

Análisis de Algoritmos y Estructura de Datos

Algoritmos sobre grafos II

Prof. Violeta Chang C

Semestre 2 – 2023



TDA grafo

- **Contenidos:**

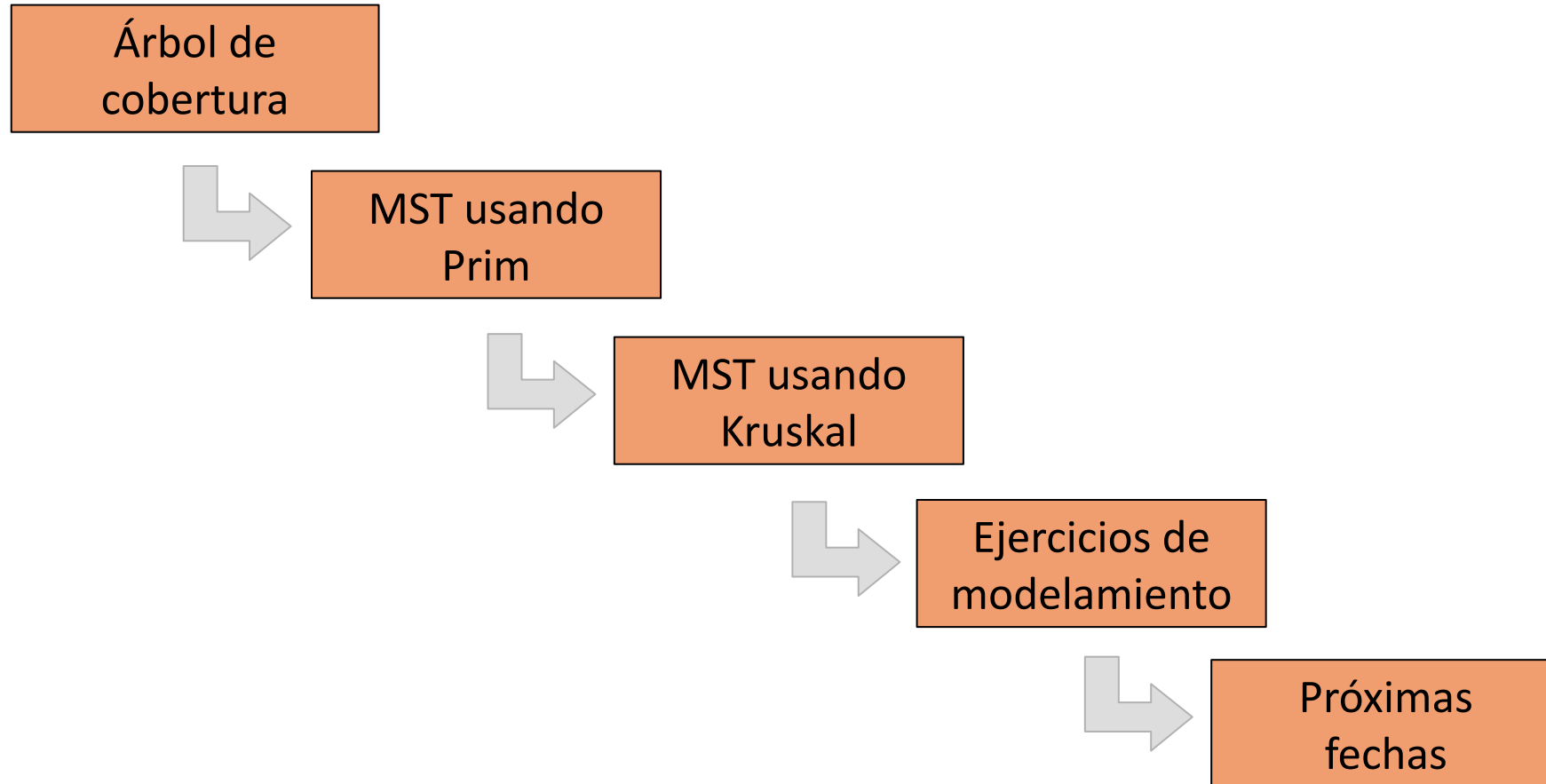
- Árbol mínimo de cobertura (MST)
- Algoritmo de MST Prim
- Algoritmo de MST Kruskal

- **Objetivos:**

- Comprender concepto de árbol de cobertura y árbol mínimo de cobertura
- Comprender funcionamiento de algoritmos Prim y Kruskal para obtener MST y aplicarlos para resolver problemas específicos



Ruta de la sesión

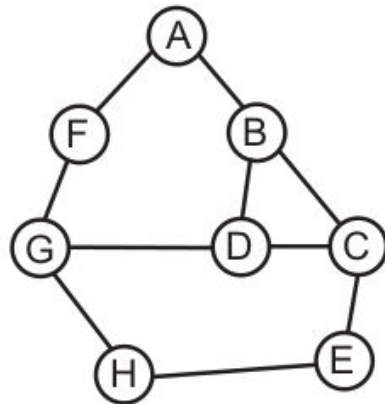




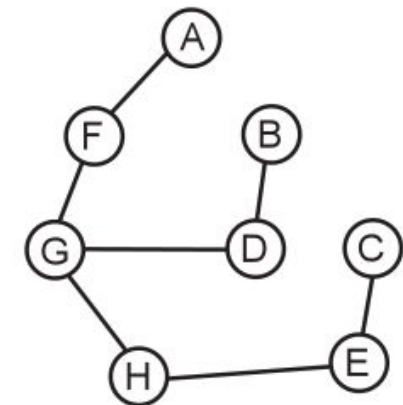
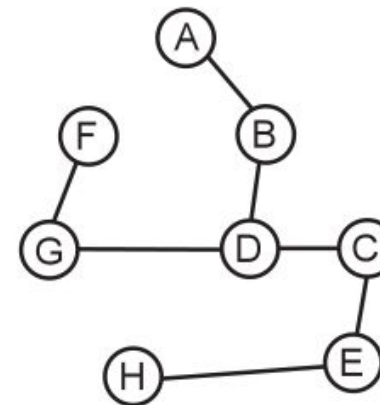
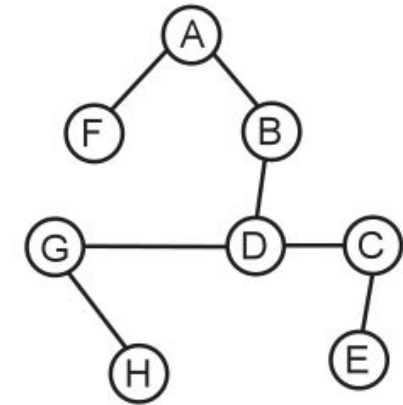
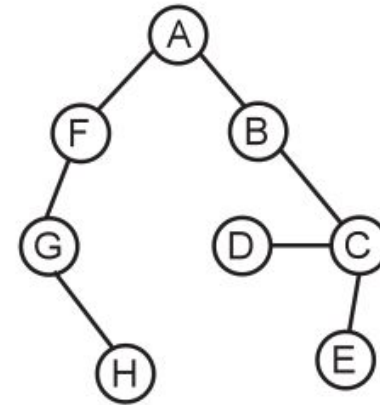
Árbol de cobertura de grafo

Un **árbol de cobertura** de un grafo G es un **subgrafo acíclico conexo** de G que tiene:

- $|V|$ vértices (los mismos que G)
- $|V|-1$ aristas (menor cantidad posible)



Grafo G

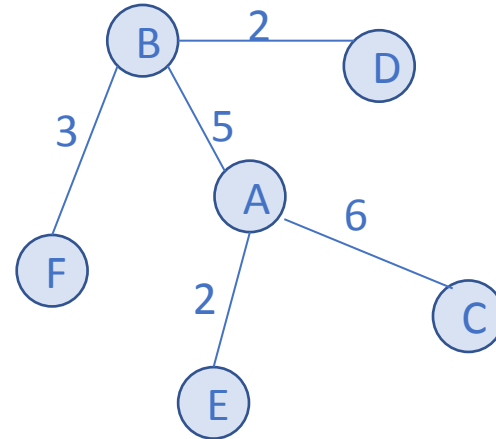
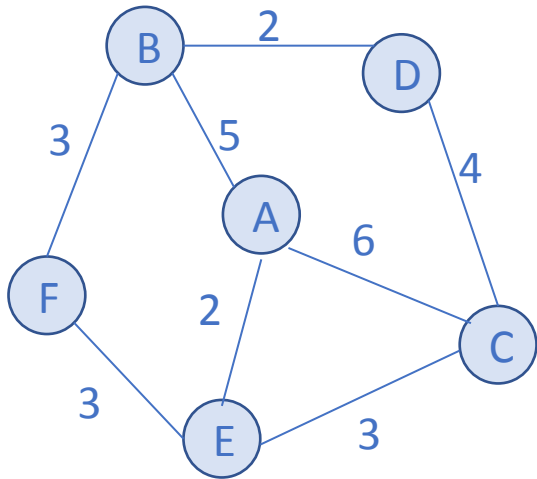


