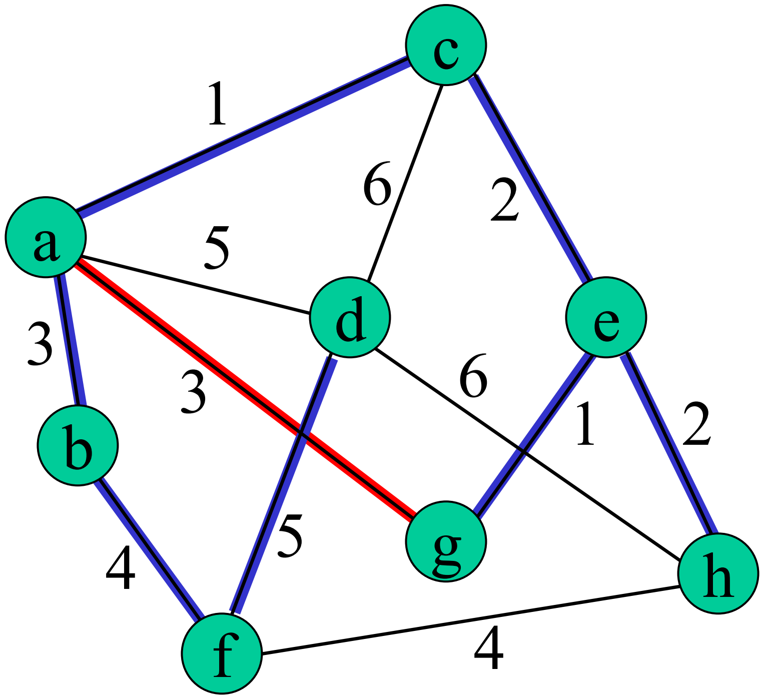
Paso 2. Hay que elegir la arista de menor peso que no forme ciclo con las aristas previamente elegidas.
Este paso se efectúa n-1 veces.

¿Cómo detectar que no se forme un ciclo?

Asignando etiquetas a los vértices de modo que vértices de la misma componente conexa reciban la misma etiqueta.

Algoritmo de Kruskal

Complejidad



(G,w)

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

Algoritmo de Kruskal

Complejidad

Preproceso. Ordenar las aristas por su peso *$O(q \log q)$*

Paso 2. Coste de las comparaciones:
Se hace una comparación por arista *$O(q)$*

Coste de las actualizaciones en las etiquetas:
En el peor de los casos se actualizan los n vértices en cada uno de los n – 1 pasos. En total *$O(n^2)$*

Coste total $O(q \log q) + O(q) + O(n^2) = O(n^2 + q \log q)$

Algoritmo de Kruskal

Demostración

Sea T el árbol (generador) que se obtiene en el algoritmo

Cada arista $a \notin T$ forma un ciclo $C_T(a)$ en $T+a$. Además cualquier arista a' de $C_T(a)$ tiene un peso $w(a') \leq w(a)$ (1)

¿T es de peso mínimo?

Sea S árbol de peso mínimo. Demostraremos que $w(S)=w(T)$

Inducción sobre q, número de aristas distintas en S y T

Si $q=0$, entonces $S=T$ y T es de peso mínimo.

Si $q>0$, supongamos que el resultado es cierto cuando el número de aristas distintas es menor que q

Elegimos una arista $e^* \in S - T$

$S - e^*$ se descompone en dos componentes conexas V' y V''

T es el árbol construido con Kruskal. Al llegar a considerar la arista e* NO se ha incluido en el árbol porque cerraba un ciclo en T, el ciclo $C_T(e^*)$. Por tanto, cualquier arista e de $C_T(e^*)$ tiene peso $w(e) \leq w(e^*)$ (Recordar (1))

Este ciclo $C_T(e^*)$ contiene una arista e' que NO está en S y que conecta V' con V''

Algoritmo de Kruskal

Demostración

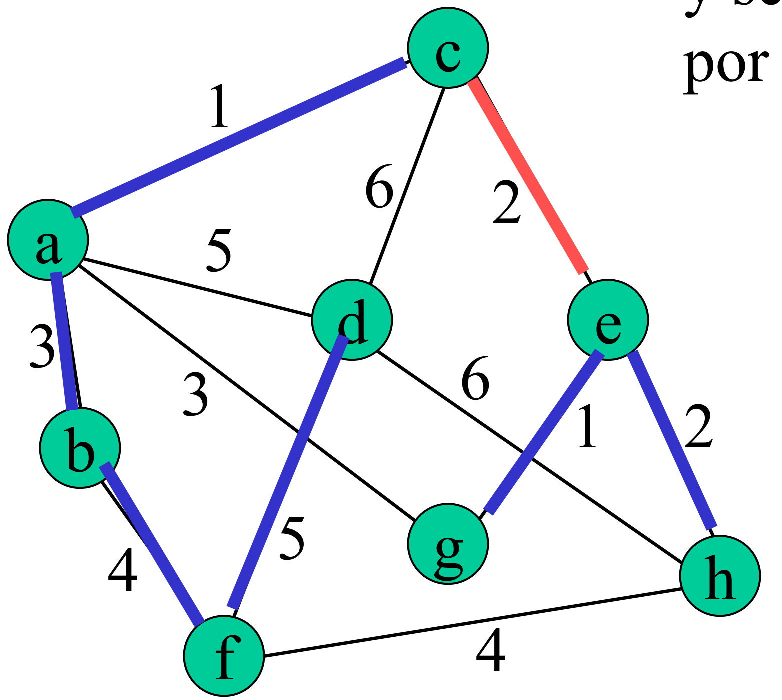
¿T es de peso mínimo?
Sea S árbol de peso mínimo. Demostraremos que $w(S)=w(T)$
Inducción sobre q, número de aristas distintas en S y T
Si $q=0$, entonces $S=T$ y T es de peso mínimo.
Si $q>0$, supongamos que el resultado es cierto cuando el número de aristas distintas es menor que q

