



# Grafos

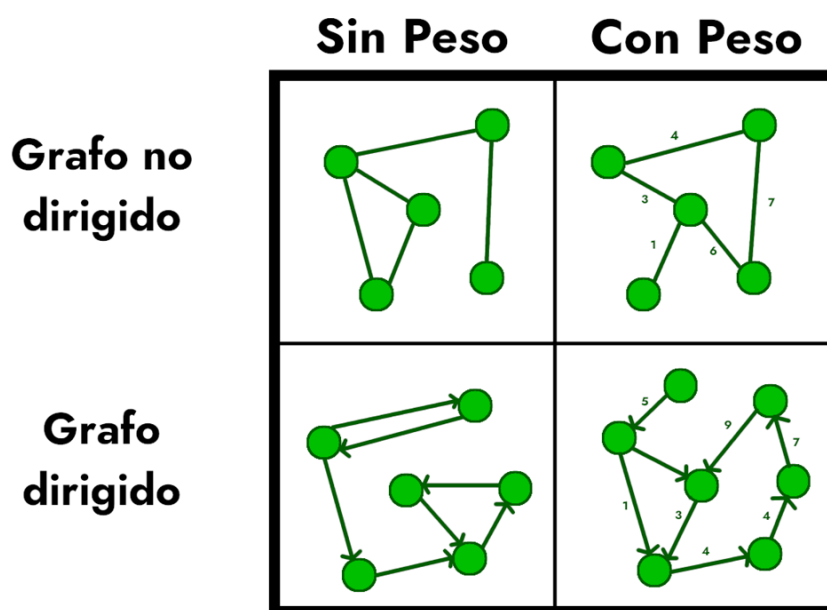


FIG 00: ejemplos de varios tipos de grafos.

Es una estructura matemática la cual modela la relación entre objetos, un ejemplo básico y claro son los puntos de interés de una ciudad ir del punto A al punto B. un grafo  $G$  se define como una colección de nodos y aristas  $(V, E)$  en donde:

- $V$  es un conjunto de nodos.
- $E$  es un conjunto de aristas.

Terminología básica de los grafos:

Término	Significado
Vértice	Es un nodo o un punto del grafo.
Arista	Es una conexión entre 2 vértices.
Camino	Secuencia de vértices donde cada par consecutivo está conectado por aristas.
Ciclo	Camino que comienza y termina en el mismo vértice sin repetir aristas/vértices.

Subgrafo	parte de un grafo que es el sí mismo un grafo.
----------	--

Existen los siguientes tipos de grafos:

- Grafos no dirigidos.
- Grafos dirigidos.
- Grafos ponderados.
- Grafos no ponderados.
- Grafos conectados.
- Grafos desconectados.
- Grafos cíclicos.
- Grafos acíclicos.

Como se puede representar un grafo:

- lista de adyacencia: cada vértice tiene una lista de otros vértices a los que está conectado.
- Matriz de adyacencia: es una matriz de tamaño  $n \times n$  (donde  $n$  es el total de vértices) para indicar si existe una arista se indica en el punto  $i,j$
- Matriz de Incidencia: Matriz que relaciona vértices con aristas.

Algoritmos más comunes de los grafos:

- DFS
- BFS
- Dijkstra
- Kruskal
- Prim

## ***Búsqueda en anchura BFS***

Es un algoritmo capaz de recorrer árboles o grafos **NO DIRIGIDOS**, comienza en la raíz/nodo inicial y se va moviendo por cada uno de los nodos vecinos hasta recorrer todos los nodos.

Características:

- Se tiene que indicar un nodo inicial (PARA INICIAR EL RECORRIDO).
- Se usa una cola para mantener un seguimiento de los nodos que se deben explorar.
- Es útil para encontrar el camino más corto en un grafo no ponderado.

- visita todos los nodos del mismo nivel antes de pasar al siguiente.

### Algoritmo:

Inicio:

- crear un arreglo para guardar los nodos que serán visitados.
- agregar el nodo inicial a la cola.

Mientras que la cola no este vacia:

- extraer el primer nodo de la cola.
- agregar el nodo extraído a los visitados. “marcar como visitado”.
- Para cada arista del nodo extraído agregarlo a la cola solo en caso de que no hubiera sido visitado y que no se encuentre previamente en la cola.

Fin:

- retornar el arreglo de nodos visitados.

Código:

```
def BFS(self, initial_node):
    visited_nodes = []

    if initial_node in self.nodes:
        queue = [initial_node]

        while queue:
            _pivot = queue.pop(0)
            visited_nodes.append(_pivot)

            if _pivot in self.edges:
                for i in self.edges[_pivot]:
                    if i not in queue and i not in visited_nodes:
                        queue.append(i)

        return visited_nodes
```

# ***Búsqueda en profundidad DFS***

Es un algoritmo capaz de recorrer árboles o grafos **NO DIRIGIDOS**, comienza en la raíz/nodo inicial y se va moviendo por cada uno de los nodos vecinos hasta recorrer todos los nodos.

A diferencia de BFS nosotros vamos a explorar lo más profundo posible. Se utiliza una pila para gestionar el orden de las visitas.

Inicio:

- crear un arreglo para guardar los nodos que serán visitados.
- agregar el nodo inicial a la pila.

Mientras que la pila no este vacia:

- sacar un nodo de la pila.
- agregar este nodo al arreglo de visitados.
- para cada arista del nodo extraído agregar a la pila en caso de que no haya sido visitado.

Fin:

- retornar el arreglo de visitados.

```
def DFS(self, initial_node):
    visited_nodes = []

    if initial_node in self.nodes:
        stack = [initial_node]

        while stack:
            _pivot = stack.pop()

            if _pivot not in visited_nodes:
                visited_nodes.append(_pivot)

                if _pivot in self.edges:
                    for i in reversed(self.edges[_pivot]):
                        if i not in stack and i not in visited_nodes:
                            stack.append(i)

        return visited_nodes
```

# ***Dijkstra***

Fue creado en 1956 por Edsger Dijkstra. es un algoritmo de optimización que se usa para encontrar la ruta más corta desde un nodo origen a todos los demás nodos accesibles de un grafo ponderado. (Advertencia: este algoritmo funciona únicamente con grafos ponderados NO negativos).

Principios básicos:

- Inicialización:

Se tiene que seleccionar un nodo origen el cual va tener una distancia de 0.

Se marcan todos los nodos como no visitados.

En caso de que un nodo no sea accesible va a tener un peso infinito.

- Evaluación de vecinos:

Para cada nodo actual se consideran las distancias hacia sus vecinos.

Si la distancia de un nodo actual a un vecino es inferior a un cálculo previo entonces se actualiza la distancia previa.

- Selección de la menor distancia:

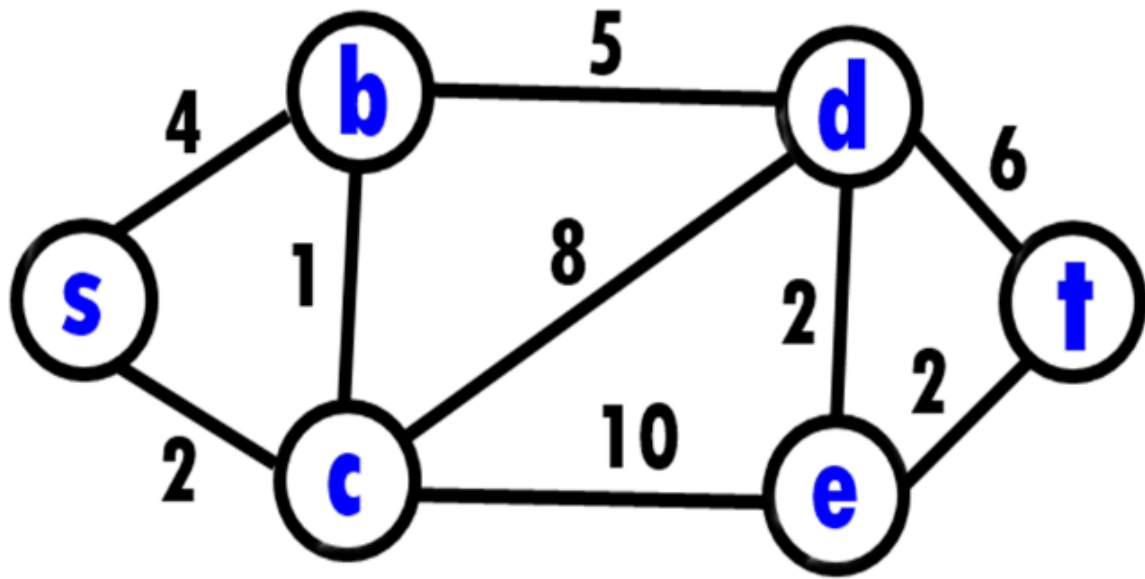
Se marca el nodo actual como visitado.