Posibles árboles de cobertura para el grafo G

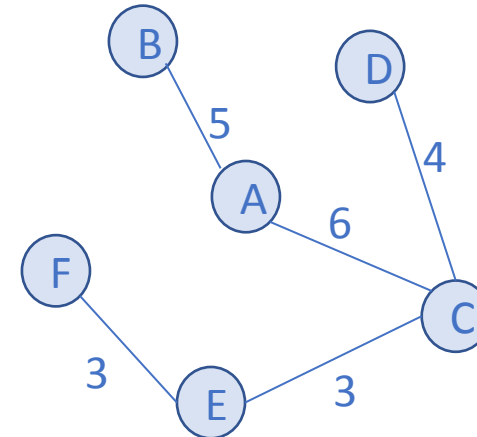


Árbol mínimo de cobertura (MST)

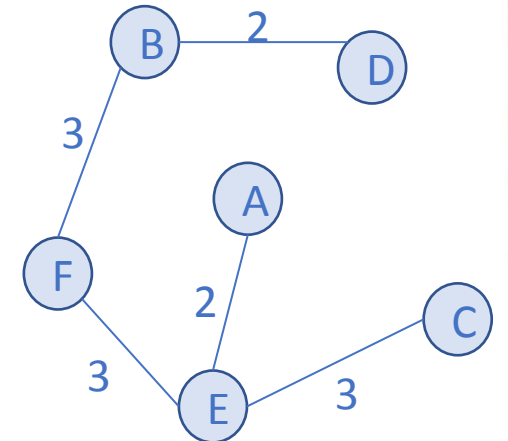
- Un grafo puede tener muchos posibles árboles de cobertura
- El **árbol de cobertura con menor peso** es el MST



Peso 18



Peso 25



Peso 13


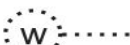



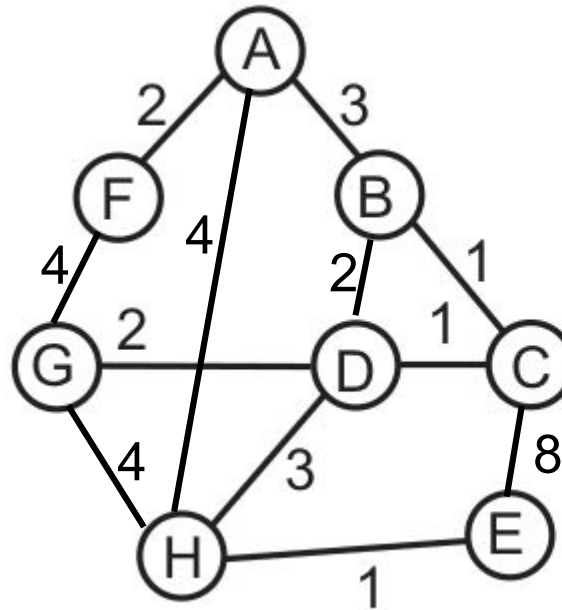
MST: Algoritmo de Prim

- Uno de los algoritmos más importantes y comunes para determinar cobertura mínima fue propuesto por **Prim** (1957)
- Idea general:
 - Se debe indicar **vértice** s para iniciar
 - El vértice se une con los **adyacentes más cercanos** examinando las aristas que poseen el **menor costo**: se selecciona la arista que tenga el menor peso
 - Se continúa hasta revisar todos los vértices del grafo
 - El MST que se obtiene **no debe tener ciclos**




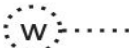

MST: Algoritmo de Prim

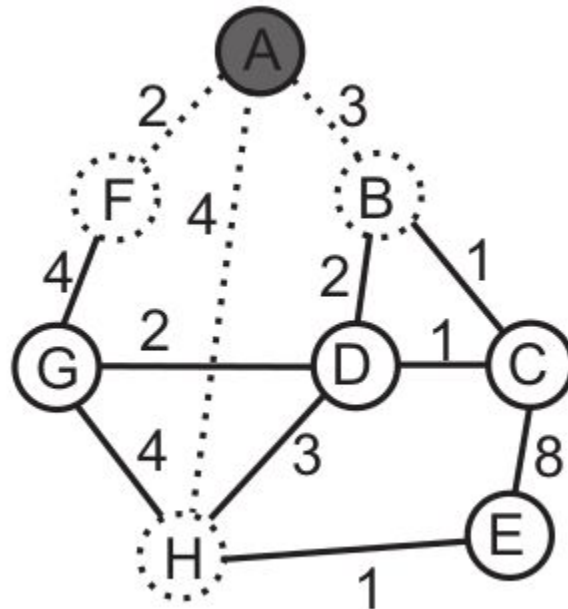
-  Vértice/arista que es parte de MST
-  Vértice/arista en la colección de aristas candidatas
-  Vértice/arista en el grafo que no ha sido revisada





MST: Algoritmo de Prim

-  Vértice/arista que es parte de MST
-  Vértice/arista en la colección de aristas candidatas
-  Vértice/arista en el grafo que no ha sido revisada

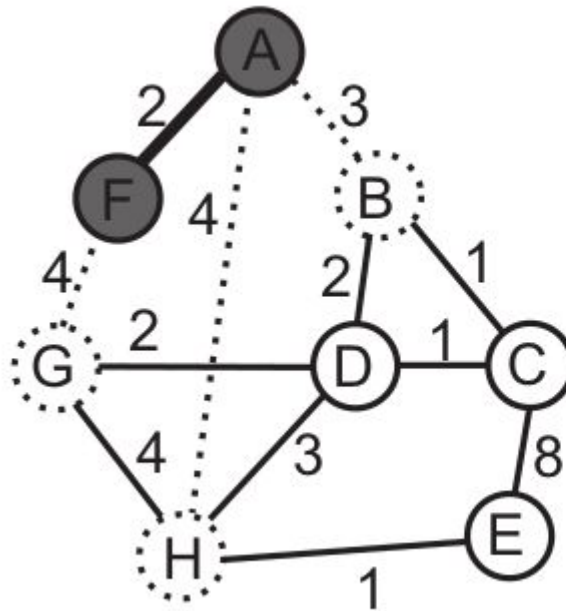
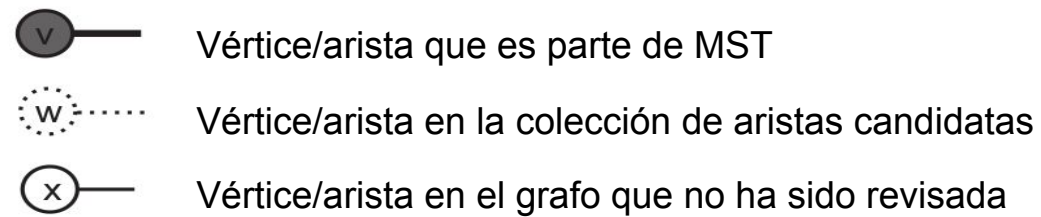


MST: A

Candidate Edges:
(A, F, 2), (A, H, 4),
(A, B, 3)



MST: Algoritmo de Prim

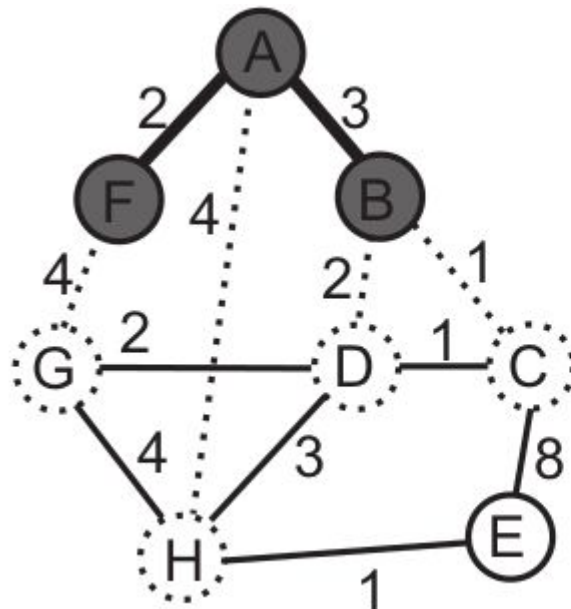
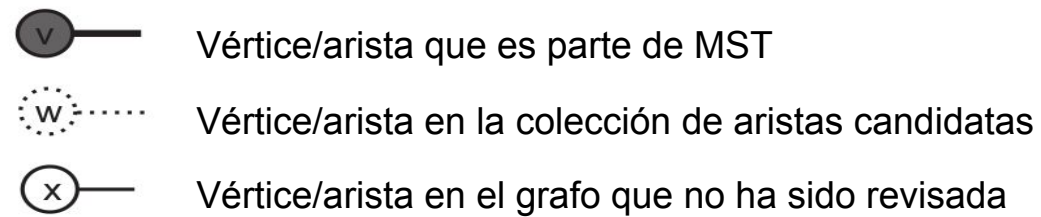


MST: A, F

Candidate Edges:
(A, B, 3), (A, H, 4),
(F, G, 4)



