



Universidad Autónoma De San Luis Potosí
Facultad De Ingeniería
Área de Computación e Informática

Alumnos:

Sánchez Juache Francisco Javier

Profesora: *María Teresa Lucio Gil*

Materia: *Estructuras datos avanzadas*

Entrega Proyecto Grafos

Fecha de Entrega: 02 de mayo 2021

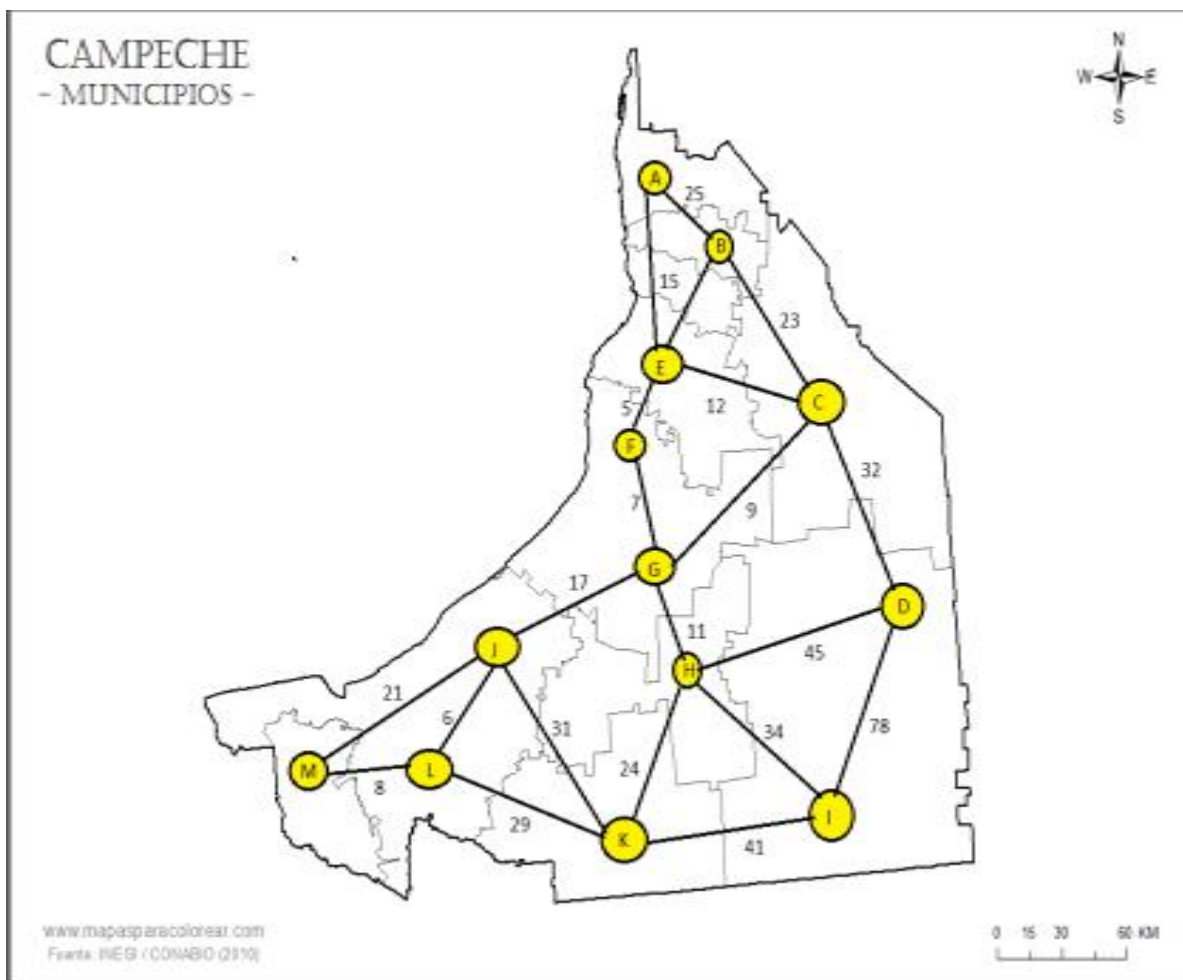
Introducción:

Árbol de expansión mínima: El árbol de expansión mínima es utilizado para problemas en los cuales necesitamos conectar una serie de nodos accesibles entre ellos, esta conexión deberá del ser de menor costo posible y además es muy importante que en ningún momento esta conexión de lugar a que se forme un ciclo.

