

Cedillo Cruz Erik Leonel

1TMA

Estructura de Datos.

Examen Final

1- Describa con sus propias palabras los siguientes Algoritmos:

o Algoritmo de BPA. (busqueda Primero en Amplitud o Anchura)

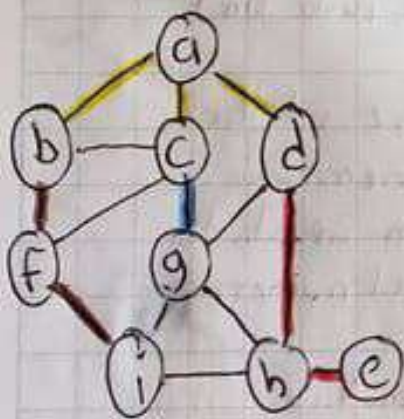
Nos sirve para buscar y recorrer todos los elementos que tiene nuestro grafo "optimizar"

\* Primero se recorre  
Desarrollamos su  
lista de adyacencia  
del grafo dado.

Pasos para su aplicación:

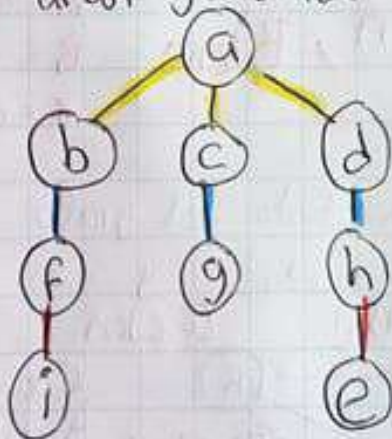
o Necesitamos un origen, podemos elegir cualquiera, según la lista que tengamos, añadimos sus vértices que tiene según la lista.

o Repetimos lo mismo con los nuevos nodos que se añadieron al recorrido, colocamos según su lista sus acompañantes o "vecinos" (y que no hallamos coloreado ya) y quedaría como resultado nuestro árbol generado.



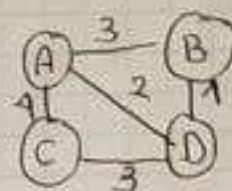
lista

a	b	c	d	e	f	g	h	i
<del>b</del>	<del>a</del>	<del>a</del>	<del>a</del>	x	<del>b</del>	<del>c</del>	<del>d</del>	<del>f</del>
<del>c</del>	<del>c</del>	<del>b</del>	<del>c</del>		<del>c</del>	<del>i</del>	<del>g</del>	<del>d</del>
<del>d</del>	<del>f</del>	<del>f</del>	<del>x</del>		<del>i</del>	<del>x</del>	<del>x</del>	<del>h</del>
		<del>y</del>						<del>e</del>





- Algoritmo de Dijkstra: su objetivo es encontrar la ruta mas corta.
  - Lo que tenemos que hacer para la aplicacion de este algoritmo son los siguientes pasos:
1. Plantear el punto de origen y el punto final
  2. Tambien necesitamos el nombre de cada nodo y su costo del camino por ejemplo:

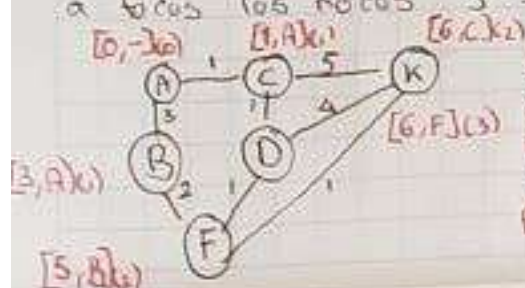


Lo que se intenta hacer es conectar con el menor costo posible punto origen y punto final.