Se selecciona el nodo no visitado con la menor distancia calculada como nuevo nodo actual y se repite una y otra vez el proceso hasta que todos los nodos hayan sido visitados.

es un algoritmo que es capaz de encontrar la distancia mínima para llegar a cualquier punto, se basa en tabular los nodos y luego moverse por las distancias acumuladas más cortas:

Ejemplo:

Calcular el algoritmo de Dijkstra sobre el siguiente grafo desde el nodo "s"



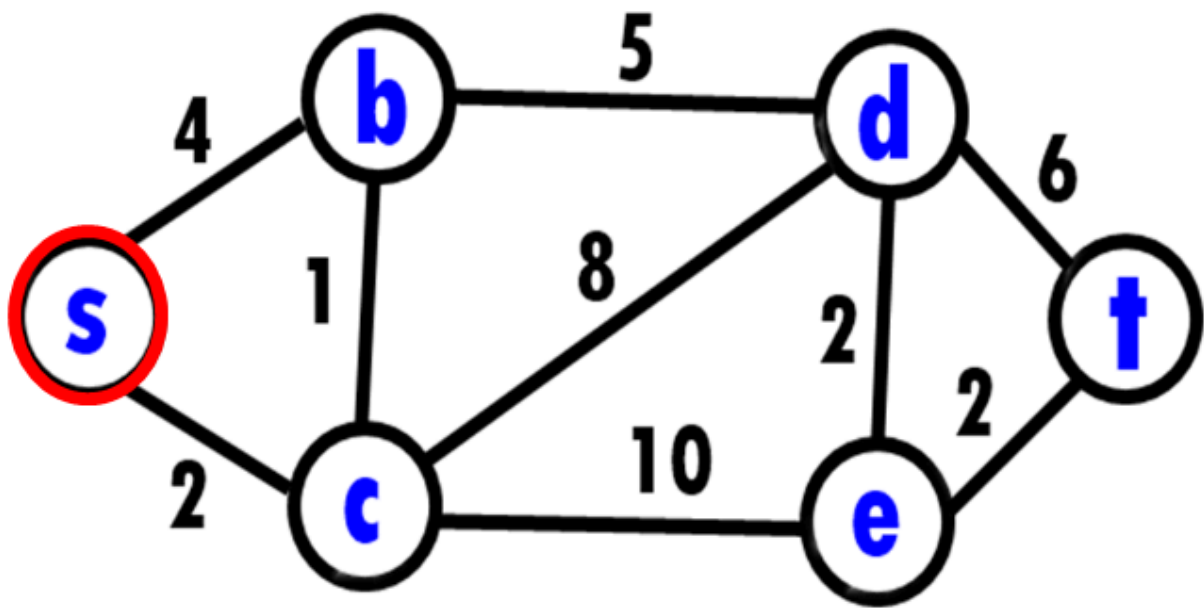
0:

Crear una matriz de tamaño “cantidad de nodos \* cantidad de nodos” para este caso nuestro grafo tiene 6 nodos por ello nuestra tabla empieza de tamaño 6x6:

	A	B	C	D	E	F	G
NODO	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5	
s							
b							
c							
d							
e							
t							

1:

En la casilla de “PASO 0” se indica el nodo principal con una distancia de 0 desde el nodo s, luego ese nodo se marca como visitado y se anulan las casillas siguientes.



	A	B	C	D	E	F	G
	NODO	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5
s		(0, s)	x	x	x	x	x
b							
c							
d							
e							
t							

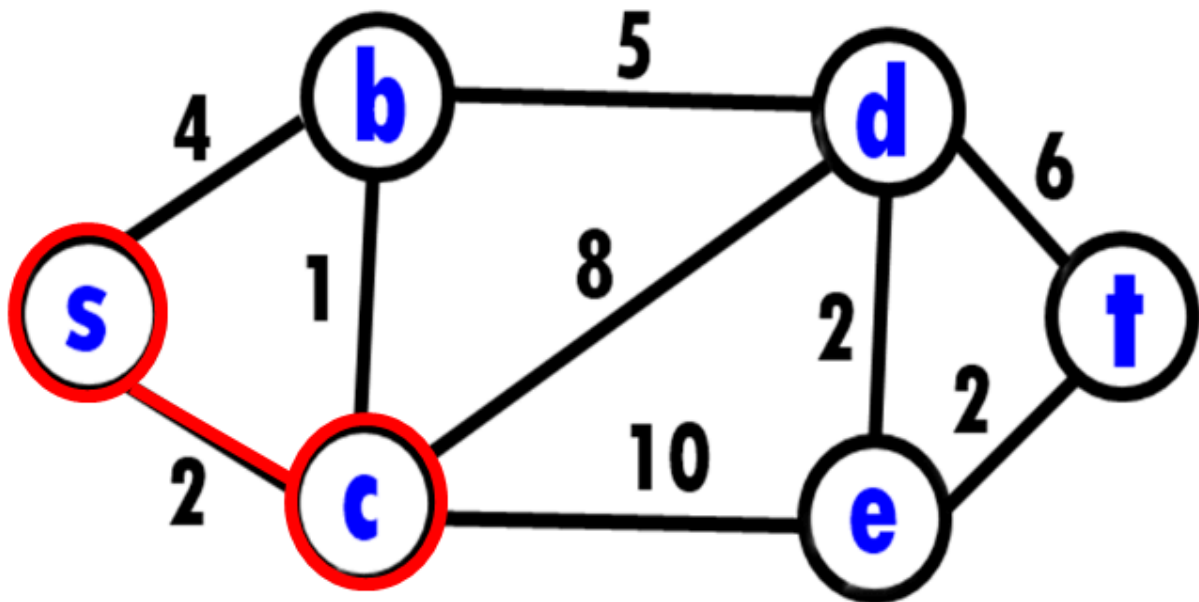
2:

En este paso se van a calcular todas las distancias de la casilla “PASO 0”, en este momento la distancia acumulada es 0 y se tiene que calcular la menor distancia para llegar a los vecinos de s (Advertencia aquellas partes que no son accesibles desde S van a tener una distancia de infinito)

	A	B	C	D	E	F	G
	NODO	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5
s		(0, s)	x	x	x	x	x
b		(4, s)					
c		(2, s)					
d		INFINITY					
e		INFINITY					
t		INFINITY					

3:

De todas las distancias calculadas anteriormente se tiene que seleccionar el camino más corto, este camino pasa a la siguiente casilla “PASO 1” se marca como visitada y se anulan las casillas siguientes.



	A	B	C	D	E	F	G	H
	NODO	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5	
s		(0, s)	x	x	x	x	x	
b		(4, s)						
c		(2, s)	(2, s)	x	x	x	x	
d		INFINITY						
e		INFINITY						
t		INFINITY						

En este momento nuestra distancia acumulada es 2 y ahora estamos parados en el nodo “c” por ello la nueva búsqueda debe hacerse desde c con una distancia acumulada de 2.

4:

Se recalculan las distancias menores desde el nodo c recordando que tenemos que tenemos una distancia acumulada de 2. (Advertencia hay que ignorar todas las casillas anuladas y visitadas)

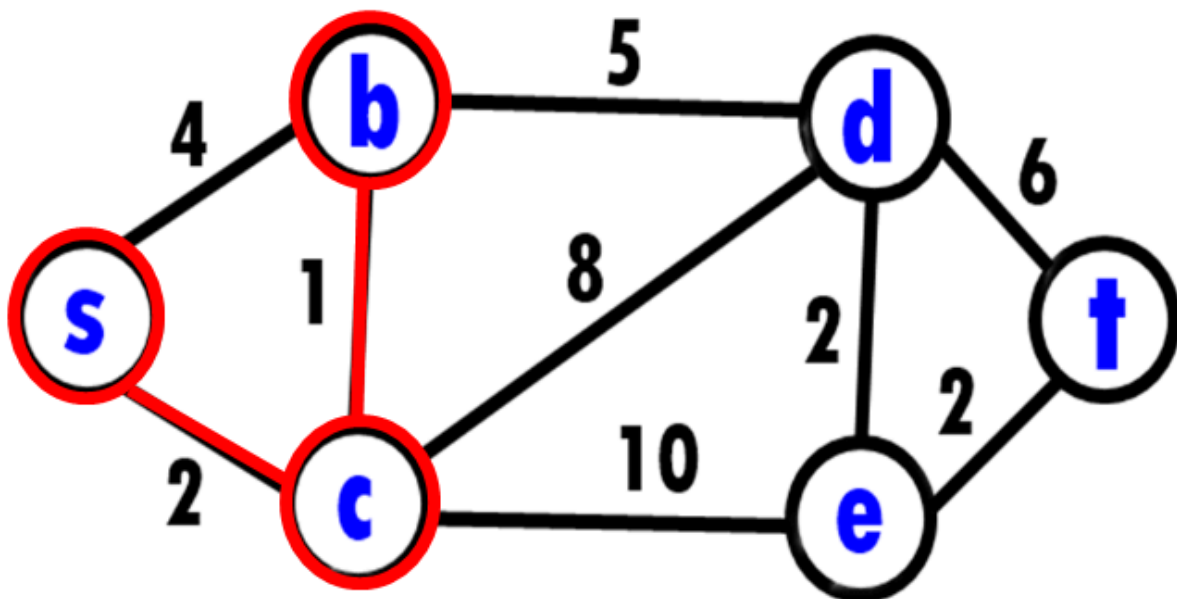


A	B	C	D	E	F	G
NODO	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5
s	(0, s)	x	x	x	x	x
b	(4, s)	(3, c)				
c	(2, s)	(2, s)	x	x	x	x
d	INFINITY	(10, c)				
e	INFINITY	(12, c)				
t	INFINITY	INFINITY				

Si usted observa la casilla b en el paso 1 notará que se actualizó la distancia mínima.

5:

De todas las distancias calculadas anteriormente se tiene que seleccionar el camino más corto, este camino pasa a la siguiente casilla “PASO 2” se marca como visitada y se anulan las casillas siguientes.



A	B	C	D	E	F	G
NODO	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5
s	(0, s)	x	x	x	x	x
b	(4, s)	(3, c)	(3, c)	x	x	x
c	(2, s)	(2, s)	x	x	x	x
d	INFINITY	(10, c)				
e	INFINITY	(12, c)				
t	INFINITY	INFINITY				

En este momento nuestra distancia acumulada es 3 y ahora estamos parados en el nodo “b” por ello la nueva búsqueda debe hacerse desde b con una distancia acumulada de 3.

