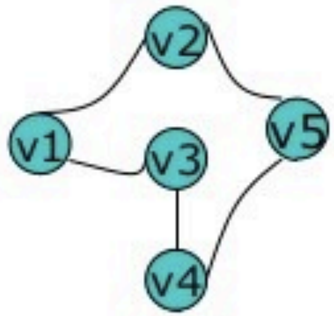
UNIVERSIDAD
DE SANTIAGO
DE CHILE

Optimización I

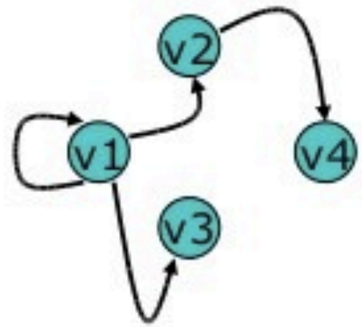
Luis Rojo-González
luis.rojo.g@usach.cl
Departamento de ingeniería industrial,
Universidad de Santiago, Chile

Ingeniería civil industrial

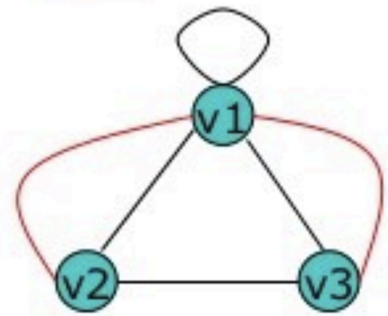
Grafos



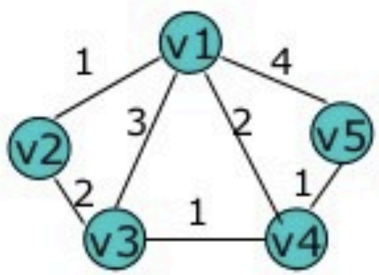
No Dirigido



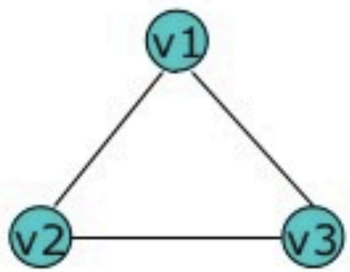
Dirigido
(Digrafo)



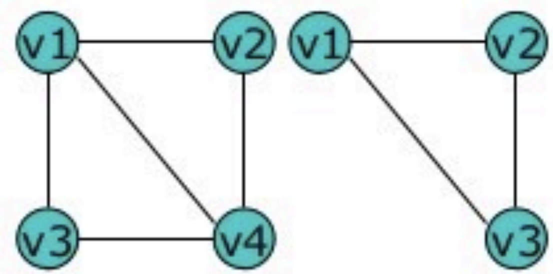
No Simple



Ponderado



Completo



De Similitud

Fuente: http://fundamentosteoriadelacomputacion.blogspot.com/2015/08/teoria-de-grafos.html

Unimodularidad

¿Existe alguna forma de que el problema dado por $max\{c^T x: Ax \leq b, x \geq 0\}$ tenga por vector solución $x^* \in Z_+$ para cualquier b?

Teorema (Hoffman y Kruskal): Sea A una matriz integral $m \times n$. El poliedro $Q := \{x: c \leq Ax \leq d, l \leq x \leq u\}$ es integral por todo vector entero c, d, l, u si y sólo si A es una matriz totalmente unimodular.

Una matriz A es totalmente unimodular si cada sub-matriz cuadrada tiene determinante $\Delta \in \{-1,0,+1\}$. Esto implica necesariamente que las celdas de esta matriz sea -1, 0 ó $+1$.

Teorema: Las matrices de incidencia son totalmente unimodulares.

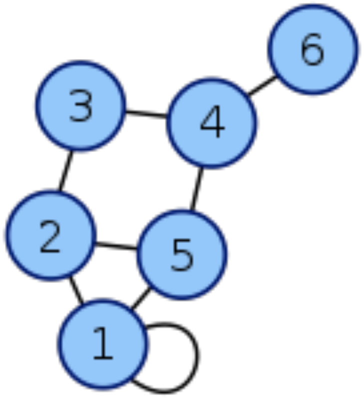
Este teorema implica que los problemas que pueden ser representados como un grafo (también llamados problemas de redes, network problems), al ser formulados como problemas de programación lineal tienen solución entera.

Grafos

Un grafo, $G = (V, A)$, se define como conjunto de vértices, V, conectados por un conjunto de arcos, A.

- Para representar un grafo se pueden utilizar tanto estructuras de listas como matrices. Cualquiera sea la forma que se elija utilizar, estas contienen la información respecto a la adyacencia e incidencia de este grafo.

  1. **Adyacencia**: Contiene la información de cuáles vértices están conectados entre sí.
  2. **Indicencia:** Contiene la información de cuáles arcos *inciden* en cada vértice.