MST: Algoritmo de Prim

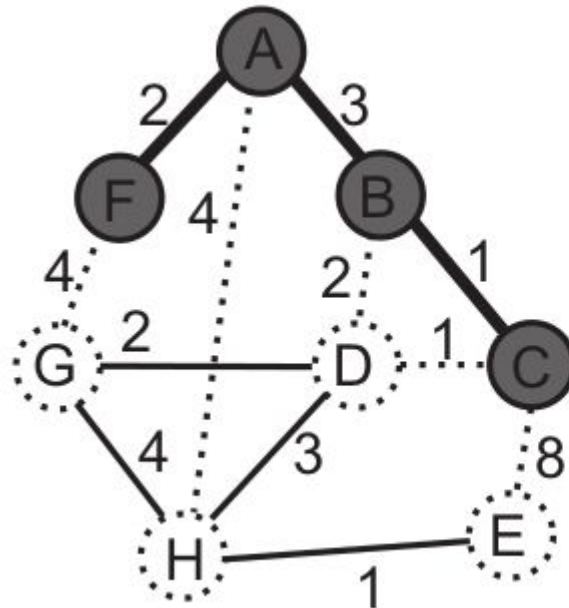
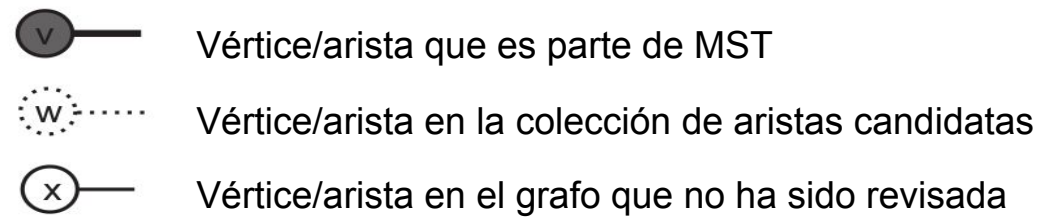


MST: A, F, B

Candidate Edges:
(B, C, 1), (A, H, 4),
(F, G, 4), (B, D, 2)



MST: Algoritmo de Prim

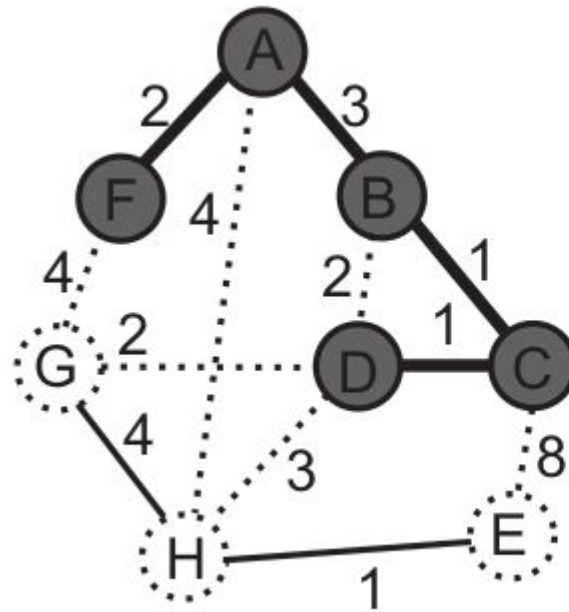
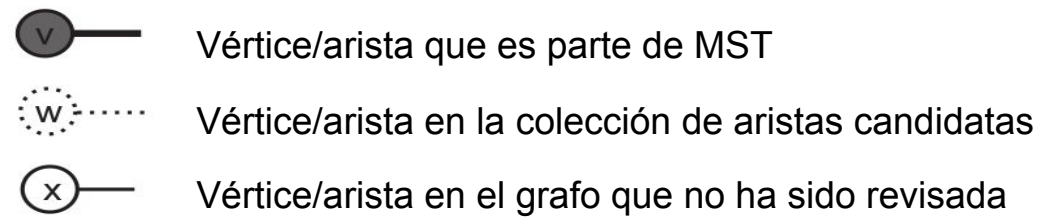


MST: A, F, B, C

Candidate Edges:
(C, D, 1), (A, H, 4),
(F, G, 4), (B, D, 2),
(C, E, 8)



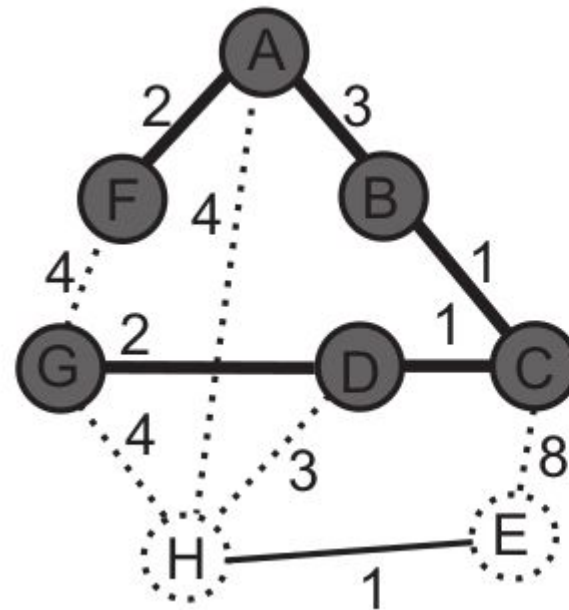
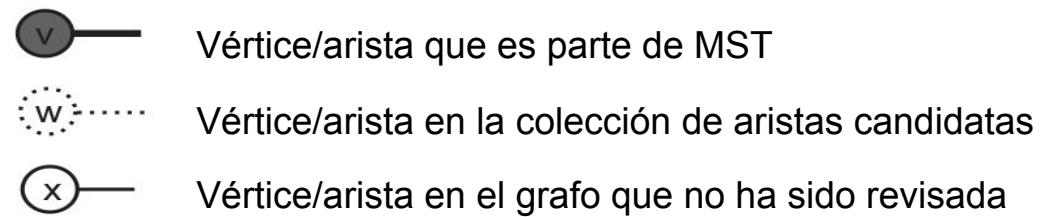
MST: Algoritmo de Prim



MST: A, F, B, C, D

Candidate Edges:
(B, D, 2), (A, H, 4)
(F, G, 4), (C, E, 8),
(D, G, 2), **(A, H, 4)**

MST: Algoritmo de Prim

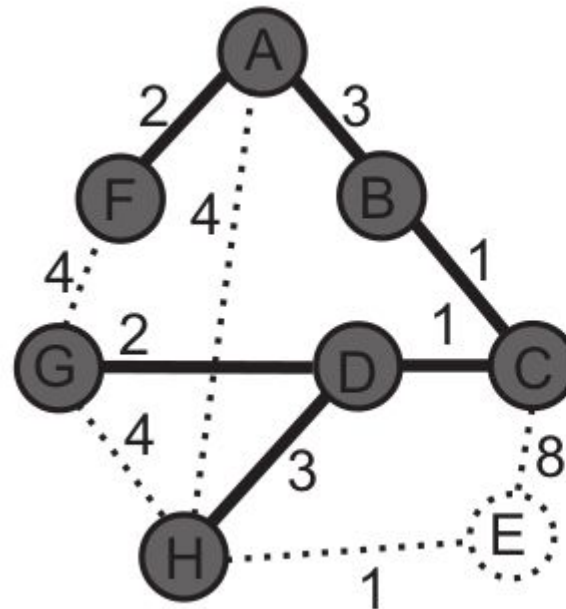
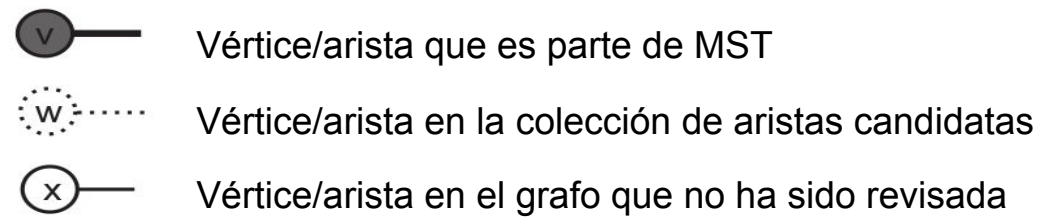


MST: A, F, B, C, D,
G

Candidate Edges:
(D, H, 3), (A, H, 4)
 (F, G, 4), (C, E, 8),
 (G, H, 4)