La aplicación de estos problemas se ubica en las redes de comunicación eléctrica, telefónica, carretera, ferroviaria, aérea, etc.

Problema:

La empresa Telecom será la encargada de las comunicaciones en el estado de Campeche por lo tanto la compañía quiere proveer el servicio de internet para todo el estado, Lo hará instalando un cableado de fibra óptica. La empresa ya ha recopilado los datos necesarios y ha construido un mapa marcando en él las rutas factibles por donde el cableado puede ser colocado y también a adjuntado las distancias que hay entre cada punto.



Mapa

Objetivo del general:

El objetivo general es que en base a la información brindada por la compañía aplicar la teoría de grafos para poder resolver su problema.

La empresa necesita conectar toda la ciudad y por obvias razones esta necesita optimizar sus recursos por lo tanto necesita que la longitud sea la menor posible pero que a su vez no haya ninguna ciudad sin conectar.

Descripción del programa:

De acuerdo con la teoría de grafos la respuesta para este problema de la empresa Telecom es que tomando en cuenta el mapa realizado por ellos podemos obtener un grafo no dirigido con pesos, por lo tanto, esta empresa lo que busca es el Árbol de expansión mínimo del grafo. Este nos dirá las rutas por donde el cable será colocado y además sabremos cual es la será la longitud total del cable. Para encontrar el árbol de expansión mínimo tenemos dos algoritmos que no podrán ayudar uno es el Algoritmo de Prim y otro es el algoritmo de Kruskal.

En esta ocasión se usaremos el algoritmo de Prim ya que es un poco más sencillo que el de Kruskal.

Desarrollo de la solución:

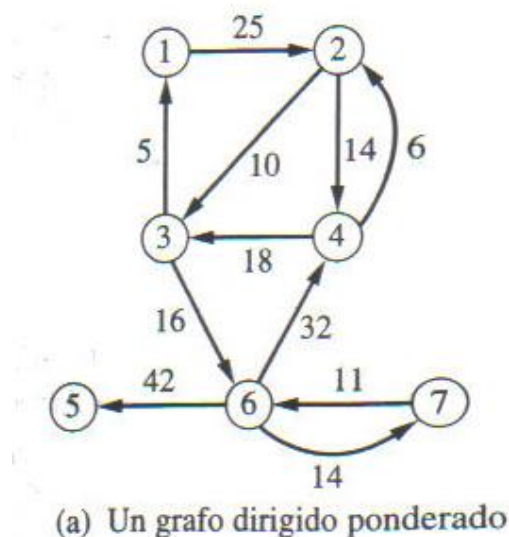
Para comenzar el desarrollo de la solución es saber muy bien que representación de grafos podemos utilizar, las opciones son: usar la representación de lista de adyacencia y la otra es la representación de matriz de adyacencia.