6:

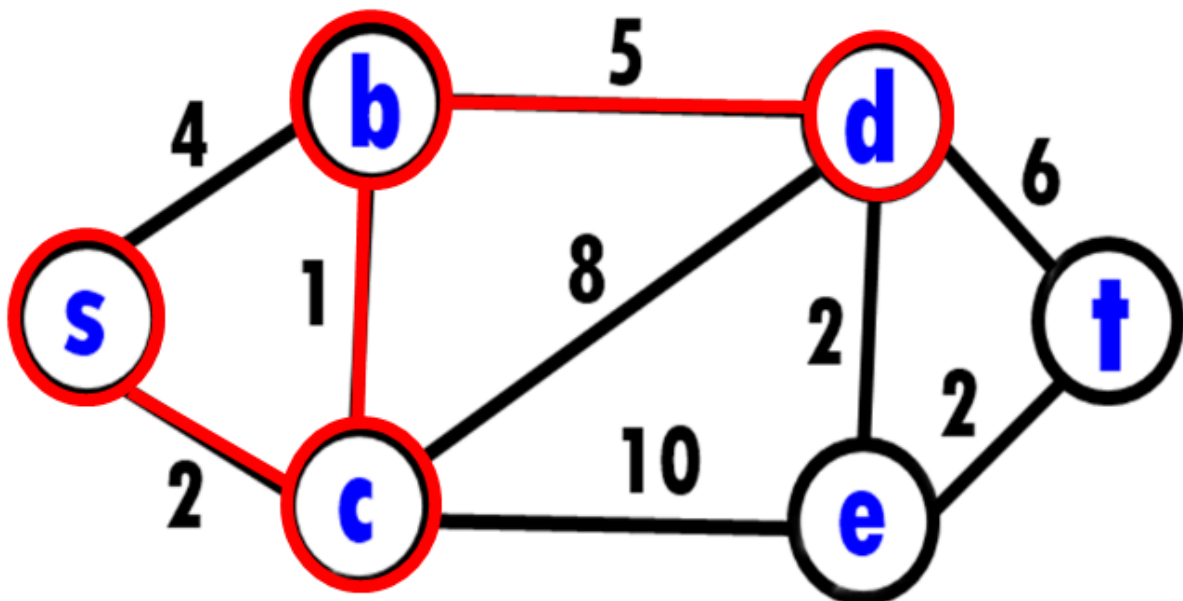
Se recalculan las distancias menores desde el nodo b recordando que tenemos que tenemos una distancia acumulada de 3. (Advertencia hay que ignorar todas las casillas anuladas y visitadas)

	A	B	C	D	E	F	G
	NODO	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5
s		(0, s)	x	x	x	x	x
b		(4, s)	(3, c)	(3, c)	x	x	x
c		(2, s)	(2, s)	x	x	x	x
d		INFINITY	(10, c)	(8, b)			
e		INFINITY	(12, c)	(12, c)			
t		INFINITY	INFINITY	INFINITY			

Si usted observa la casilla d en el paso 2 notará que se actualizó la distancia mínima. También hay que notar que la casilla e no tiene conexión para este paso con e pero no se coloca infinito sino que se deja el peso anterior.

7:

De todas las distancias calculadas anteriormente se tiene que seleccionar el camino más corto, este camino pasa a la siguiente casilla "PASO 3" se marca como visitada y se anulan las casillas siguientes.



	A	B	C	D	E	F	G
	NODO	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5
s		(0, s)	x	x	x	x	x
b		(4, s)	(3, c)	(3, c)	x	x	x
c		(2, s)	(2, s)	x	x	x	x
d		INFINITY	(10, c)	(8, b)	(8, b)	x	x
e		INFINITY	(12, c)	(12, c)			
t		INFINITY	INFINITY	INFINITY			

En este momento nuestra distancia acumulada es 8 >> (2+1+5) y ahora estamos parados en el nodo “d” por ello la nueva búsqueda debe hacerse desde d con una distancia acumulada de 8.

8:

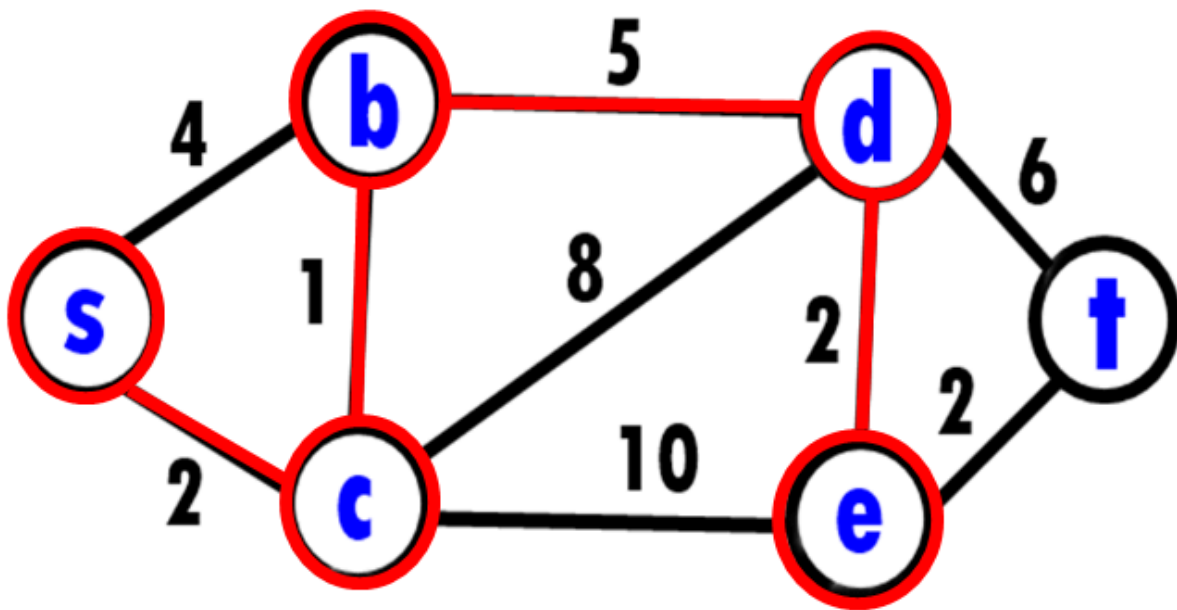
Se recalculan las distancias menores desde el nodo d recordando que tenemos que tenemos una distancia acumulada de 8. (Advertencia: hay que ignorar todas las casillas anuladas y visitadas).

	A	B	C	D	E	F	G
	NODO	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5
s		(0, s)	x	x	x	x	x
b		(4, s)	(3, c)	(3, c)	x	x	x
c		(2, s)	(2, s)	x	x	x	x
d		INFINITY	(10, c)	(8, b)	(8, b)	x	x
e		INFINITY	(12, c)	(12, c)	(10, d)		
t		INFINITY	INFINITY	INFINITY	(14, d)		

Si usted observa la casilla “e” en el paso 3, notará que se actualizó la distancia mínima.

9:

De todas las distancias calculadas anteriormente se tiene que seleccionar el camino más corto, este camino pasa a la siguiente casilla “PASO 4” se marca como visitada y se anulan las casillas siguientes.



	A	B	C	D	E	F	G
	NODO	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5
s		(0, s)	x	x	x	x	x
b		(4, s)	(3, c)	(3, c)	x	x	x
c		(2, s)	(2, s)	x	x	x	x
d		INFINITY	(10, c)	(8, b)	(8, b)	x	x
e		INFINITY	(12, c)	(12, c)	(10, d)	(10, d)	x
t		INFINITY	INFINITY	INFINITY	(14, d)		

En este momento nuestra distancia acumulada es 10 y ahora estamos parados en el nodo “e” por ello la nueva búsqueda debe hacerse desde e con una distancia acumulada de 10.

10:

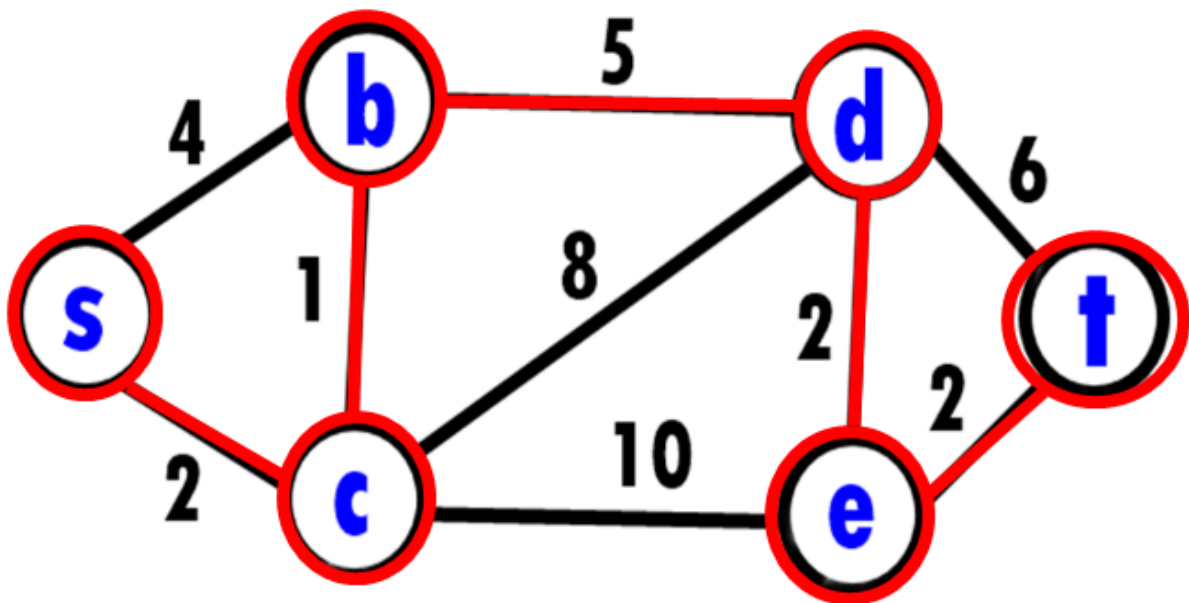
Se recalculan las distancias menores desde el nodo “e” recordando que tenemos que tenemos una distancia acumulada de 10. (Advertencia: hay que ignorar todas las casillas anuladas y visitadas).

	A	B	C	D	E	F	G
	NODO	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5
s		(0, s)	x	x	x	x	x
b		(4, s)	(3, c)	(3, c)	x	x	x
c		(2, s)	(2, s)	x	x	x	x
d		INFINITY	(10, c)	(8, b)	(8, b)	x	x
e		INFINITY	(12, c)	(12, c)	(10, d)	(10, d)	x
t		INFINITY	INFINITY	INFINITY	(14, d)	(12, e)	

Si usted observa la casilla “t” en el paso 4, notará que se actualizó la distancia mínima.

11:

De todas las distancias calculadas anteriormente se tiene que seleccionar el camino más corto, este camino pasa a la siguiente casilla “PASO 5”... en estos momentos ya terminamos con el algoritmo... por ello marcamos como visitada y se completaría la tabla en la siguiente iteración.