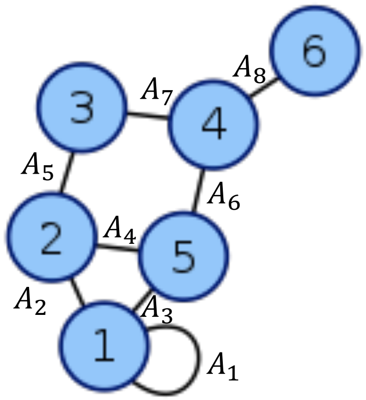
Conjuntos:

$V = \{1,2,3,4,5,6\}$

$A = \{\{1,1\},\{1,2\},\{1,5\},\{2,3\},\{2,5\},\{3,4\},\{4,5\},\{4,6\}\}$

Fuente: https://es.wikipedia.org/wiki/Teor%C3%ADa_de_grafos

Grafos



Fuente: https://es.wikipedia.org/wiki/Teor%C3%ADa_de_grafos

Conjuntos:

$V = \{1,2,3,4,5,6\}$

$A = \{\{1,1\},\{1,2\},\{1,5\},\{2,3\},\{2,5\},\{3,4\},\{4,5\},\{4,6\}\}$

donde, $a_{i,j} \coloneqq 1, si\ existe\ conexión$

Matriz de adyacencia

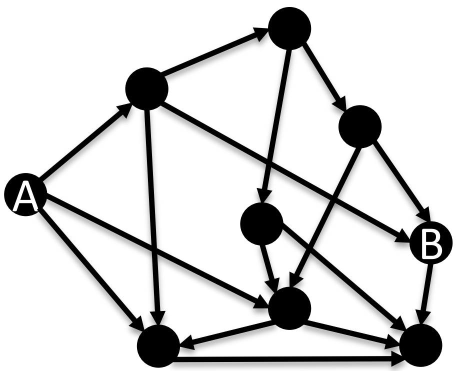
$$
\begin{array}{c c}
\begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{matrix} &
\begin{pmatrix}
1 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix} \\
 & \begin{matrix} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 \end{matrix}
\end{array}
$$

Matriz de adyacencia

$$
\begin{array}{c c}
\begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{matrix} &
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix} \\
 & \begin{matrix} A_1 & A_2 & A_3 & A_4 & A_5 & A_6 & A_7 & A_8 \end{matrix}
\end{array}
$$

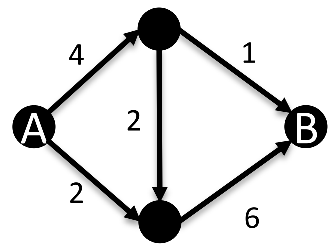
Ejemplos

Camino más corto



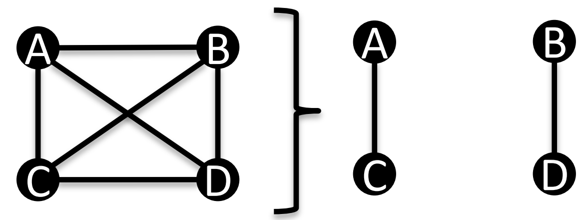
- ¿Cuál es el camino de menor distancia entre A y B?
- ¿Cuál es el camino de menor coste entre A y B?
- ¿Cuál es el camino de menor distancia entre A y B con un coste no superior a k?

Planificación de carreteras, tuberías, sistemas eléctricos, …



- ¿Cuál es la máxima cantidad de energía que se puede transportar desde A a B?
- ¿Cuál es el máximo tráfico que soporta el trayecto de A a B?

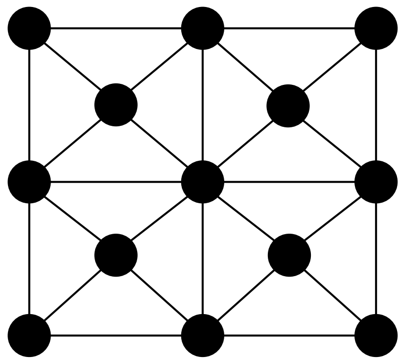
Matching



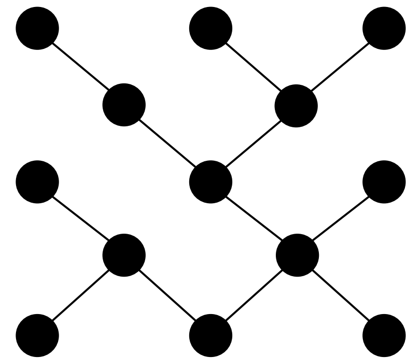
- ¿Cómo hacer parejas de nodos? (dating apps)