3. En mi caso lo comprendi que tenemos que utilizar etiquetas para cada nodo, donde tenemos la distancia que llevamos de costo, un apuntador hacia el Predecesor y un contador del numero de iteraciones que se han realizado

Etiqueta  $[n, B, k]$    
 $n$  = costo acumulado   
 $B$  = vertice o nodo   
 $k$  = numero de iteraciones

4. Se recorren todo el grafo desde el punto ~~XXXX~~ inicial a todos los nodos y se etiquetara

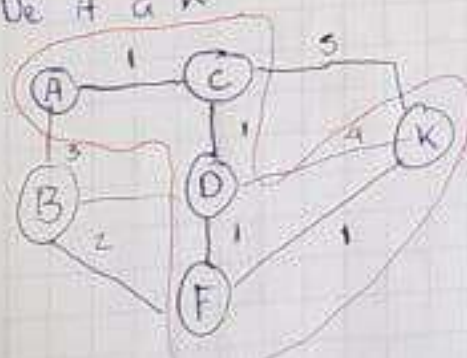


Cuando se termine de etiquetar y llegamos a un punto se el numero de costo es menor que el que ya tenia, se sustituye por el menor.

5. ya cuando tengamos todo el etiquetado de costos, recorremos nuestras etiquetas y ya se con un contador y condicionales, con paramos los caminos de un punto origen a un final el cual en la etiqueta resultado sea menor su costo y generamos ese camino con referencia con apuntadores.

En ese caso seria el camino:

De A a K



$A \rightarrow C \rightarrow D \rightarrow F \rightarrow K$

Etiqueta  $[4, F, 4]$

Costo = 4 es el minimo.



# Cedillo Cruz Erik Leonel

## Algoritmo de Kruskal

"nos ayuda a encontrar el árbol mínimo"

Lo que nos indica es que tenemos que poder conectar todos los nodos que tenemos en nuestro grafo con el mínimo costo creando un árbol.

Para poder encontrar esta ruta debemos de iniciar con los caminos "aristas" de menor valor y debemos de tener en cuenta no crear bucles o ciclos al ir seleccionando, debido a que buscamos crear un árbol, no un grafo.

Pasos:

- Buscamos el menor costo de arista, como se repiten elegimos el que sea o uno de los menores (en este caso (F))

- De F nos vamos al siguiente camino con menor costo

- Repetimos las veces que sean necesarias sin "generar ciclos"

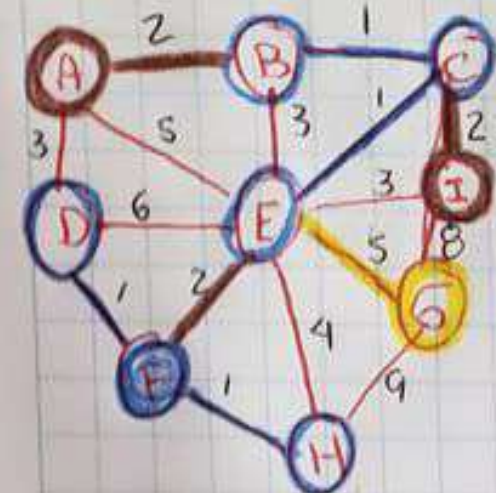
- Terminar primero los menores

- Ahora empezamos con el siguiente menor

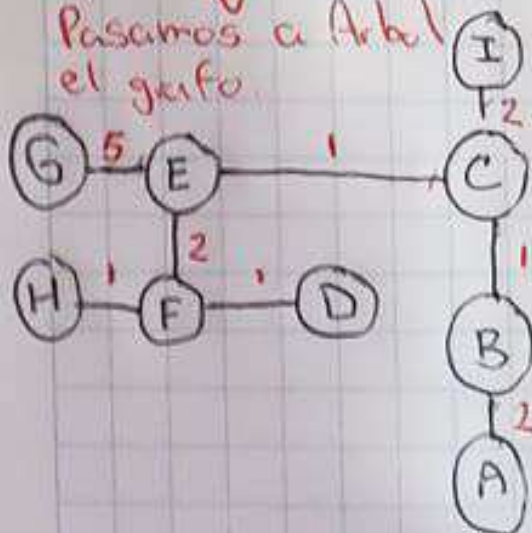
**MUY IMPORTANTE NO CREAR CICLOS**

y continuamos con el que le sigue

- Para saber el costo del nuevo árbol/grafico solo sumamos los costos de cada arista.



↓  
Pasamos a árbol el grafo



$$5 + 1 + 2 + 1 + 2 + 2 + 1 + 1 = 15 \rightarrow \text{costo Total}$$