El árbol $T' = S - e^* + e'$ es árbol generador.
Su peso es $w(T') = w(S) - w(e^*) + w(e') \leq w(S)$

Por ser S el de peso mínimo, $w(T') = w(S)$.
Así T' es otro árbol de peso mínimo con una arista más en común con T que S. Es decir, el número de aristas distintas de T' y T es $q - 1$
Por hipótesis de inducción $w(T')=w(T)$, luego $w(S)=w(T)$

Algoritmo de Borůvka

Estrategia: Se toma un bosque generador F y se une cada componente de F con otra por una arista de mínimo peso

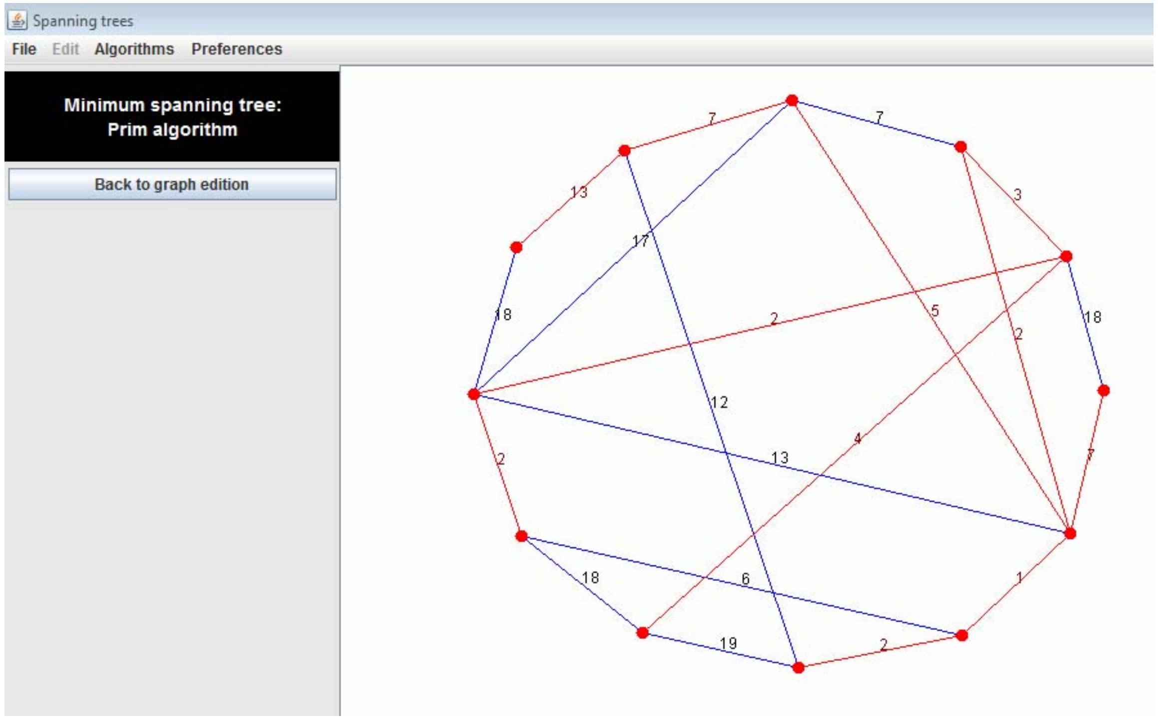


(G,w)

1. F bosque trivial con todos los vértices de G
2. Mientras F no sea conexo, repetir:

   Para cada componente F' de F se elige la arista de mínimo peso que la conecta con otra componente
   (S=conjunto de estas aristas)
   Añadir S al bosque F

Visualización de los algoritmos para MST y otros árboles óptimos
http://www.dma.fi.upm.es/personal/gregorio/grafos/web/visualizacion_arbol_optimo/



Y de otros árboles óptimos (MRCT, Steiner)
http://www.dma.fi.upm.es/personal/gregorio/grafos/web/arboles_optimos/Principal.html

Otros problemas sobre árboles generadores

Dado un grafo ponderado que representa las conexiones entre ciertos nodos, diseñar un árbol generador que:

- Minimice la máxima distancia entre nodos.

  ÁRBOL GENERADOR DE DIÁMETRO MÍNIMO

- Indique cuáles son las caminos mínimos desde un nodo a los restantes.

  ÁRBOL GENERADOR DE CAMINOS MÍNIMOS

- Minimice la diferencia entre pesos de sus aristas.

  ÁRBOL GENERADOR UNIFORME

- Conecte sólo una parte de los nodos (vértices “terminales”)

  ÁRBOL MÍNIMO DE STEINER

- Minimice la suma de los pesos de TODOS los caminos

  ÁRBOL de RUTAS CON MÍNIMO COSTE (MRCT)

………