UNIVERSIDAD DE SANTIAGO DE CHILE

Spanning trees



Un grafo



Spanning tree

<u>Teorema</u>: Sea un grafo $G = (V, A)$. El politopo del spanning tree de G es descrito por las siguientes desigualdades

$$\sum_{e \in E[S]} x_e \leq |S| - 1, \qquad \emptyset \neq S \subset V$$

$$\sum_{e \in E[S]} x_e = |V| - 1$$

$$x_e \geq 0, \qquad e \in E$$

Algoritmo de Kruskal
(minimum spanning tree, MST)

Algoritmo de Kruskal:

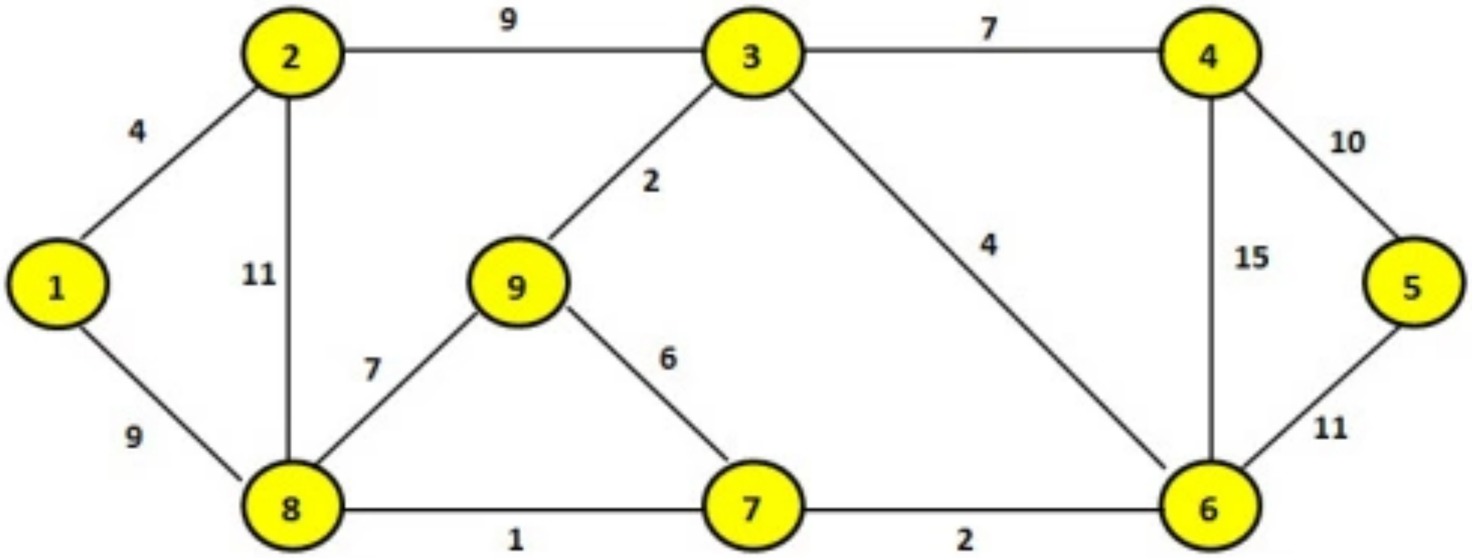
1. Primero elegir el arco o ramal de menor longitud entre toda la red, si el valor de longitud se repite se selecciona de forma arbitraria.

2. Segundo vuelvo a repetir el primer paso esta vez teniendo la precaución de no formar ciclos.

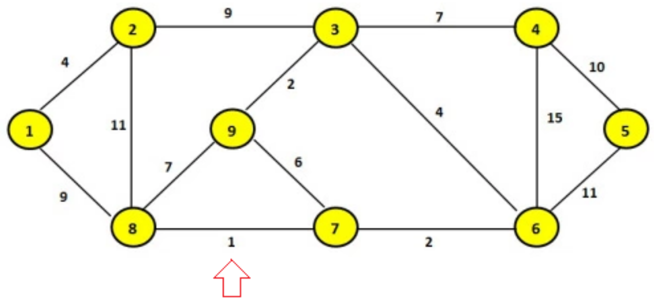
3. Finalmente el algoritmo termina cuando todos los nodos de la red este conectados.

Algoritmo de Kruskal (minimum spanning tree, MST) - Ejemplo

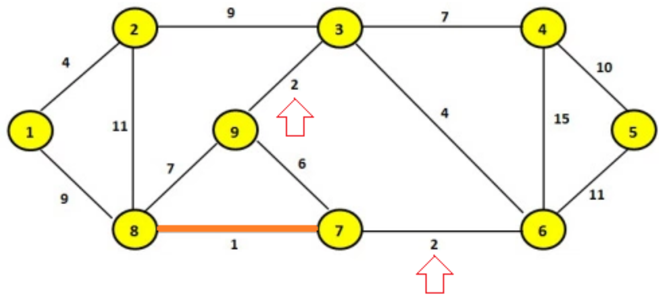
Suponga que usted está encargado de construir los caminos que unirán a un conjunto de pueblos que hasta ahora no tienen contacto entre sí. Se estudiaron las posibles conexiones entre los pueblos y el costo asociado a construir estos caminos. Busque la solución por algoritmo de Kruskal.