MST: Algoritmo de Prim

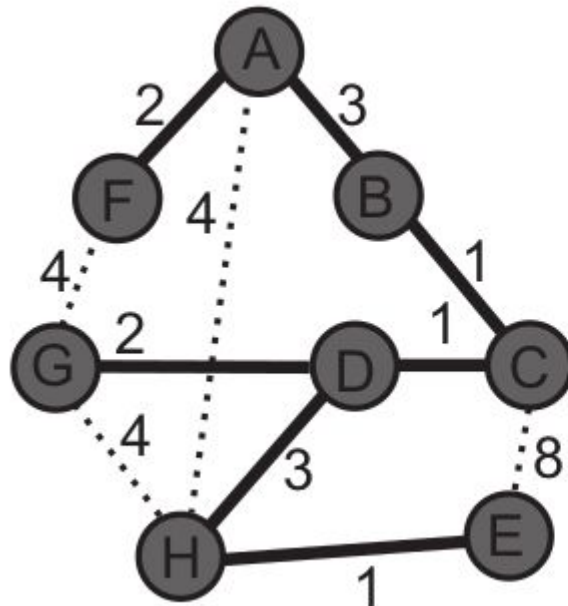
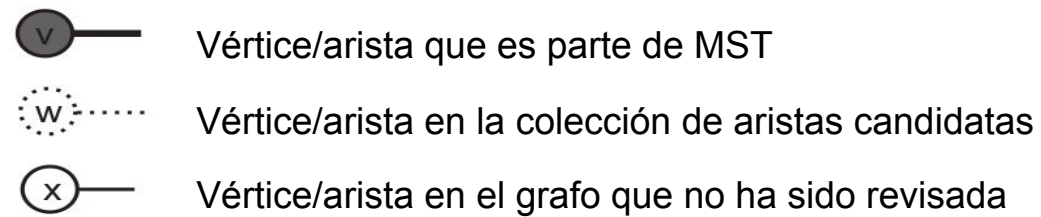


MST: A, F, B, C, D,
G, H

Candidate Edges:
(H, E, 1), (A, H, 4)
(F, G, 4), (C, E, 8),
(G, H, 4)



MST: Algoritmo de Prim

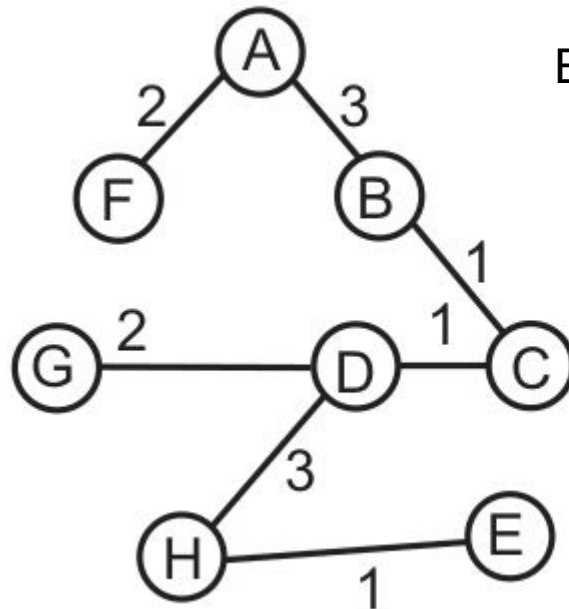
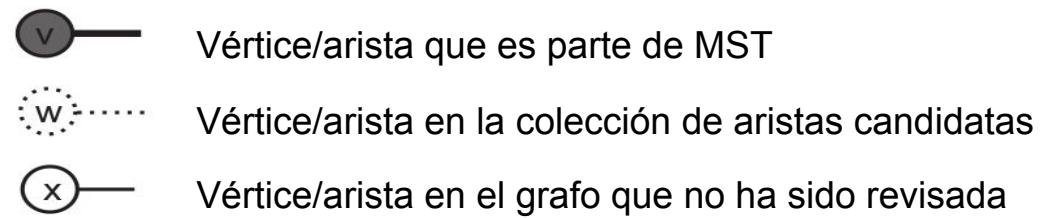


MST: A, F, B, C, D,
G, H, E

Candidate Edges:
(A, H, 4), (F, G, 4)
(C, E, 8), (G, H, 4)



MST: Algoritmo de Prim



El MST con el vértice A como su raíz



MST: Algoritmo de Prim

```
Prim (Grafo G, vertice s): LSE
  MST←crearListaVacía()
  n←calcularLargoArreglo(G→V)
  candidatos←crearColaPrioridadesVacía(n*n)
  cuenta←1
  u←s
  marcarVisitado(u)
  insertarNodoFinal(MST,u)
  mientras cuenta<n hacer
    adyacentes←ObtenerAdyacentes(G,u)
    mientras adyacentes<>NULO hacer
      v←adyacente→dato
      elemento←(u,v,G→W(u,v))
      encolar(candidatos,elemento)
      adyacentes←adyacentes→puntero
    elegido←frente(candidatos)
    mientras buscarDato(MST,elegido→fin) hacer
      descolar(candidatos)
      elegido←frente(candidatos)
    si elegido<>NULO entonces
      marcarVisitado(elegido→fin)
      insertarNodoFinal(MST,elegido)
      descolar(candidatos)
      cuenta←cuenta+1
      u←elegido→fin
  devolver (MST)
```

candidatos/MST

inicio	fin	peso
--------	-----	------

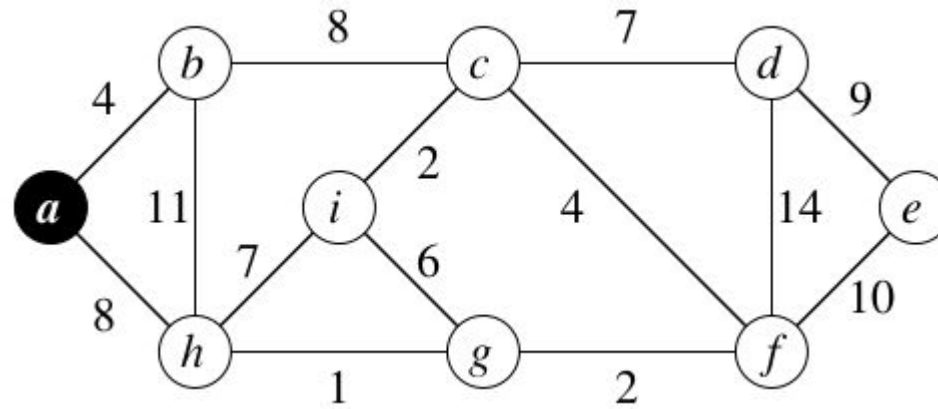


MST: Algoritmo de Kruskal

- Uno de los algoritmos más importantes y comunes para determinar cobertura mínima fue propuesto por **Kruskal** (1956)
- Idea general:
 - No se debe indicar vértice s para iniciar
 - **Se escoge la arista de menor peso** (en el caso de haber varias de igual peso se escoge cualquiera)
 - Se repite elección de arista de menor peso, hasta revisar todas las aristas del grafo, y se considera como parte de MST siempre que no genere ciclo con las aristas ya seleccionadas
 - El MST **no debe tener ciclos**