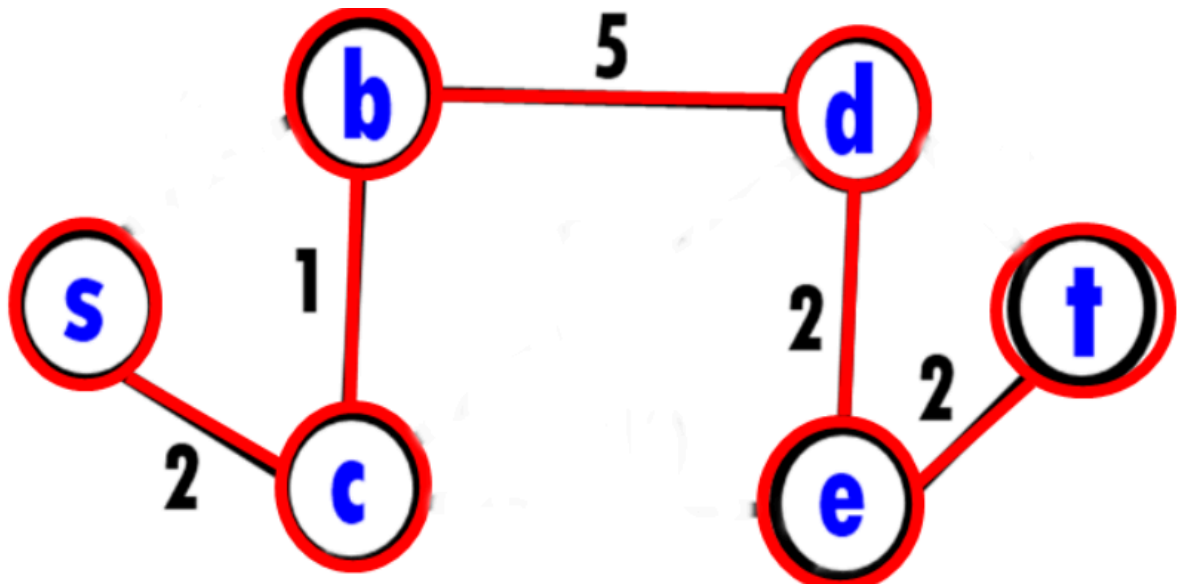


	A	B	C	D	E	F	G
	NODO	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5
s		(0, s)	x	x	x	x	x
b		(4, s)	(3, c)	(3, c)	x	x	x
c		(2, s)	(2, s)	x	x	x	x
d		INFINITY	(10, c)	(8, b)	(8, b)	x	x
e		INFINITY	(12, c)	(12, c)	(10, d)	(10, d)	x
t		INFINITY	INFINITY	INFINITY	(14, d)	(12, e)	(12, e)

El resultado del algoritmo es la ruta más corta a todos los caminos accesibles.

## ***Cómo moverse en la tabla para obtener la ruta más corta***

El resultado de Dijkstra desde el nodo “s” es un camino del siguiente estilo:



Con la siguiente tabla desde el nodo “s”:

NODO	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5
s	(0, s)	x	x	x	x	x
b	(4, s)	(3, c)	(3, c)	x	x	x
c	(2, s)	(2, s)	x	x	x	x
d	INFINITY	(10, c)	(8, b)	(8, b)	x	x
e	INFINITY	(12, c)	(12, c)	(10, d)	(10, d)	x
t	INFINITY	INFINITY	INFINITY	(14, d)	(12, e)	(12, e)

Ejemplos de movimientos:

para ir de el nodo “s” al nodo “c” usted puede verlo a simple vista observando el grafo, pero si usted observa la tabla se puede perder.

Si usted observa la columna NODO para encontrar el camino tiene que ir a donde está marcado el nodo “c”

NODO	PASO 0	PASO 1
s	(0, s)	x
b	(4, s)	(3, c)
c	(2, s)	(2, s)
d	INFINITY	(10, c)
e	INFINITY	(12, c)
t	INFINITY	INFINITY

Usted observará que el nodo "c" tiene un valor de (2, s) ... lo que se traduce en que se necesitan 2 para llegar a "s" y terminaríamos la búsqueda.

Ahora que pasa si quiero ir desde "s" hasta "e"

NODO	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5
s	(0, s)	x	x	x	x	x
b	(4, s)	(3, c)	(3, c)	x	x	x
c	(2, s)	(2, s)	x	x	x	x
d	INFINITY	(10, c)	(8, b)	(8, b)	x	x
e	INFINITY	(12, c)	(12, c)	(10, d)	(10, d)	x
t	INFINITY	INFINITY	INFINITY	(14, d)	(12, e)	(12, e)

paso a paso:

Lo primero que tengo que hacer es buscar la fila "e" y me doy cuenta que su valor es (10, d) osea sumó 10 a la cuenta y me voy para la fila "d"

Lo segundo que tengo que hacer es buscar la fila "d" y me doy cuenta que su valor es (8, b) osea la suma va en 18 y me voy para la fila "b"

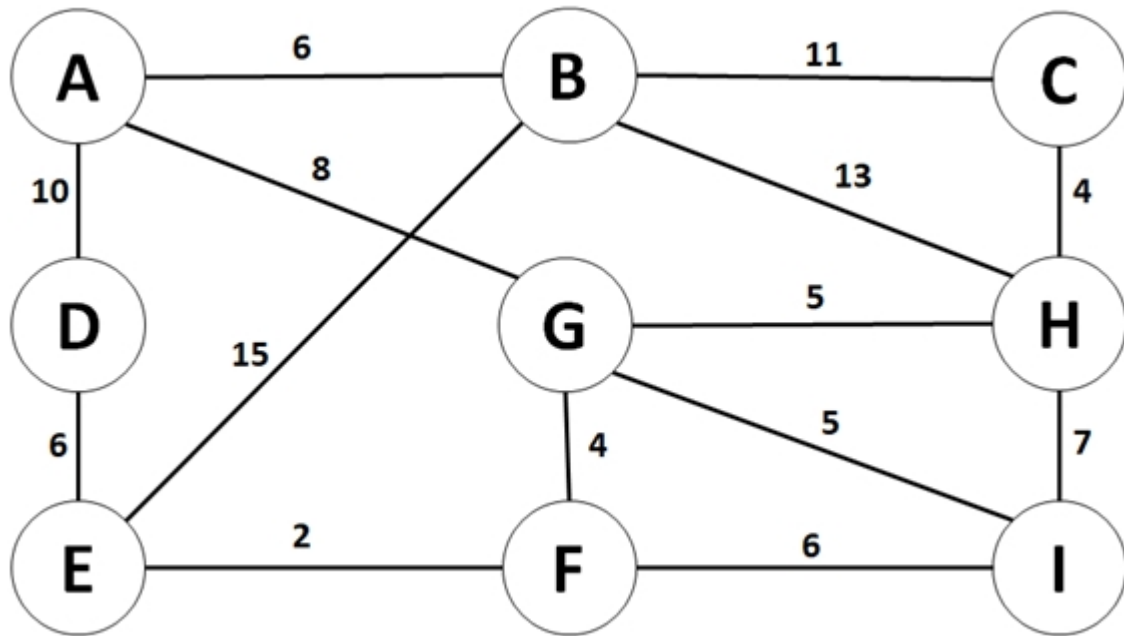
Lo tercero que tengo que hacer es buscar la fila "b" y me doy cuenta que su valor es (3, c) osea la suma va en 21 y me voy para la fila "c"

Lo cuarto que tengo que hacer es buscar la fila "c" y me doy cuenta que su valor es (2, s) osea la suma va en 23 y termina el recorrido.

Si usted lo pudo notar lo único que se está haciendo es ir en reversa en los nodos marcados como visitados.

Ejemplo 02:

Calcular el algoritmo de Dijkstra sobre el siguiente grafo desde el nodo "A"



0:

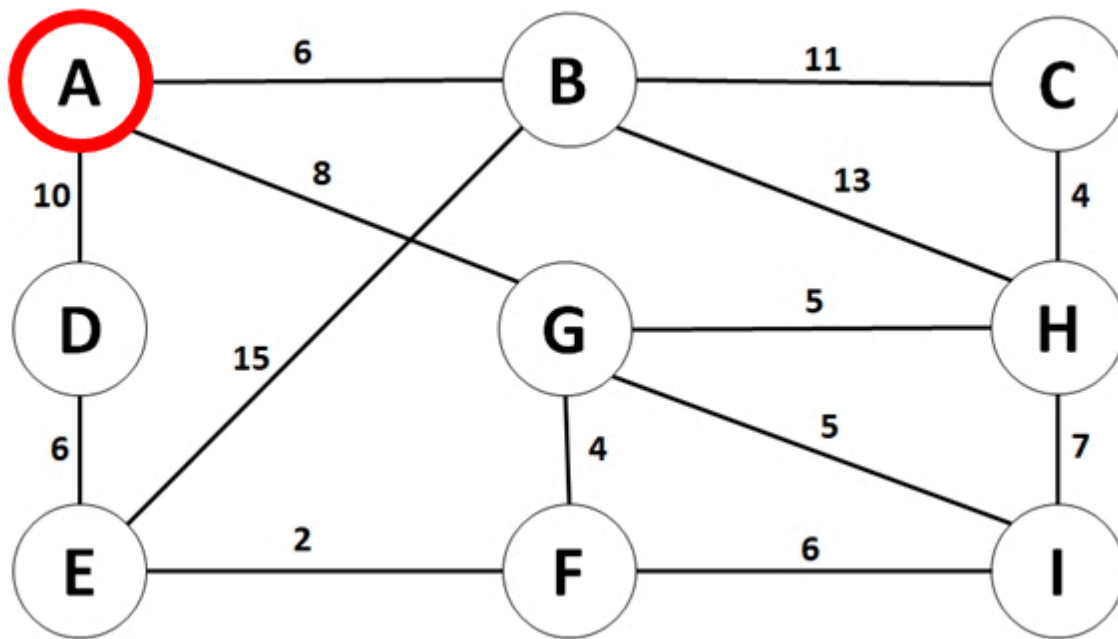
Crear una matriz de tamaño "cantidad de nodos \* cantidad de nodos" para este caso nuestro grafo tiene 9 nodos por ello nuestra tabla empieza de tamaño 9x9:

	A	B	C	D	E	F	G	H	I	J
NODO	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5	PASO 6	PASO 7	PASO 8	
A										
B										
C										
D										
E										
F										
G										
H										
I										

1:

En la casilla de "PASO 0" se indica el nodo principal con una distancia de 0 desde el nodo A, luego ese nodo se marca como visitado y se anulan las casillas siguientes.