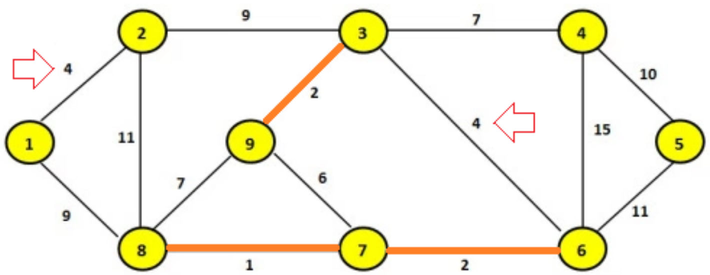
Algoritmo de Kruskal (minimum spanning tree, MST) - Ejemplo



1. Buscar el arco de menor costo entre los pueblos y seleccionarlo.

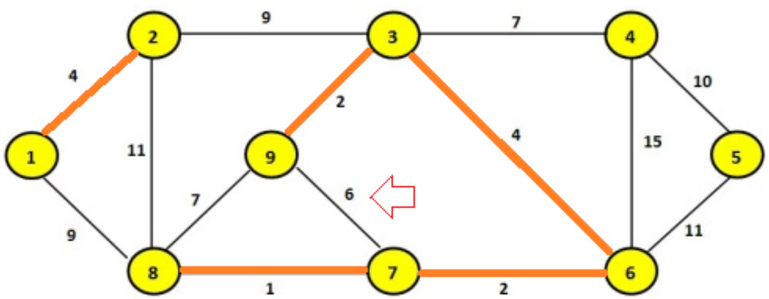


2. Selecciono el siguiente menor, tal que no existan ciclos, hasta unir toda la red (arbitrario en caso de empate).

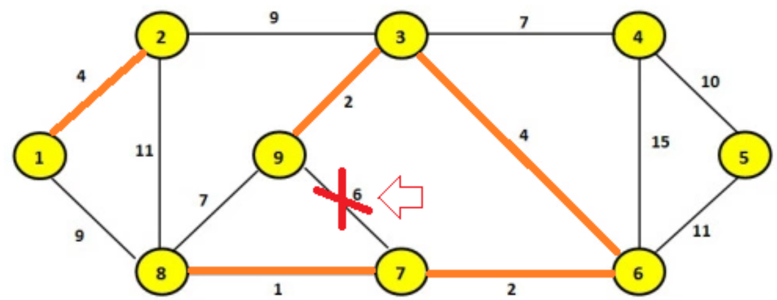


2. Continuación…

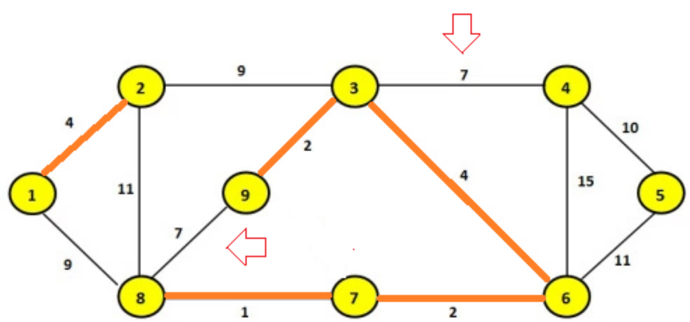
Algoritmo de Kruskal (minimum spanning tree, MST) - Ejemplo