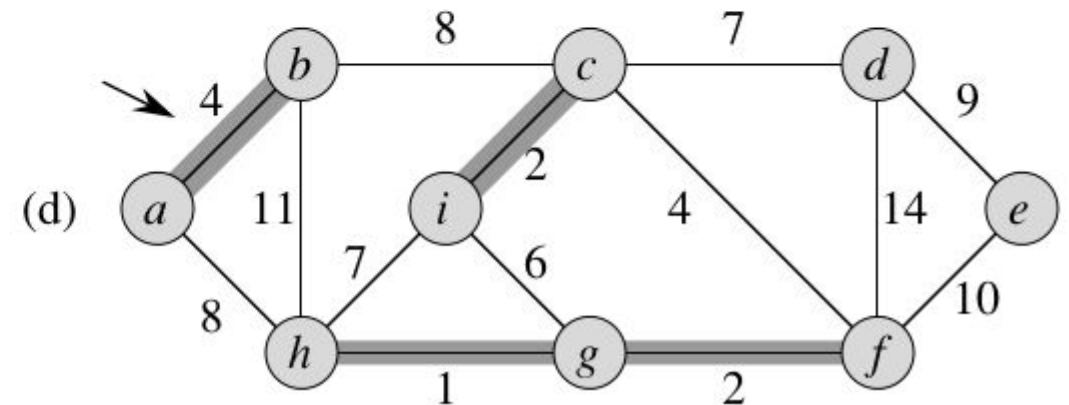
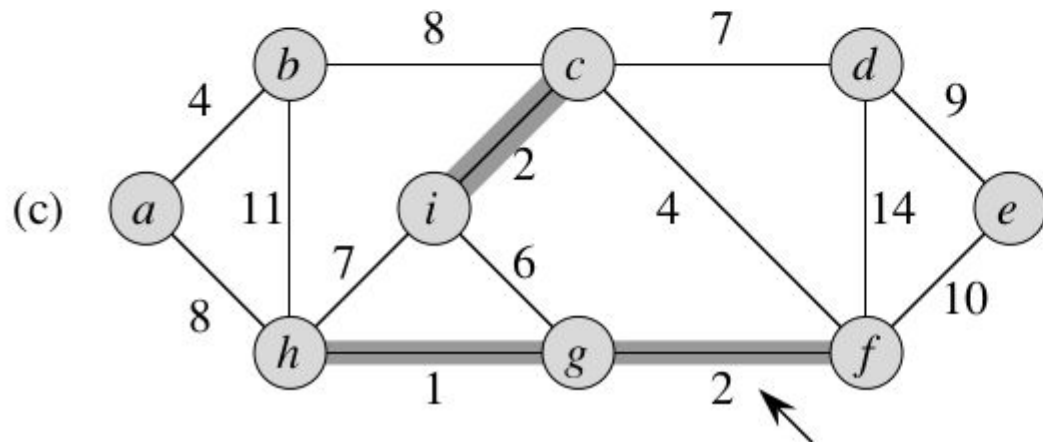
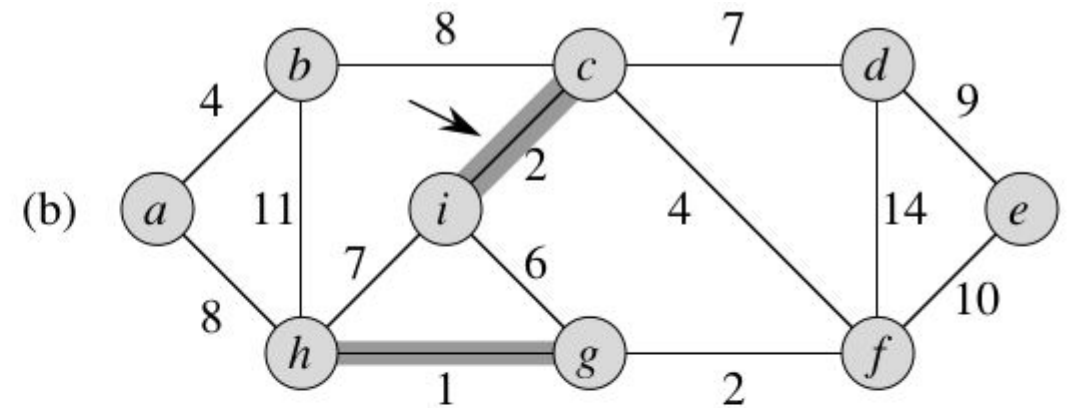
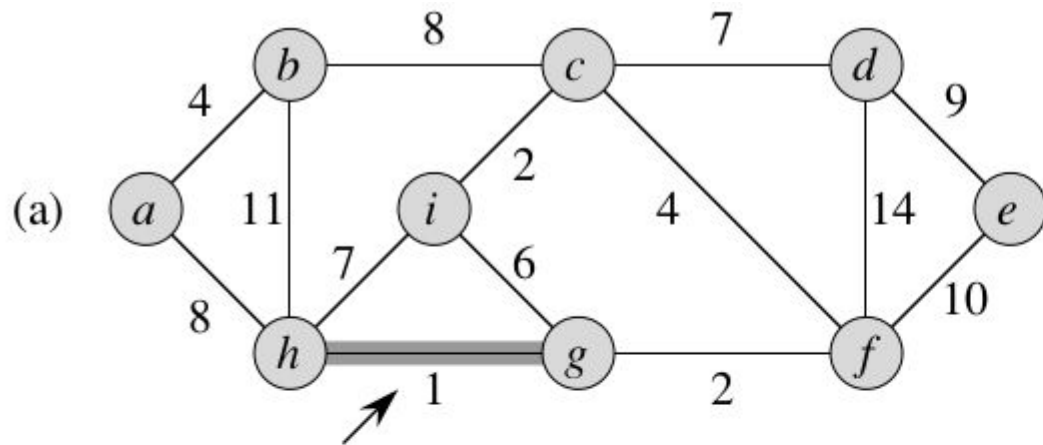


MST: Algoritmo de Kruskal



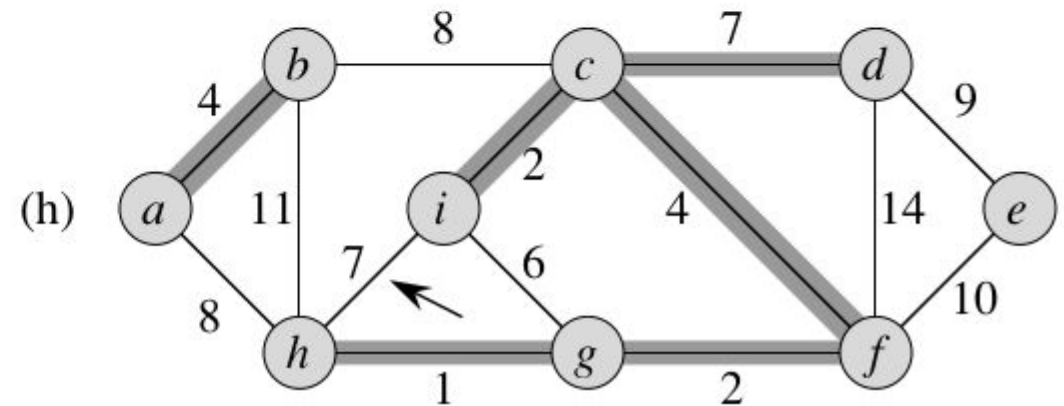
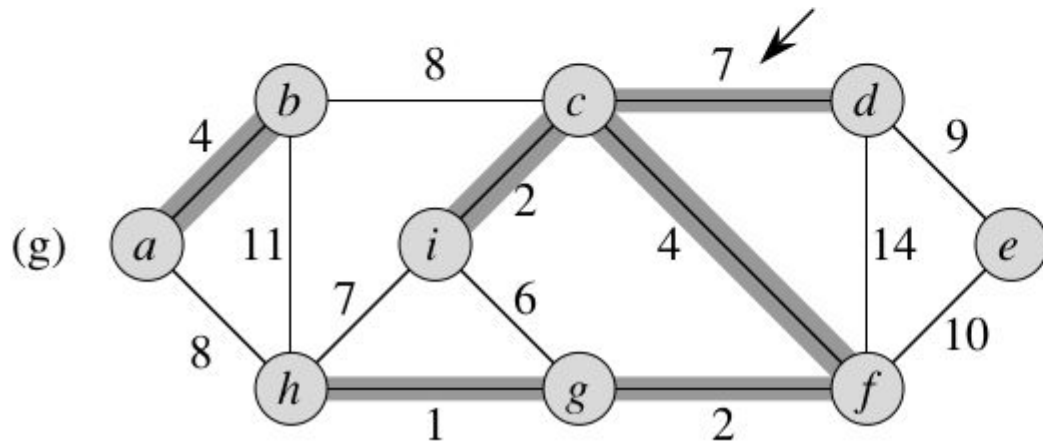
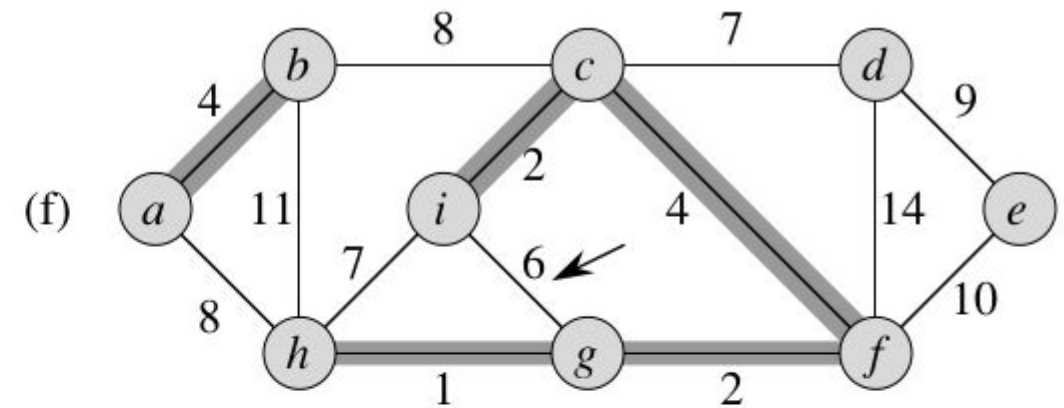
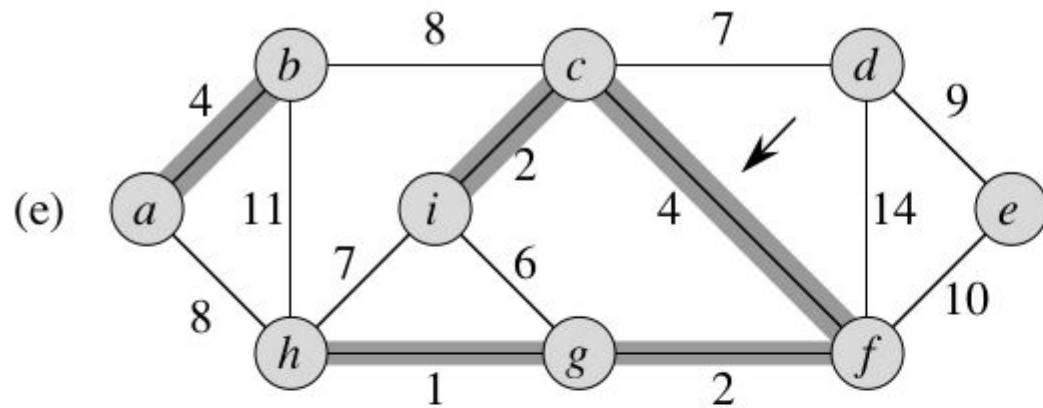