	A	B	C	D	E	F	G	H	I	J
		PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5	PASO 6	PASO 7	PASO 8
A		(0, A)	X	X	X	X	X	X	X	X
B										
C										
D										
E										
F										
G										
H										
I										

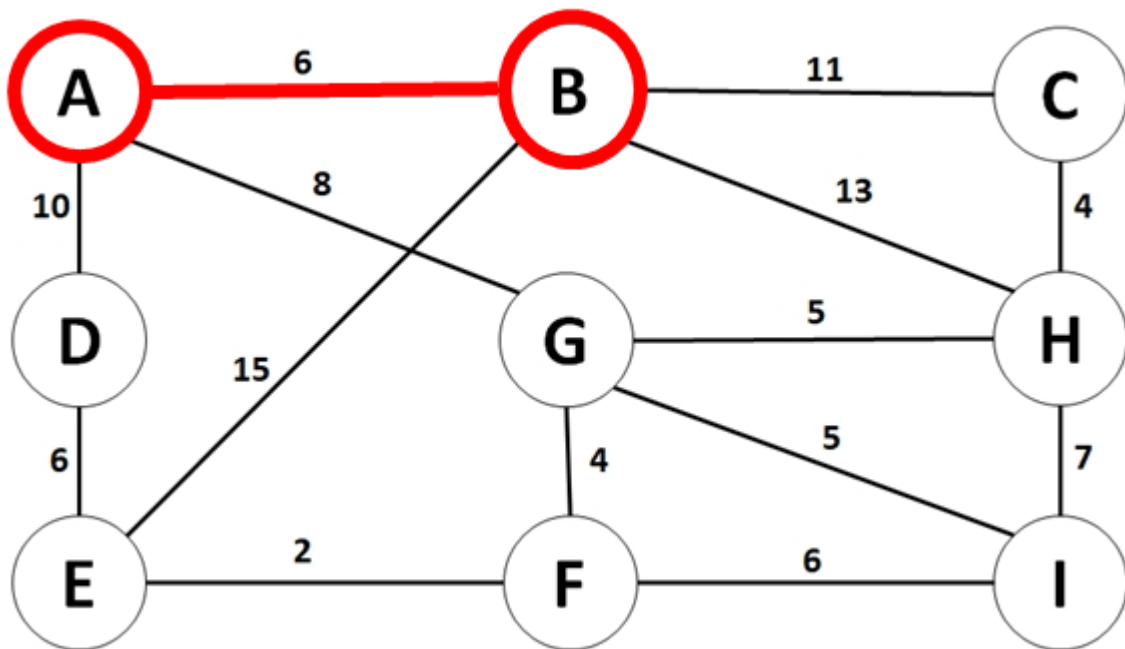
2:

En este paso se van a calcular todas las distancias de la casilla “PASO 0”, en este momento la distancia acumulada es 0 y se tiene que calcular la menor distancia para llegar a los vecinos de s (Advertencia aquellas partes que no son accesibles desde A van a tener una distancia de infinito).

	A	B	C	D	E	F	G	H	I	J
		PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5	PASO 6	PASO 7	PASO 8
A		(0, A)	X	X	X	X	X	X	X	X
B		(6, A)								
C		INFINITY								
D		(10, A)								
E		INFINITY								
F		INFINITY								
G		(8, A)								
H		INFINITY								
I		INFINITY								

3:

De todas las distancias calculadas anteriormente se tiene que seleccionar el camino más corto, este camino pasa a la siguiente casilla “PASO 1” se marca como visitada y se anulan las casillas siguientes.



A	B	C	D	E	F	G	H	I	J
	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5	PASO 6	PASO 7	PASO 8
A	(0, A)	X	X	X	X	X	X	X	X
B	(6, A)	(6, A)	X	X	X	X	X	X	X
C	INFINITY								
D	(10, A)								
E	INFINITY								
F	INFINITY								
G	(8, A)								
H	INFINITY								
I	INFINITY								

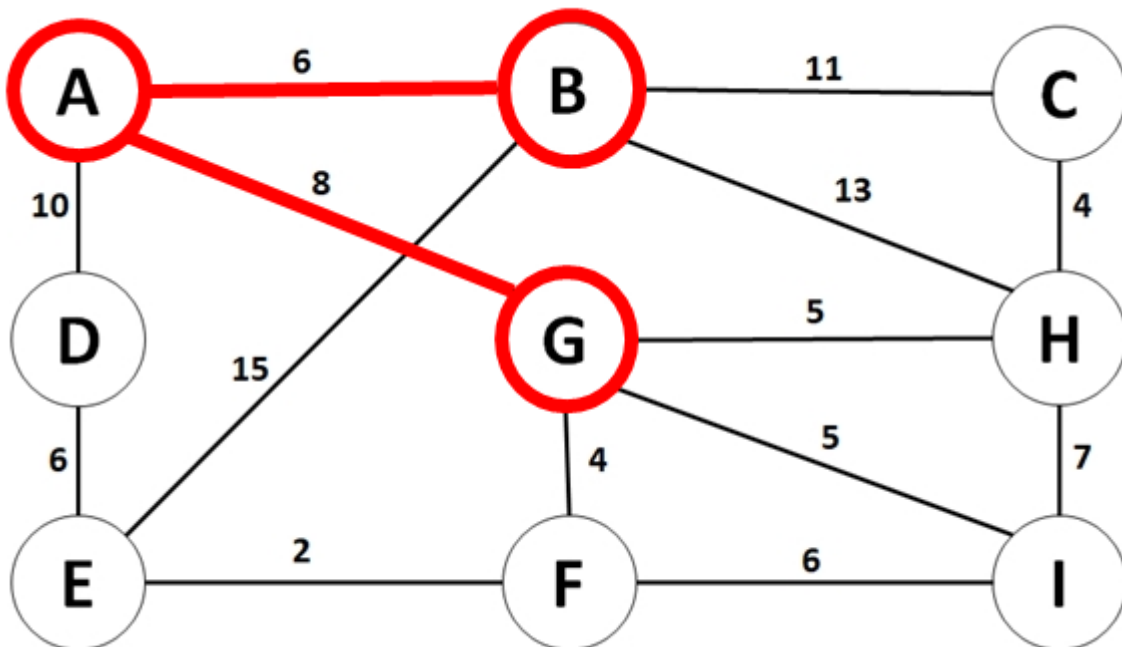
4:

Se recalculan las distancias menores desde el nodo B recordando que tenemos que tenemos una distancia acumulada de 6. (Advertencia hay que ignorar todas las casillas anuladas y visitadas)

A	B	C	D	E	F	G	H	I	J
	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5	PASO 6	PASO 7	PASO 8
A	(0, A)	X	X	X	X	X	X	X	X
B	(6, A)	(6, A)	X	X	X	X	X	X	X
C	INFINITY	(17, B)							
D	(10, A)	(10, A)							
E	INFINITY	(21, B)							
F	INFINITY	INFINITY							
G	(8, A)	(8, A)							
H	INFINITY	(19, B)							
I	INFINITY	INFINITY							

5:

De todas las distancias calculadas anteriormente se tiene que seleccionar el camino más corto, este camino pasa a la siguiente casilla “PASO 2” se marca como visitada y se anulan las casillas siguientes.



	A	B	C	D	E	F	G	H	I	J
		PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5	PASO 6	PASO 7	PASO 8
A		(0, A)	X	X	X	X	X	X	X	X
B		(6, A)	(6, A)	X	X	X	X	X	X	X
C		INFINITY	(17, B)							
D		(10, A)	(10, A)							
E		INFINITY	(21, B)							
F		INFINITY	INFINITY							
G		(8, A)	(8, A)	(8, A)	X	X	X	X	X	X
H		INFINITY	(19, B)							
I		INFINITY	INFINITY							

Advertencia: la distancia acumulada es 8 y no 6+8 debido a que nosotros ya no estamos partiendo desde B... por ello la distancia acumulada se hace desde el origen A.

En este momento nuestra distancia acumulada es 8 y ahora estamos parados en el nodo “G” por ello la nueva búsqueda debe hacerse desde G con una distancia acumulada de 8.

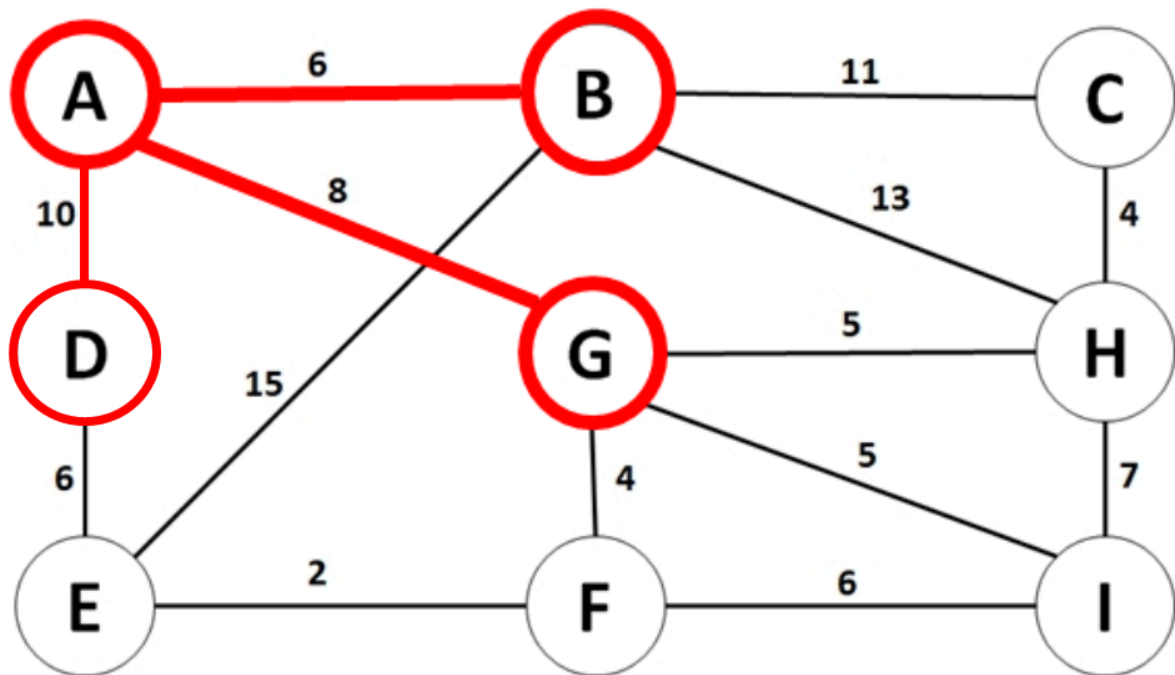
6:

Se recalculan las distancias menores desde el nodo G recordando que tenemos que tenemos una distancia acumulada de 8. (Advertencia hay que ignorar todas las casillas anuladas y visitadas)

	A	B	C	D	E	F	G	H	I	J
		PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5	PASO 6	PASO 7	PASO 8
A		(0, A)	X	X	X	X	X	X	X	X
B		(6, A)	(6, A)	X	X	X	X	X	X	X
C		INFINITY	(17, B)	(17, B)						
D		(10, A)	(10, A)	(10, A)						
E		INFINITY	(21, B)	(21, B)						
F		INFINITY	INFINITY	(12, G)						
G		(8, A)	(8, A)	(8, A)	X	X	X	X	X	X
H		INFINITY	(19, B)	(13, G)						
I		INFINITY	INFINITY	(13, G)						

7:

De todas las distancias calculadas anteriormente se tiene que seleccionar el camino más corto, este camino pasa a la siguiente casilla "PASO 3" se marca como visitada y se anulan las casillas siguientes.