2. Continuación… (ojo con el ciclo!)

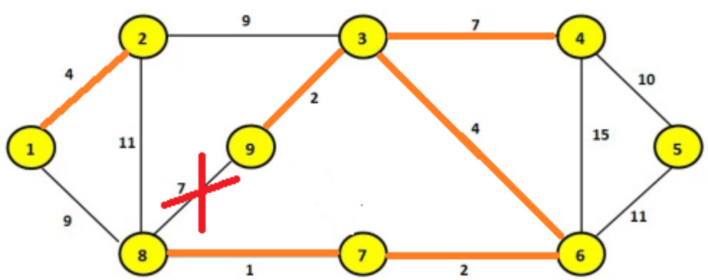


2. Continuación…

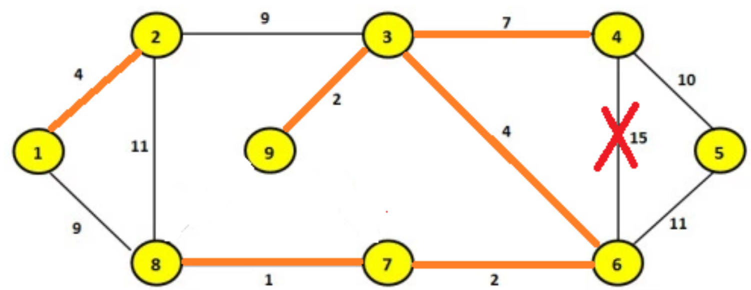


2. Continuación… (ojo con el ciclo!)

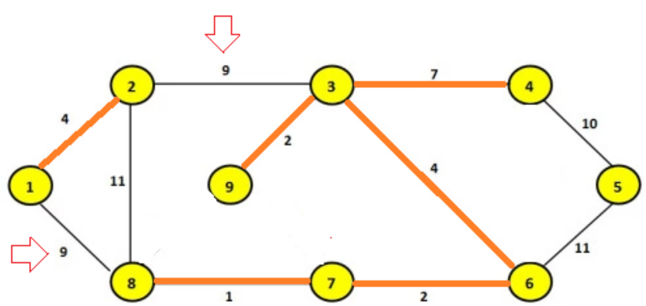
Algoritmo de Kruskal (minimum spanning tree, MST) - Ejemplo



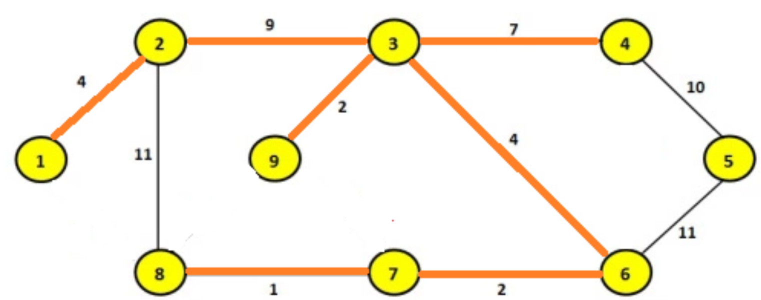
2. Continuación… (ojo con el ciclo!)



2. Continuación…

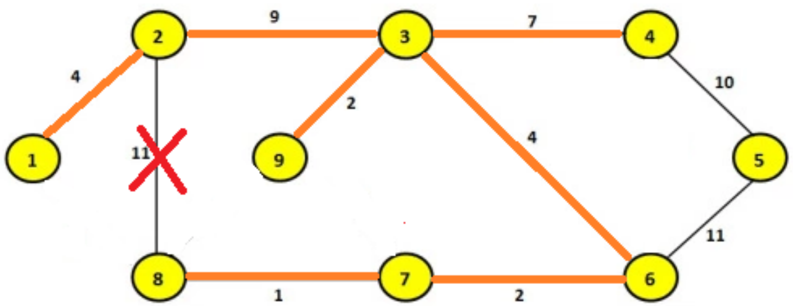


2. Continuación…

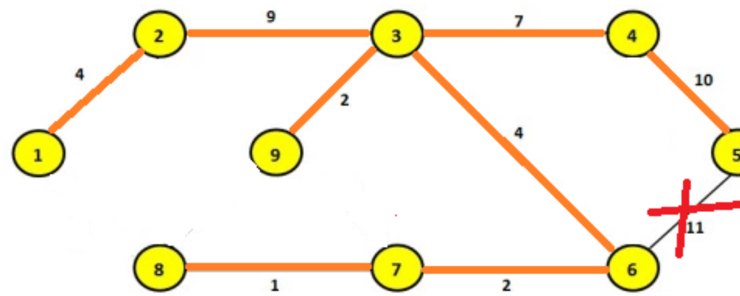


2. Continuación… (ojo con el ciclo!)

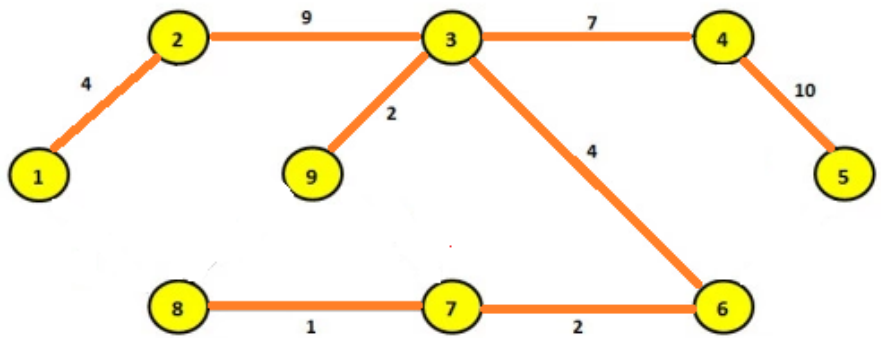
Algoritmo de Kruskal (minimum spanning tree, MST) - Ejemplo



2. Continuación… (ojo con el ciclo!)



2. Continuación…



Solución: 4+9+2+7+4+2+10+1 = 39

Algoritmo de Prim
(minimum spanning tree, MST)