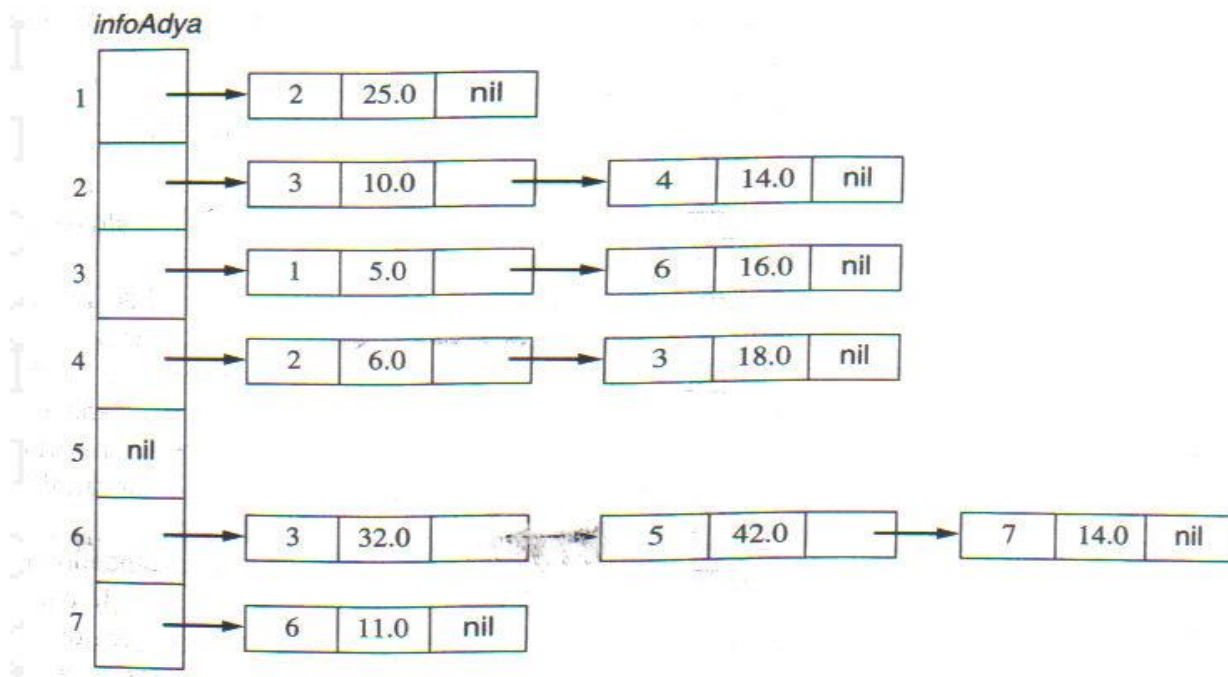
En esta ocasión utilizaremos la lista de adyacencia ya que con esta representación es muy fácil y además es mucho mas optima que la matriz de adyacencia.

Lista de adyacencia:

La lista de adyacencia para un vértice v es una lista enlazada de todos los vértices w adyacentes a v . Un grafo puede ser representado por $|V|$ listas de adyacencias, una para cada vértice.

Ejemplo:





Una vez que conocemos la representación y como se obtiene cada elemento de ella podemos empezar a desarrollar nuestro algoritmo.

En este algoritmo utilizaremos una cola de prioridad, en esta cola nosotros iremos guardando cada nodo que nos encontremos, además esta cola siempre nos regresará el elemento con mayor prioridad, nosotros podemos modificar el “dequeue” para que así nos regrese el elemento con menor prioridad, otra opción es no modificarla, pero al momento de agregar los elementos la prioridad la colocamos en valores negativos.

Algoritmo de Prim(Lista de adyacencia)

Prim(nodo n , Lista<Nodo> arbol)

Definimos la Cola de prioridad

Agregamos el n a la cola con prioridad 0

Mientras (Cola de prioridad no vacia)

 Deafinimos un nuevo nodo auxiliar = extraemos el primer elemento de la cola

 Si(auxiliar no esta en arbol) hacer

 Añadir auxliar a árbol

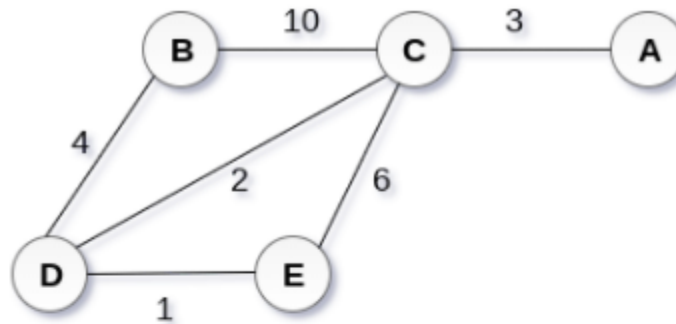
 Para(nodo relacion de auxiliar) hacer

 Agregar a la cola de prioridad relación de auxiliar y en prioridad colocar su peso.

regresar árbol

El algoritmo nos retornara una lista con la secuencia de los nodos que conforman el árbol de expansión mínima.

Ejemplo:



Eligiendo nodo a como inicio

A
0

While

Aux =A

if A no esta en
arbol

Arbol

A

for

Cola de
prioridad

C

3

...