MST: Algoritmo de Kruskal



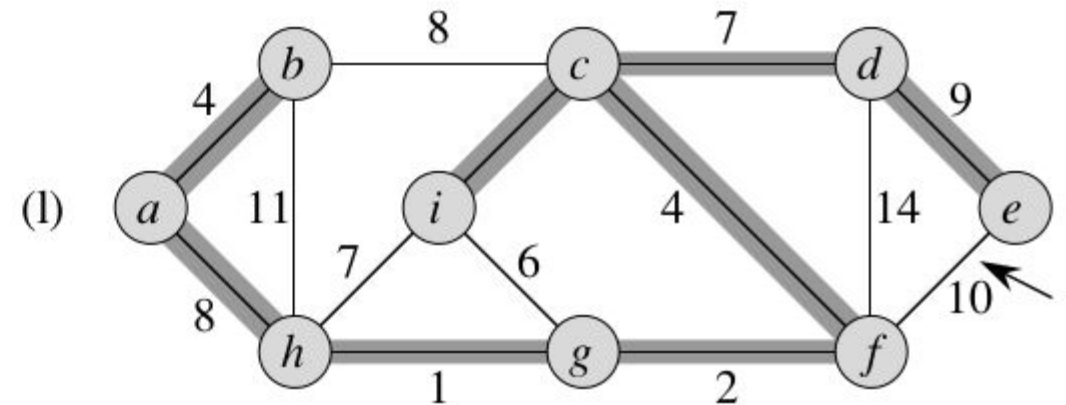
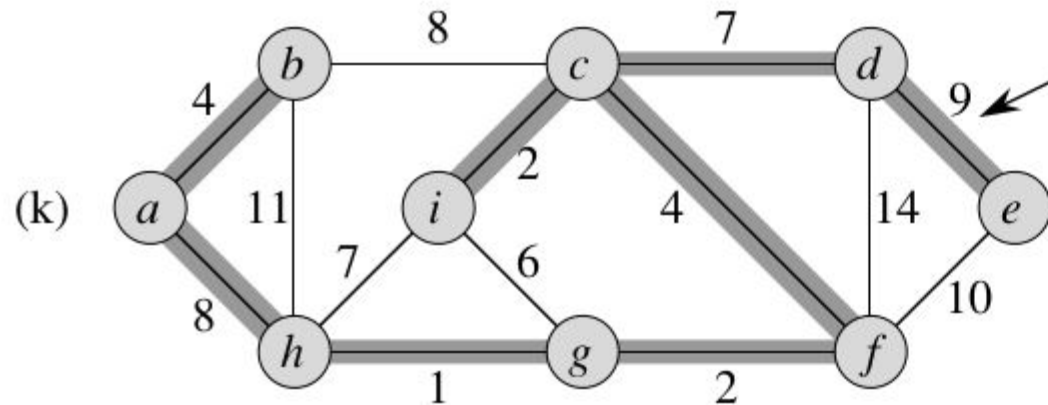
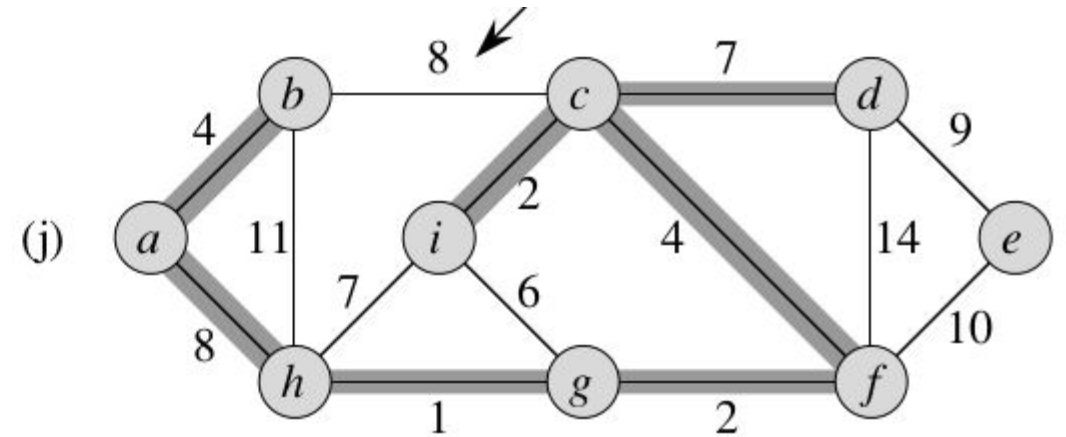
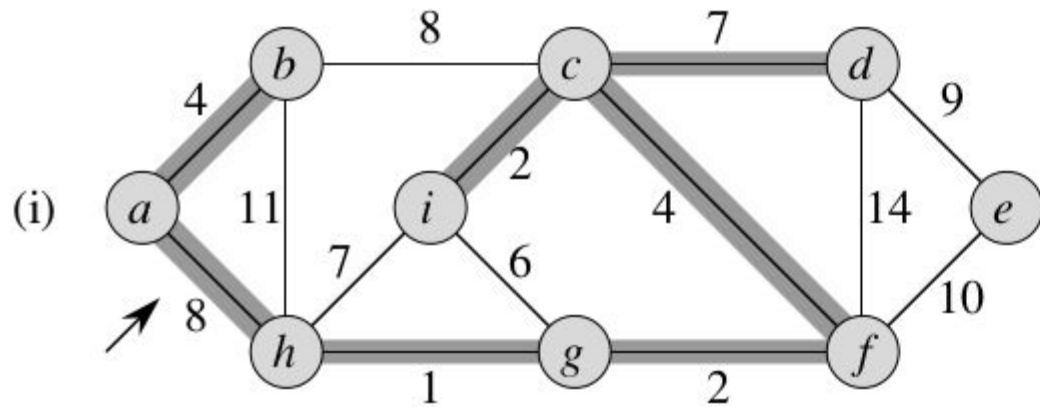


MST: Algoritmo de Kruskal



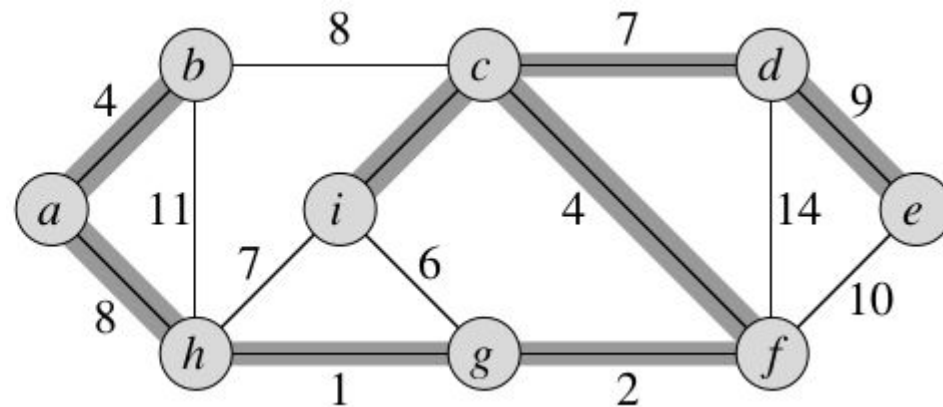
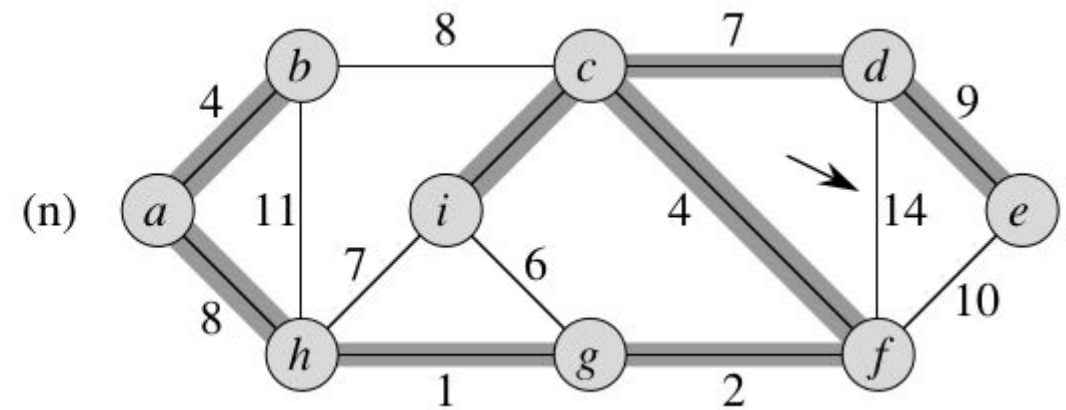
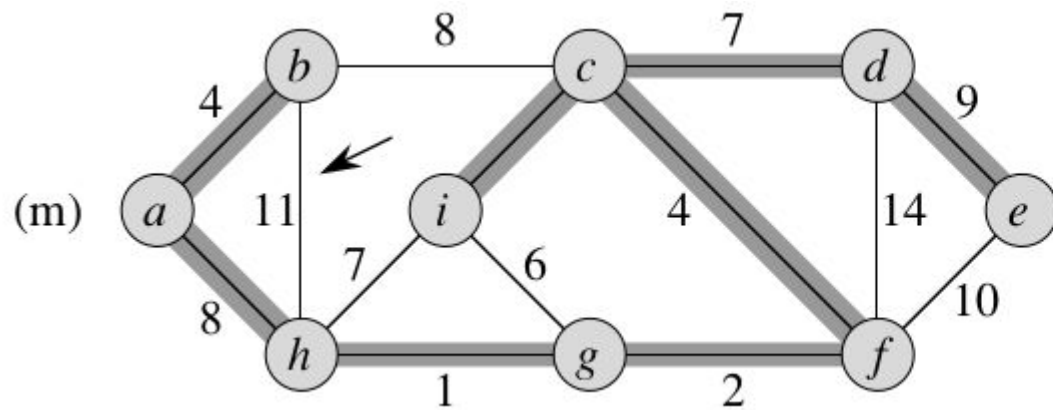