Algoritmo de Prim:

1. Primero se elige cualquier nodo de toda la red y se conecta con el otro nodo mas cercano de menor costo o longitud.

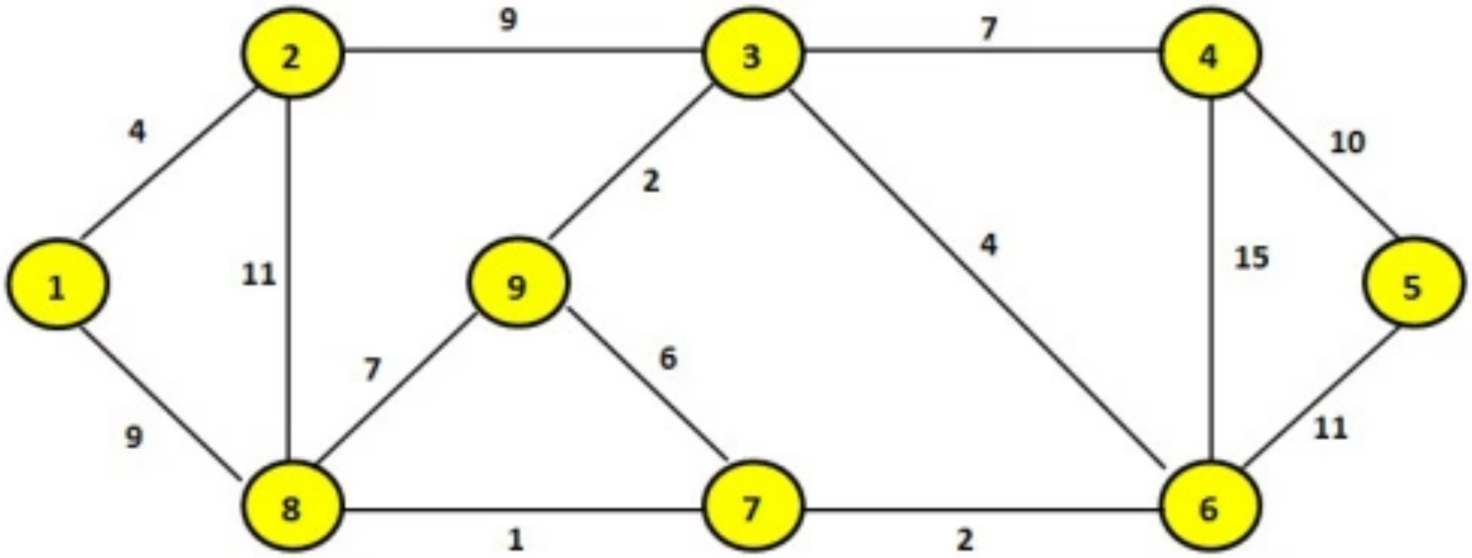
2. Segundo seguir con el primer paso , identificando el nodo no conectado que esta mas cerca o menos costoso de alguno de los nodos conectados.

3. Tercero se agrega este nodo al conjunto de nodos conectados en caso de empate se rompe en forma arbitraria.

4. Finalmente Repetir estos paso hasta que se hayan conectado todos los nodos.

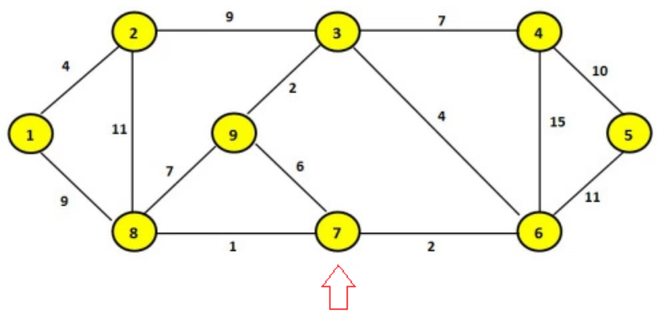
Algoritmo de Prim
(minimum spanning tree, MST) - Ejemplo

Suponga que usted está encargado de construir los caminos que unirán a un conjunto de pueblos que hasta ahora no tienen contacto entre sí. Se estudiaron las posibles conexiones entre los pueblos y el costo asociado a construir estos caminos. Busque la solución por algoritmo de Kruskal.