While

Aux =A

if A no esta en
arbol

Arbol

A

for

Cola de
prioridad

C

3

While

Aux =C

if C no esta en
arbol

Arbol

A

C

for

Cola de
prioridad

D

E

B

2

6

10

While

Aux =D

if D no esta en
arbol

Arbol

A

C

D

for

Cola de
prioridad

E

B

E

B

1

4

6

10

While

Aux =E

if

E no esta en
arbol

Arbol

A

C

D

E

for

Cola de
prioridad

D

B

E

C

B

1

4

6

6

10

While

Aux = D	if	D no esta en arbol			
<hr/>					
Arbol		A	C	D	E

While

Aux =B

if

E no esta en
arbol

Arbol

A

C

D

E

B

for

Cola de
prioridad

D

E

C

B

C

4

6

6

10

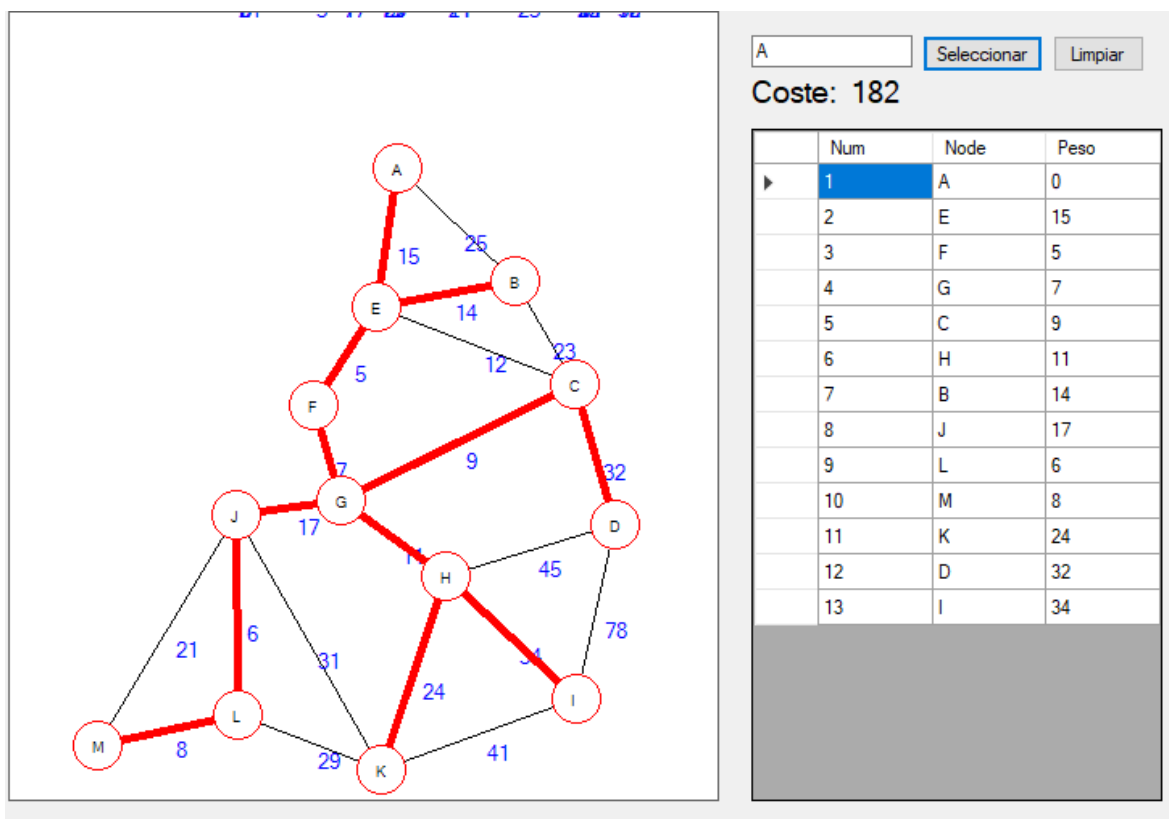
10

Arbol

A	C	D	E	B
---	---	---	---	---

Ahora una vez ya implementado el algoritmo podemos aplicarlo al grafo del problema.

El resultado sería el siguiente:



Podemos ver los arcos que conforman el árbol de expansión mínima y a la derecha podemos ver el orden de los nodos y además sus los costos de cada uno.

Esta sería la respuesta al problema de la empresa.

Conclusión:

La teoría de grafos tiene diversas aplicaciones con las cuales podemos resolver problemas de mundo real. Los grafos en el área de las ciencias de la computación son importantes ya que gracias a ellos podemos hacer modelos para resolver problemas específicos ya sean de redes, trayectos, etc.