MST: Algoritmo de Kruskal





MST: Algoritmo de Kruskal





MST: Algoritmo de Kruskal

```
Kruskal(grafo G): LSE
    MST←crearListaVacia()
    colaAristas←OrdenarAristas(G)
    inicio←1
    mientras no(esColaVacia(colaAristas)) hacer
        arista←frente(colaAristas)
        descolar(colaAristas)
        si inicio=1 entonces
            insertarNodoFinal(MST,arista)
            inicio←0
        sino
            si no(existeCiclo(MST,arista)) entonces
                insertarNodoFinal(MST,arista)
    devolver (MST)
```

MST

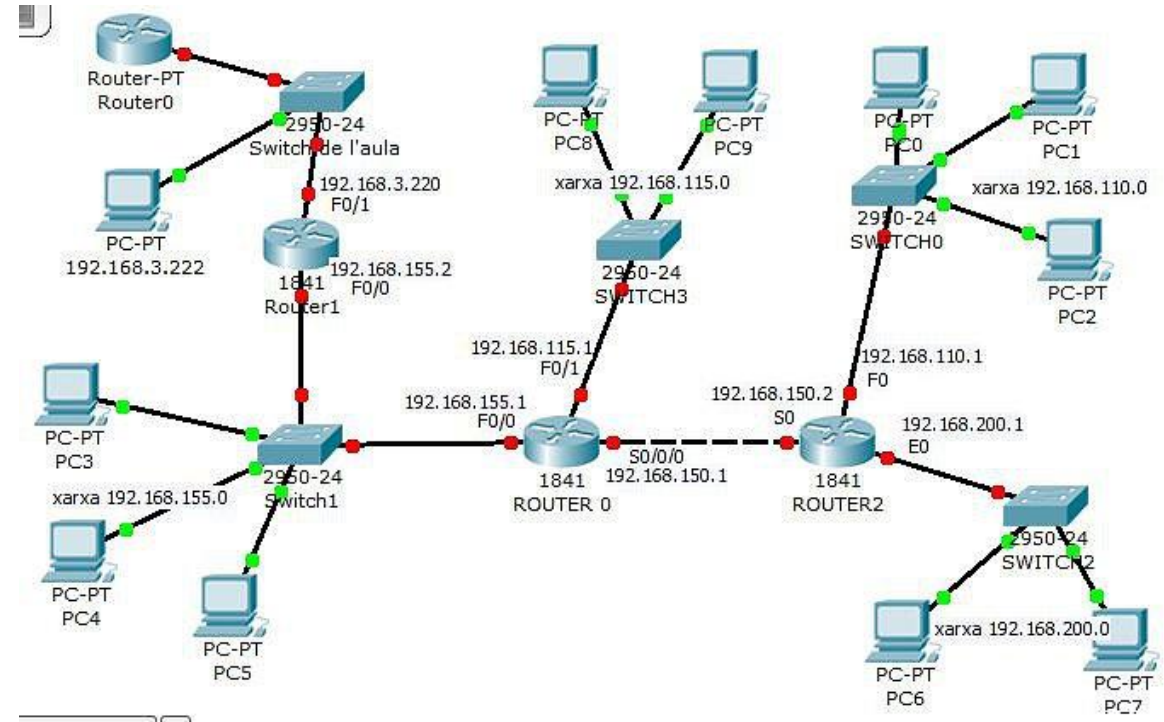
inicio	fin	peso
--------	-----	------



Ejemplos de uso de MST

En Internet cuando se quiere hacer un broadcast (transmisión a múltiples destinos) las redes de routers calculan el MST.

Cuando se requiere hacer el broadcast, cada router reenvía paquetes a los routers en el MST.





Proyecto de tren subterráneo

- La ciudad de La Serena está planificando el desarrollo de un nuevo sistema de tren subterráneo. El sistema debe unir de alguna forma 5 sectores de la ciudad (zona centro, norte, sur, este y oeste), más otros 3 puntos de interés: el sector Financiero, el centro comercial y la Universidad. Se realizaron distintos estudios de factibilidad técnica de unir distintos puntos, el cual se tradujo a costos y se presenta en la tabla inferior. Los patrocinadores de la iniciativa necesitan saber cuál sería el trazado que cumpla con los requisitos y que signifique la menor inversión, además del monto requerido.
- Modelar el problema, identificando: objetos, relaciones, gráfico, pregunta puntual sobre grafo, estrategia de solución
- Escribir algoritmo
- Solucionar el problema

	Zona Norte	Zona Sur	Zona Centro	Zona Este	Zona Oeste	Sector Financiero	Sector Comercial	Universidad
Z. Norte			34	55	33	30		50
Z. Sur			37	44	40		36	
Z. Centro	34	37			28	32	41	35
Z. Este	55	44					43	38
Z. Oeste	33	40	28					
S. Financiero	30		32					39
S. Comercial		36	41	43				45
Universidad	50		35	38		39	45	



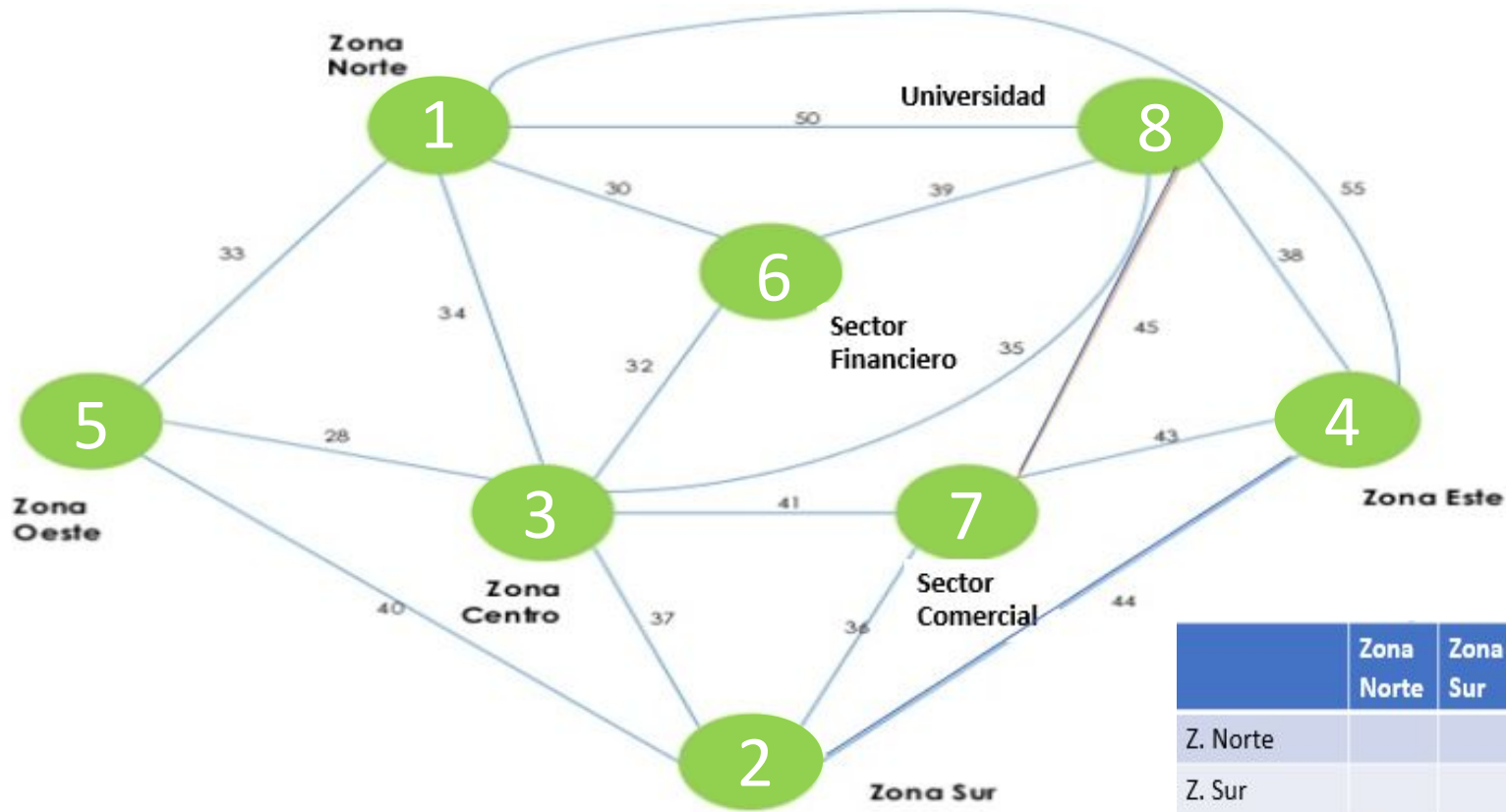
Proyecto de tren subterráneo

- La ciudad de La Serena está planificando el **desarrollo de un nuevo sistema de tren subterráneo**. El sistema debe unir de alguna forma **5 sectores** de la ciudad (**zona centro, norte, sur, este y oeste**), más otros **3 puntos de interés**: el **sector Financiero**, el **centro comercial** y la **Universidad**. Se realizaron distintos estudios de factibilidad técnica de **unir distintos puntos**, el cual se tradujo a costos y se presenta en la tabla inferior. Los patrocinadores de la iniciativa necesitan saber cuál sería el **trazado que cumpla con los requisitos y que signifique la menor inversión**, además del **monto requerido**.