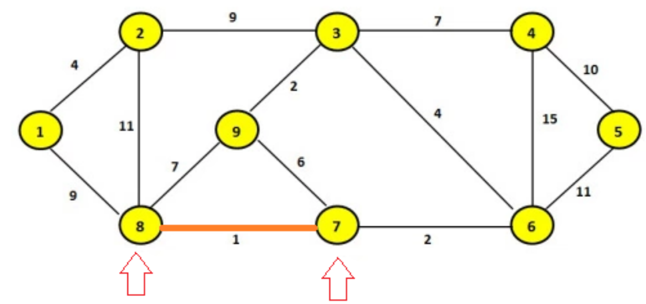


UNIVERSIDAD
DE SANTIAGO
DE CHILE

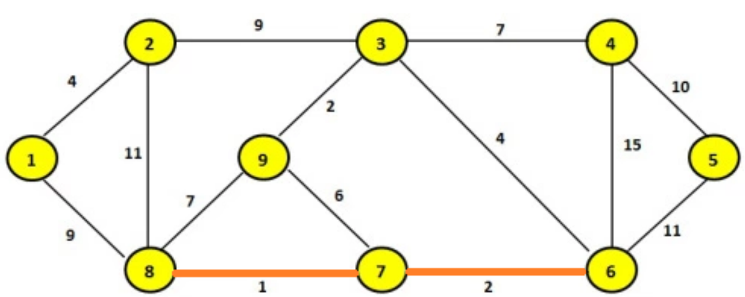
Algoritmo de Prim
(minimum spanning tree, MST) - Ejemplo



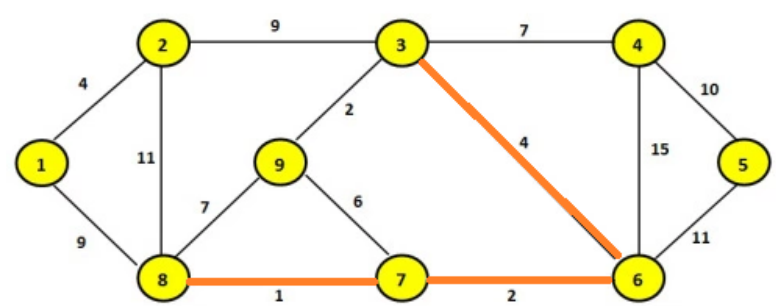
1. Se selecciona cualquier nodo inicialmente elegimos el nodo 7 y luego se conecta con el nodo de costo menor.



2. Selecciono el arco de menor costo.



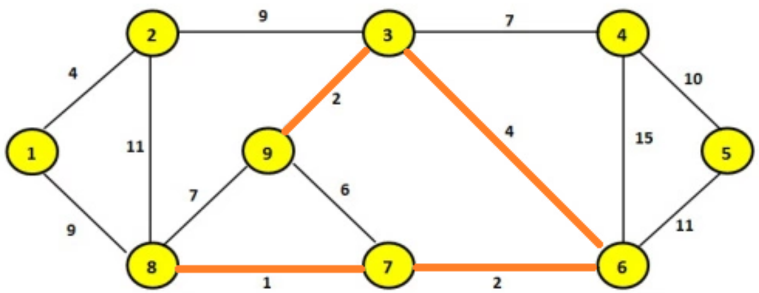
3. Expando a partir del arco antes seleccionado.



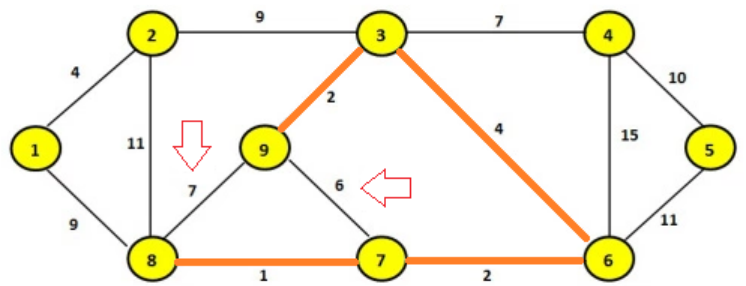
3. Continuación…

UNIVERSIDAD DE SANTIAGO DE CHILE

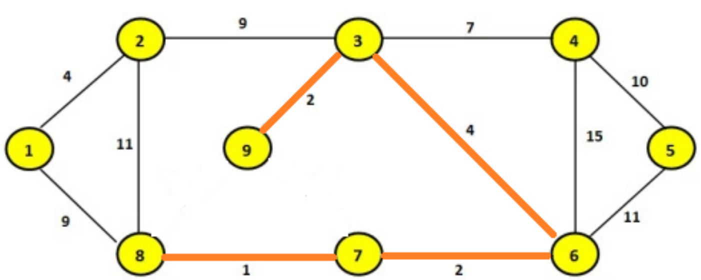
Algoritmo de Prim
(minimum spanning tree, MST) - Ejemplo