	A	B	C	D	E	F	G	H	I	J
		PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5	PASO 6	PASO 7	PASO 8
A		(0, A)	X	X	X	X	X	X	X	X
B		(6, A)	(6, A)	X	X	X	X	X	X	X
C		INFINITY	(17, B)	(17, B)						
D		(10, A)	(10, A)	(10, A)	(10, A)	X	X	X	X	X
E		INFINITY	(21, B)	(21, B)						
F		INFINITY	INFINITY	(12, G)						
G		(8, A)	(8, A)	(8, A)	X	X	X	X	X	X
H		INFINITY	(19, B)	(13, G)						
I		INFINITY	INFINITY	(13, G)						

Advertencia: la distancia acumulada es 10 y el nodo donde estamos parados va a ser D

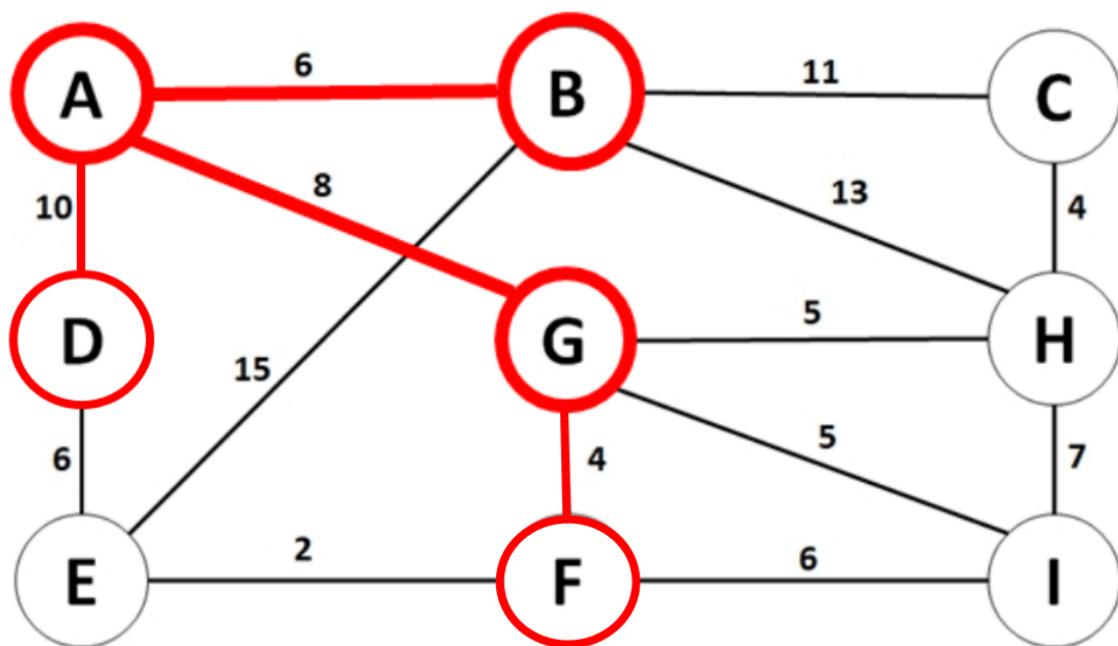
8:

Se recalculan las distancias menores desde el nodo D recordando que tenemos que tenemos una distancia acumulada de 10. (Advertencia hay que ignorar todas las casillas anuladas y visitadas)

	A	B	C	D	E	F	G	H	I	J
		PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5	PASO 6	PASO 7	PASO 8
A		(0, A)	X	X	X	X	X	X	X	X
B		(6, A)	(6, A)	X	X	X	X	X	X	X
C		INFINITY	(17, B)	(17, B)	(17, B)					
D		(10, A)	(10, A)	(10, A)	(10, A)	X	X	X	X	X
E		INFINITY	(21, B)	(21, B)	(16, D)					
F		INFINITY	INFINITY	(12, G)	(12, G)					
G		(8, A)	(8, A)	(8, A)	X	X	X	X	X	X
H		INFINITY	(19, B)	(13, G)	(13, G)					
I		INFINITY	INFINITY	(13, G)	(13, G)					

9:

De todas las distancias calculadas anteriormente se tiene que seleccionar el camino más corto, este camino pasa a la siguiente casilla "PASO 4" se marca como visitada y se anulan las casillas siguientes.



	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5	PASO 6	PASO 7
A	(0, A)	X	X	X	X	X	X	X
B	(6, A)	(6, A)	X	X	X	X	X	X
C	INFINITY	(17, B)	(17, B)	(17, B)				
D	(10, A)	(10, A)	(10, A)	(10, A)	X	X	X	X
E	INFINITY	(21, B)	(21, B)	(16, D)				
F	INFINITY	INFINITY	(12, G)	(12, G)	(12, G)	X	X	X
G	(8, A)	(8, A)	(8, A)	X	X	X	X	X
H	INFINITY	(19, B)	(13, G)	(13, G)				
I	INFINITY	INFINITY	(13, G)	(13, G)				

Advertencia: la distancia acumulada es 12 y el nodo donde estamos parados va a ser F.

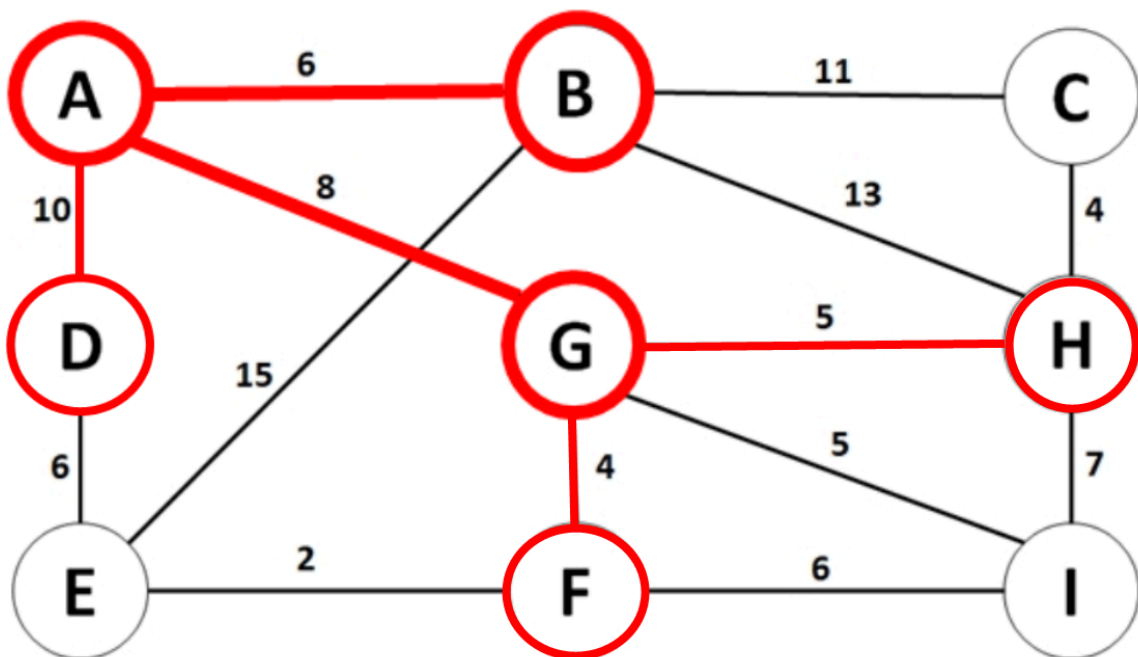
10:

Se recalculan las distancias menores desde el nodo F recordando que tenemos que tenemos una distancia acumulada de 12. (Advertencia hay que ignorar todas las casillas anuladas y visitadas)

	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5	PASO 6	PASO 7
A	(0, A)	X	X	X	X	X	X	X
B	(6, A)	(6, A)	X	X	X	X	X	X
C	INFINITY	(17, B)	(17, B)	(17, B)	(17, B)			
D	(10, A)	(10, A)	(10, A)	(10, A)	X	X	X	X
E	INFINITY	(21, B)	(21, B)	(16, D)	(14, F)			
F	INFINITY	INFINITY	(12, G)	(12, G)	(12, G)	X	X	X
G	(8, A)	(8, A)	(8, A)	X	X	X	X	X
H	INFINITY	(19, B)	(13, G)	(13, G)	(13, G)			
I	INFINITY	INFINITY	(13, G)	(13, G)	(13, G)			

11:

De todas las distancias calculadas anteriormente se tiene que seleccionar el camino más corto, este camino pasa a la siguiente casilla "PASO 5" se marca como visitada y se anulan las casillas siguientes. Para este caso tenemos el infortunio de tener 2 valores repetidos entonces lo que se hace es tomar el primer menor camino... osea que nos vamos para H.