	Zona Norte	Zona Sur	Zona Centro	Zona Este	Zona Oeste	Sector Financiero	Sector Comercial	Universidad
Z. Norte			34	55	33	30		50
Z. Sur			37	44	40		36	
Z. Centro	34	37			28	32	41	35
Z. Este	55	44					43	38
Z. Oeste	33	40	28					
S. Financiero	30		32					39
S. Comercial		36	41	43				45
Universidad	50		35	38		39	45	



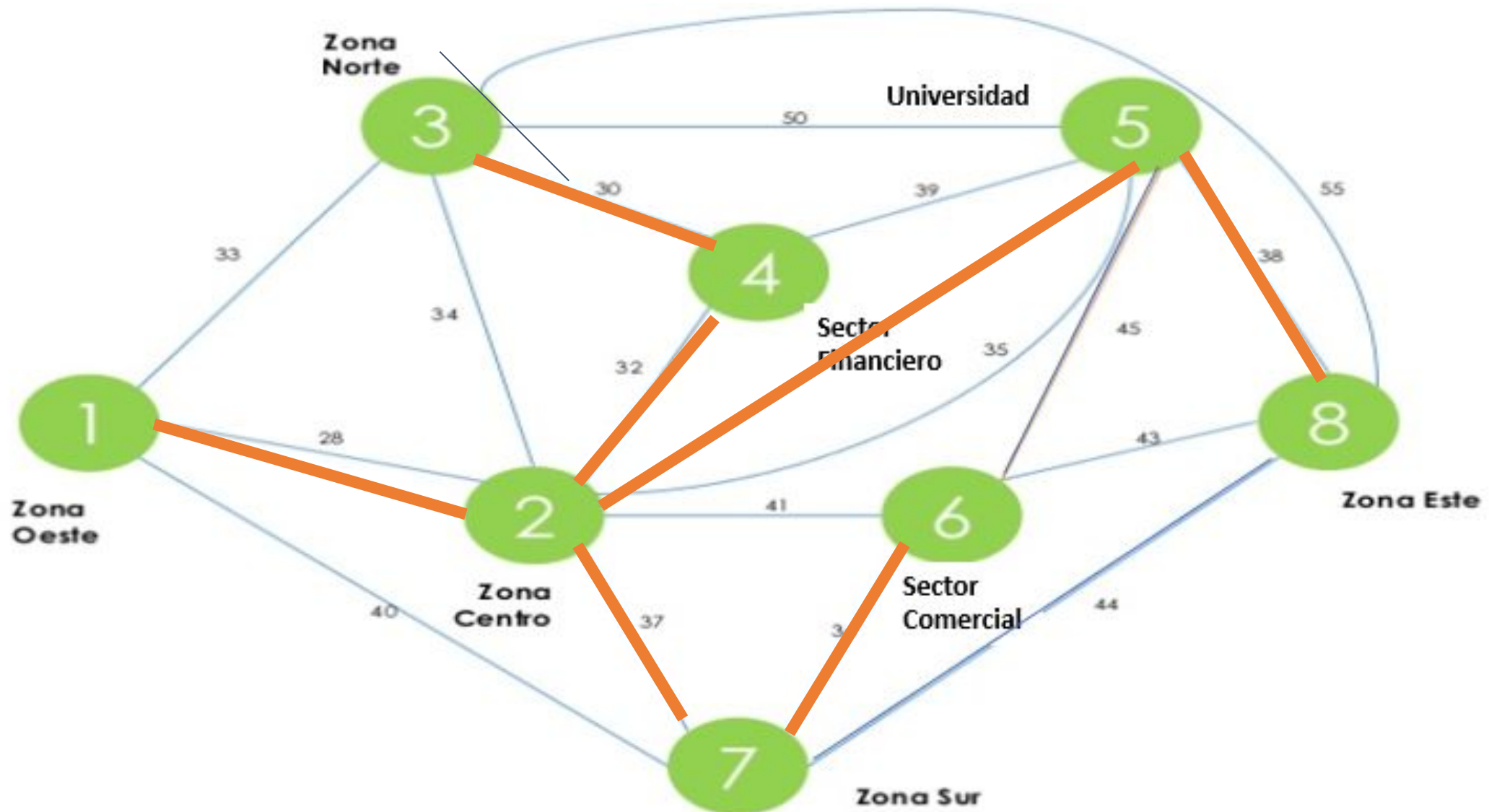
Proyecto de tren subterráneo



	Zona Norte	Zona Sur	Zona Centro	Zona Este	Zona Oeste	Sector Financiero	Sector Comercial	Universidad
Z. Norte			34	55	33	30		50
Z. Sur			37	44	40		36	
Z. Centro	34	37			28	32	41	35
Z. Este	55	44					43	38
Z. Oeste	33	40	28					
S. Financiero	30		32					39
S. Comercial		36	41	43				45
Universidad	50		35	38		39	45	



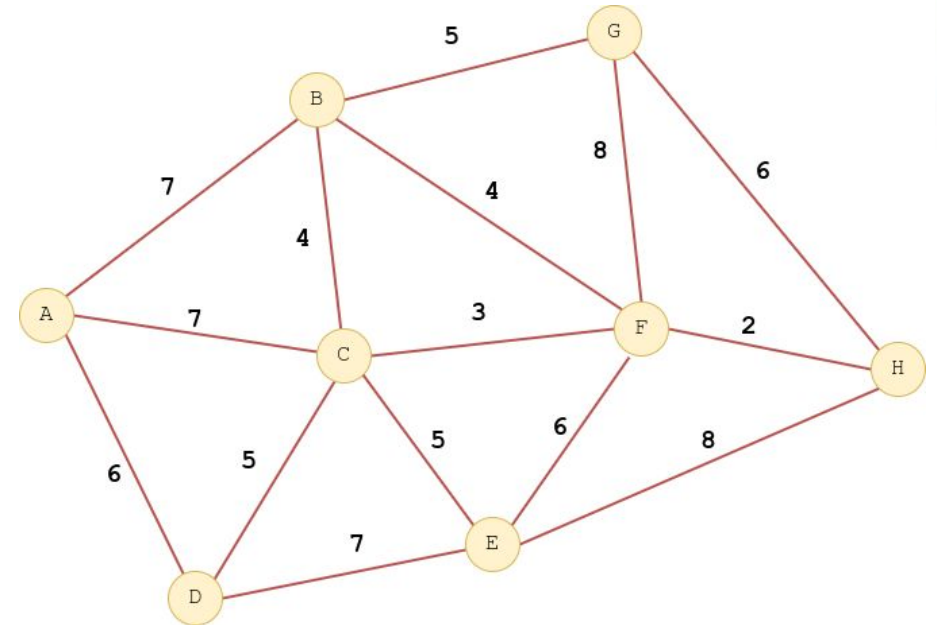
Proyecto de tren subterráneo





Cableado eléctrico

Se necesita implementar un sistema eléctrico para una ciudad, buscando mantener la conectividad entre centrales eléctricas ubicadas en diversas partes de la ciudad. El grafo mostrado considera un costo asociado a la instalación de conexión entre un par de centrales eléctricas en particular. ¿Cómo deberá ser la red resultante que asegure la conectividad pero tenga el menor costo de instalación? Escribir un algoritmo que permita conocer la configuración que se necesita para resolver el problema planteado. ¿Cuál es la complejidad del algoritmo propuesto?



Actividad de cierre



- Ir a [menti.com](https://www.menti.com) e ingresar código 7763 5129



Próximas fechas...

- Resumen de la semana:
 - Algoritmos de árboles de cobertura mínima (Prim y Kruskal)

~~cátedra~~ – ~~refuerzo~~ – laboratorio

U3 - S10

- Próxima semana:
 - TDA árbol
 - TDA árbol binario
 - TDA árbol binario de búsqueda

Noviembre 2023						
Domingo	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Receso						
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

Diciembre 2023						
Domingo	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30