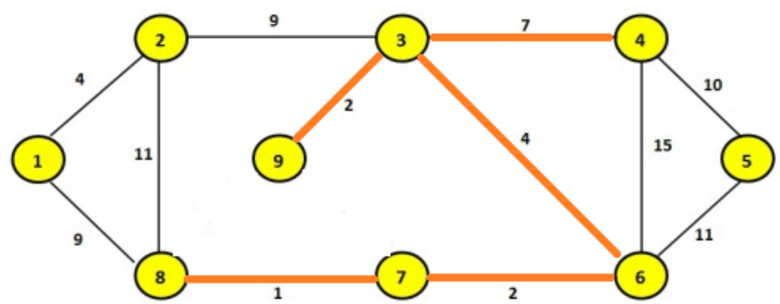
3. Continuación…



3. Continuación (ciclos!)…

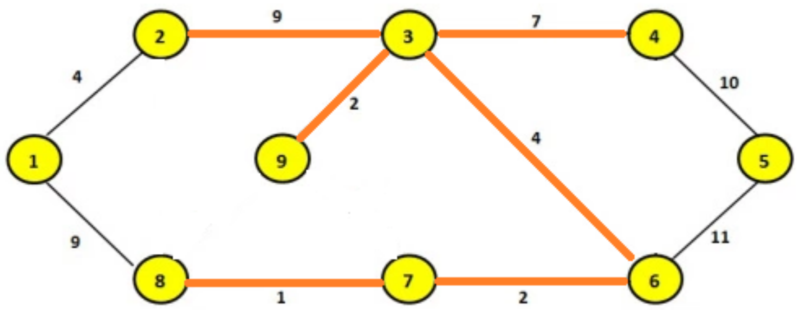


3. Continuación…

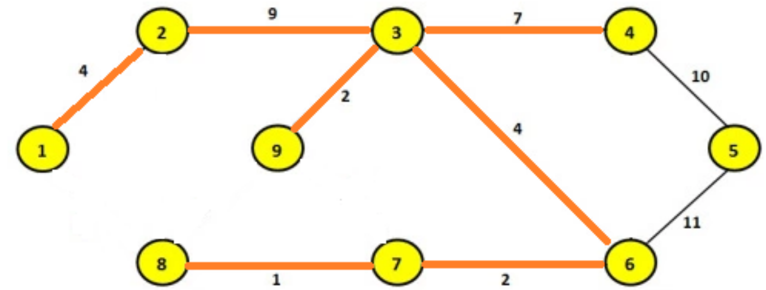


3. Continuación…

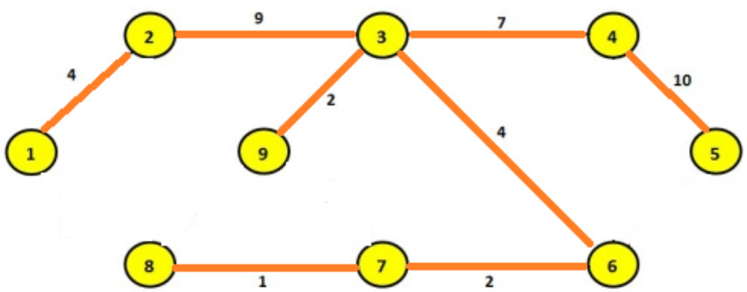
Algoritmo de Prim
(minimum spanning tree, MST) - Ejemplo



3. Continuación…



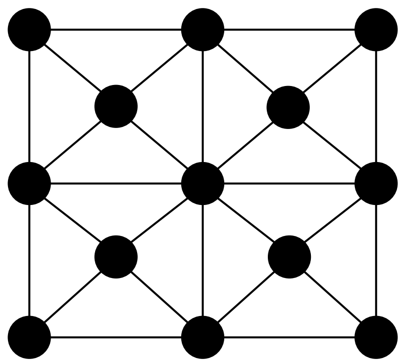
3. Continuación …



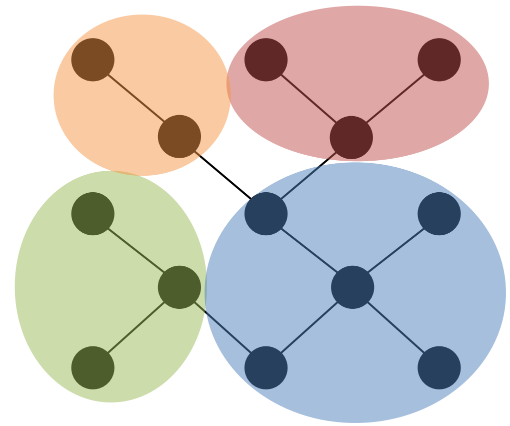
Solución: 4+9+7+10+2+4+1+2= 39

Spanning trees
Clustering

Notar que al crear un spanning tree, lo que se intenta hacer es minimizar la discrepancia entre puntos para unirlos todos a través de otros.



Un grafo



Spanning tree

Al ver tal ilustración se podrían crear sub-grupos dentro del árbol creado. Esto se conoce como agrupamiento o clustering (en inglés). Esta técnica (la idea detrás de esta) es ampliamente usada por gente que trabaja en marketing, finanzas, entre otros, para identificar patrones en el comportamiento de clientes, por ejemplo.