	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5	PASO 6	PASO 7
A	(0, A)	X	X	X	X	X	X	X
B	(6, A)	(6, A)	X	X	X	X	X	X
C	INFINITY	(17, B)	(17, B)	(17, B)	(17, B)			
D	(10, A)	(10, A)	(10, A)	(10, A)	X	X	X	X
E	INFINITY	(21, B)	(21, B)	(16, D)	(14, F)			
F	INFINITY	INFINITY	(12, G)	(12, G)	(12, G)	X	X	X
G	(8, A)	(8, A)	(8, A)	X	X	X	X	X
H	INFINITY	(19, B)	(13, G)	(13, G)	(13, G)	(13, G)	X	X
I	INFINITY	INFINITY	(13, G)	(13, G)	(13, G)			

Y nuestra distancia acumulada será 13.

12:

Se recalculan las distancias menores desde el nodo H recordando que tenemos que tenemos una distancia acumulada de 13. (Advertencia hay que ignorar todas las casillas anuladas y visitadas)

A	B	C	D	E	F	G	H	I
	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5	PASO 6	PASO 7
A	(0, A)	X	X	X	X	X	X	X
B	(6, A)	(6, A)	X	X	X	X	X	X
C	INFINITY	(17, B)	(17, B)	(17, B)	(17, B)	(17, B)		
D	(10, A)	(10, A)	(10, A)	(10, A)	X	X	X	X
E	INFINITY	(21, B)	(21, B)	(16, D)	(14, F)	(14, F)		
F	INFINITY	INFINITY	(12, G)	(12, G)	(12, G)	X	X	X
G	(8, A)	(8, A)	(8, A)	X	X	X	X	X
H	INFINITY	(19, B)	(13, G)	(13, G)	(13, G)	(13, G)	X	X
I	INFINITY	INFINITY	(13, G)	(13, G)	(13, G)	(13, G)		

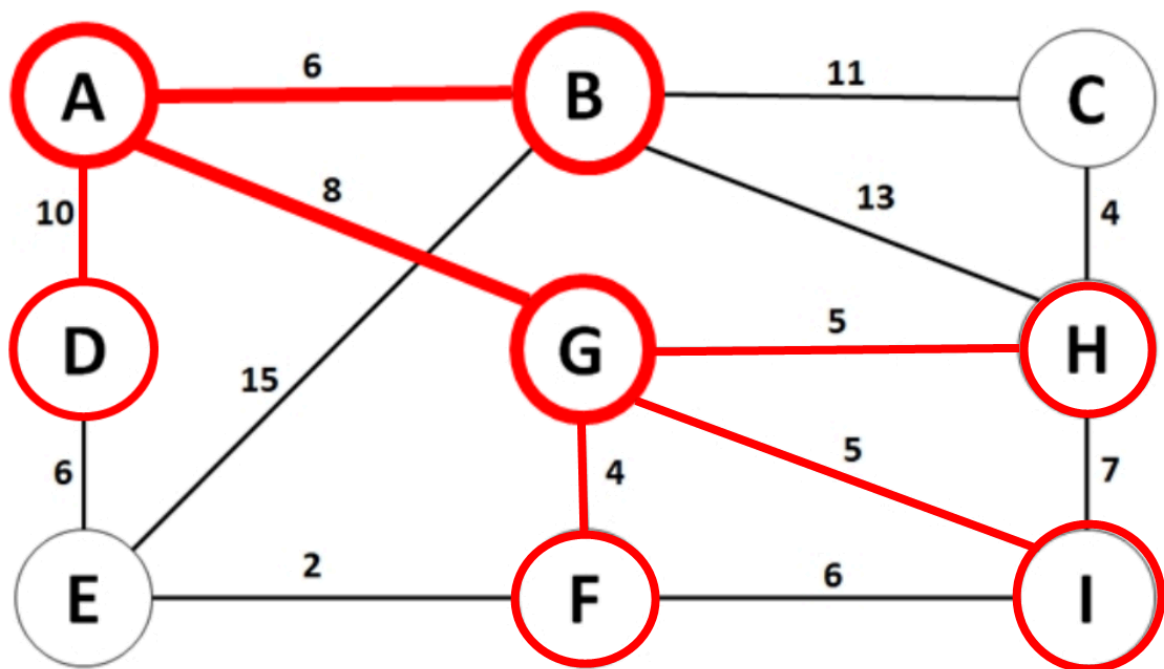
Para este caso tuvimos una ocurrencia muy interesante:

1 - al querer actualizar la casilla "Paso 5" fila "C" usted notará que si hacemos el recorrido A,G,H,C da un total de 17 lo cual es exactamente igual al 17 que viene desde A,B,C y entonces la tabla NO SE ACTUALIZA

"Solo se actualiza en caso de que la distancia es menor sino se deja como estaba"

13:

De todas las distancias calculadas anteriormente se tiene que seleccionar el camino más corto, este camino pasa a la siguiente casilla "PASO 6" se marca como visitada y se anulan las casillas siguientes.



	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5	PASO 6	PASO 7
A	(0, A)	X	X	X	X	X	X	X
B	(6, A)	(6, A)	X	X	X	X	X	X
C	INFINITY	(17, B)	(17, B)	(17, B)	(17, B)	(17, B)		
D	(10, A)	(10, A)	(10, A)	(10, A)	X	X	X	X
E	INFINITY	(21, B)	(21, B)	(16, D)	(14, F)	(14, F)		
F	INFINITY	INFINITY	(12, G)	(12, G)	(12, G)	X	X	X
G	(8, A)	(8, A)	(8, A)	X	X	X	X	X
H	INFINITY	(19, B)	(13, G)	(13, G)	(13, G)	(13, G)	X	X
I	INFINITY	INFINITY	(13, G)	(13, G)	(13, G)	(13, G)	(13, G)	X

Y nuestra distancia acumulada será 13.

14:

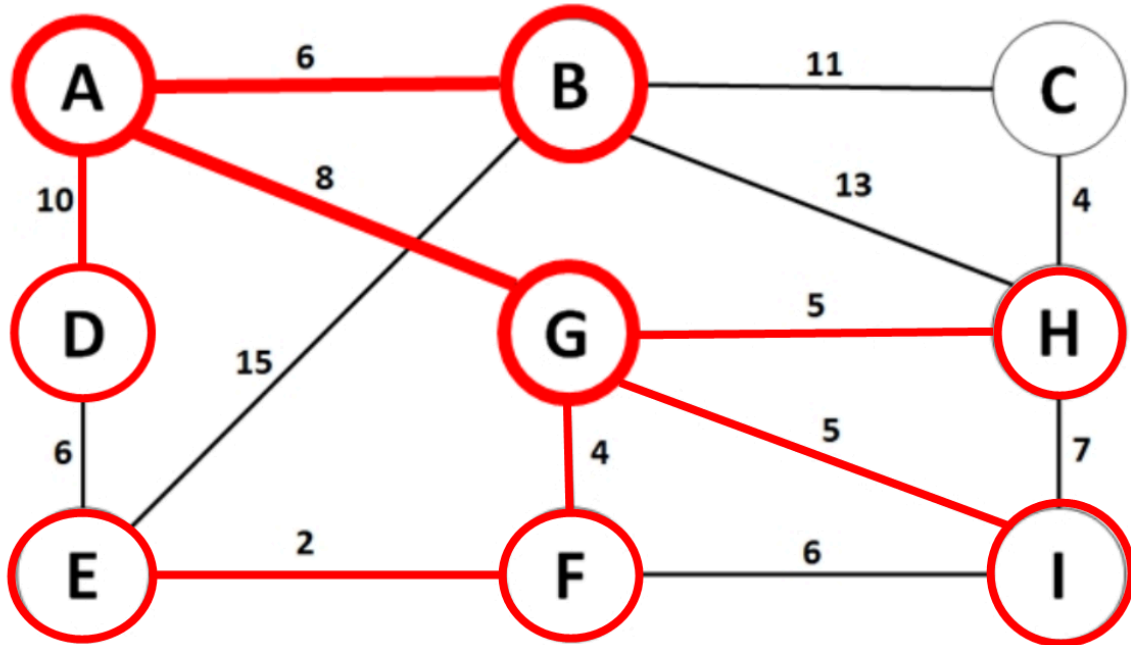
Se recalculan las distancias menores desde el nodo I recordando que tenemos que tenemos una distancia acumulada de 13. (Advertencia hay que ignorar todas las casillas anuladas y visitadas)

	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5	PASO 6	PASO 7
A	(0, A)	X	X	X	X	X	X	X
B	(6, A)	(6, A)	X	X	X	X	X	X
C	INFINITY	(17, B)	(17, B)	(17, B)	(17, B)	(17, B)	(17, B)	
D	(10, A)	(10, A)	(10, A)	(10, A)	X	X	X	X
E	INFINITY	(21, B)	(21, B)	(16, D)	(14, F)	(14, F)	(14, F)	
F	INFINITY	INFINITY	(12, G)	(12, G)	(12, G)	X	X	X
G	(8, A)	(8, A)	(8, A)	X	X	X	X	X
H	INFINITY	(19, B)	(13, G)	(13, G)	(13, G)	(13, G)	X	X
I	INFINITY	INFINITY	(13, G)	(13, G)	(13, G)	(13, G)	(13, G)	X

15:



De todas las distancias calculadas anteriormente se tiene que seleccionar el camino más corto, este camino pasa a la siguiente casilla “PASO 7” se marca como visitada y se anulan las casillas siguientes.



	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5	PASO 6	PASO 7	PASO 8
A	(0, A)	X	X	X	X	X	X	X	X
B	(6, A)	(6, A)	X	X	X	X	X	X	X
C	INFINITY	(17, B)	(17, B)	(17, B)	(17, B)	(17, B)	(17, B)		
D	(10, A)	(10, A)	(10, A)	(10, A)	X	X	X	X	X
E	INFINITY	(21, B)	(21, B)	(16, D)	(14, F)	(14, F)	(14, F)	(14, F)	X
F	INFINITY	INFINITY	(12, G)	(12, G)	(12, G)	X	X	X	X
G	(8, A)	(8, A)	(8, A)	X	X	X	X	X	X
H	INFINITY	(19, B)	(13, G)	(13, G)	(13, G)	(13, G)	X	X	
I	INFINITY	INFINITY	(13, G)	(13, G)	(13, G)	(13, G)	(13, G)	X	X

Y nuestra distancia acumulada será 14.

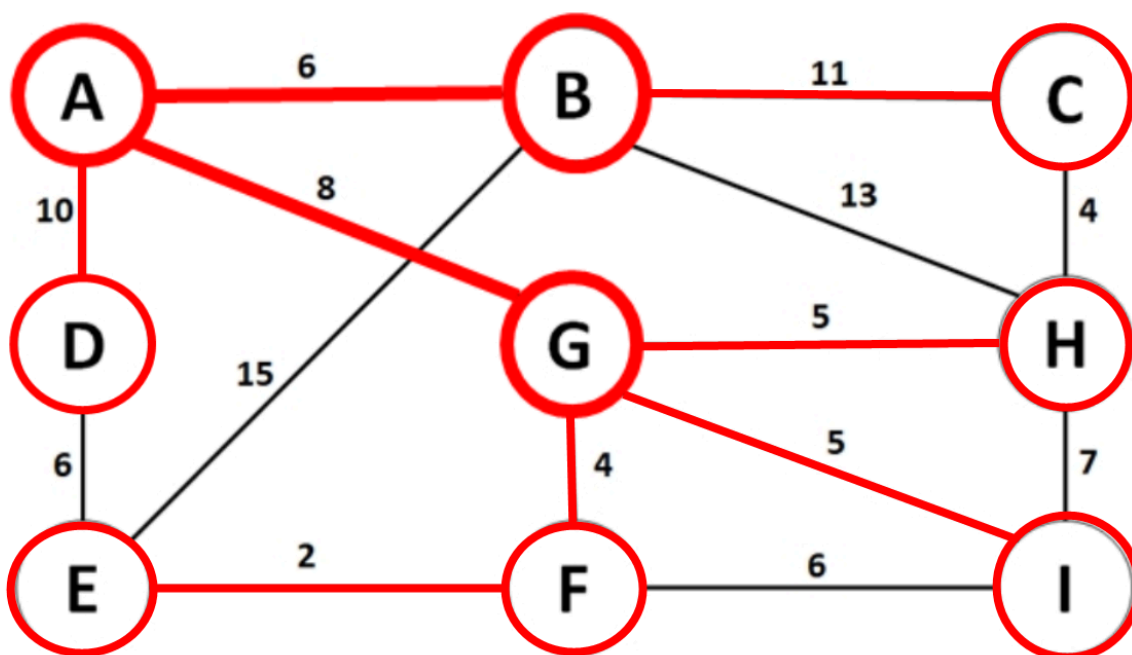
16:

Se recalculan las distancias menores desde el nodo E recordando que tenemos que tenemos una distancia acumulada de 14. (Advertencia hay que ignorar todas las casillas anuladas y visitadas)

	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5	PASO 6	PASO 7	PASO 8
A	(0, A)	X	X	X	X	X	X	X	X
B	(6, A)	(6, A)	X	X	X	X	X	X	X
C	INFINITY	(17, B)	(17, B)	(17, B)	(17, B)	(17, B)	(17, B)	(17, B)	
D	(10, A)	(10, A)	(10, A)	(10, A)	X	X	X	X	X
E	INFINITY	(21, B)	(21, B)	(16, D)	(14, F)	(14, F)	(14, F)	(14, F)	X
F	INFINITY	INFINITY	(12, G)	(12, G)	(12, G)	X	X	X	X
G	(8, A)	(8, A)	(8, A)	X	X	X	X	X	X
H	INFINITY	(19, B)	(13, G)	(13, G)	(13, G)	(13, G)	X	X	X
I	INFINITY	INFINITY	(13, G)	(13, G)	(13, G)	(13, G)	(13, G)	X	X

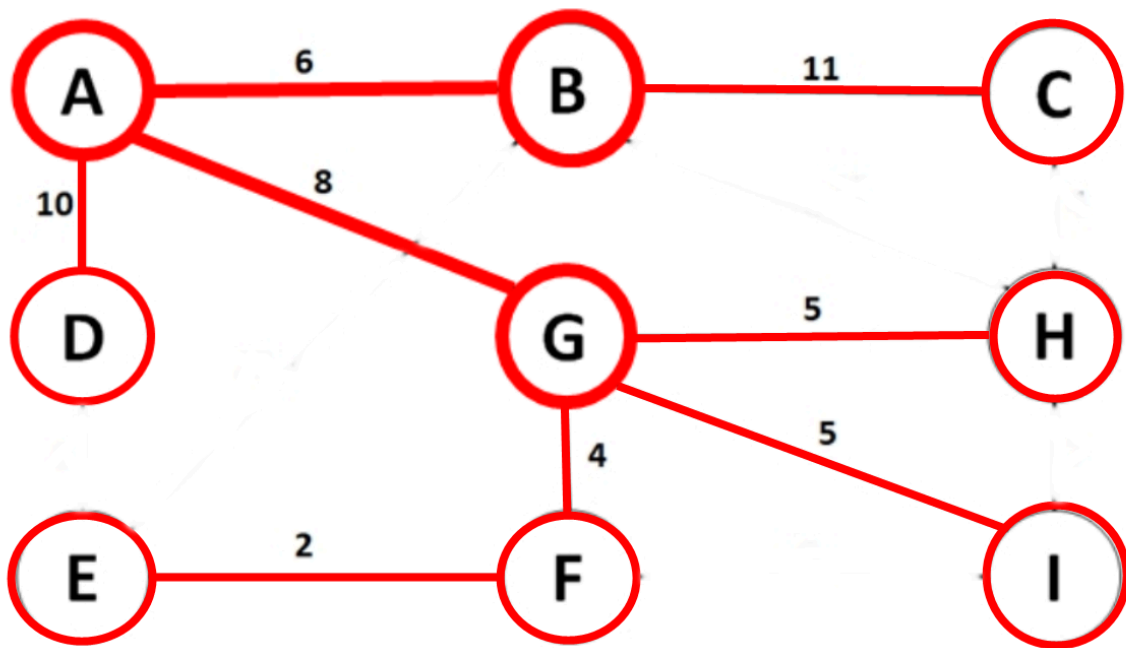
17:

De todas las distancias calculadas anteriormente se tiene que seleccionar el camino más corto, este camino pasa a la siguiente casilla “PASO 8” final.



	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5	PASO 6	PASO 7	PASO 8
A	(0, A)	X	X	X	X	X	X	X	X
B	(6, A)	(6, A)	X	X	X	X	X	X	X
C	INFINITY	(17, B)	(17, B)	(17, B)	(17, B)	(17, B)	(17, B)	(17, B)	(17, B)
D	(10, A)	(10, A)	(10, A)	(10, A)	X	X	X	X	X
E	INFINITY	(21, B)	(21, B)	(16, D)	(14, F)	(14, F)	(14, F)	(14, F)	X
F	INFINITY	INFINITY	(12, G)	(12, G)	(12, G)	X	X	X	X
G	(8, A)	(8, A)	(8, A)	X	X	X	X	X	X
H	INFINITY	(19, B)	(13, G)	(13, G)	(13, G)	(13, G)	X	X	X
I	INFINITY	INFINITY	(13, G)	(13, G)	(13, G)	(13, G)	(13, G)	X	X

Al final quedará el siguiente grafo:



## ¿Cómo se viaja de un nodo a otro?

Si usted pudo notar la tabla tiene demasiada información que no nos interesa:

	PASO 0	PASO 1	PASO 2	PASO 3	PASO 4	PASO 5	PASO 6	PASO 7	PASO 8
A	(0, A)	X	X	X	X	X	X	X	X
B	(6, A)	(6, A)	X	X	X	X	X	X	X
C	INFINITY	(17, B)	(17, B)	(17, B)	(17, B)	(17, B)	(17, B)	(17, B)	(17, B)
D	(10, A)	(10, A)	(10, A)	(10, A)	X	X	X	X	X
E	INFINITY	(21, B)	(21, B)	(16, D)	(14, F)	(14, F)	(14, F)	(14, F)	X
F	INFINITY	INFINITY	(12, G)	(12, G)	(12, G)	X	X	X	X
G	(8, A)	(8, A)	(8, A)	X	X	X	X	X	X
H	INFINITY	(19, B)	(13, G)	(13, G)	(13, G)	(13, G)	X	X	X
I	INFINITY	INFINITY	(13, G)	(13, G)	(13, G)	(13, G)	(13, G)	X	X

Al final la única información que nos interesa para poder realizar un recorrido son los nodos marcados.

La única información que me interesa es la siguiente:

```
{'A': (0, 'A'), 'B': (6, 'A'), 'G': (8, 'A'), 'D': (10, 'A'), 'F': (12, 'G'), 'H': (13, 'G'), 'I': (13, 'G'), 'E': (14, 'F'), 'C': (17, 'B'), None: (inf, None)}
```

Para observar más bonito:

A:(0, A)	B:(6, A)	G:(8, A)	D:(10, A)	F:(12, G)	H:(13, G)	I:(13, G)	E:(14, F)	C:(17, B)
----------	----------	----------	-----------	-----------	-----------	-----------	-----------	-----------

Entonces para ir de un punto a otro hay que tener en cuenta lo siguiente:

- Se tiene que calcular el algoritmo nuevamente por cada vez que cambia el origen (Dijkstra solo funciona para un único nodo de origen en caso de necesitar cambiar el

origen hay que recalcular).

- La ruta que marca de un origen a un destino está invertida. Por ejemplo si quiero ir de A,B lo que me entregará es la ruta B,A.

### ***Ejemplo ir de A al punto H***

lo primero que tenemos que hacer es fijarnos en la lista de candidatos:

A:(0, A)	B:(6, A)	G:(8, A)	D:(10, A)	F:(12, G)	H:(13, G)	I:(13, G)	E:(14, F)	C:(17, B)
----------	----------	----------	-----------	-----------	-----------	-----------	-----------	-----------

Y nos paramos en el destino osea H:

**H:(13, G)**

H contiene el valor de 13... por lo tanto el costo total para ir de A hasta H es 13... y tengo que hacerlo pasando por G osea en este momento tenemos 2 nodos H que es el destino y G que es el previo al destino.

Y nos paramos en G y lo observamos en la tabla:

**G:(8, A)**

G contiene un valor de 8 pero es irrelevante por que en el paso anterior ya se calculo la distancia total pero nos dice que aquí termina el cálculo de la ruta A >> H debido a que tenemos que pasar por A y este es el origen.

Respuesta: para ir de A hasta H la ruta es **[A, G, H]**