Taller (2). Algoritmos. / 2018-1

Caminos más cortos en grafos desde una sola fuente

Universidad Nacional de Colombia

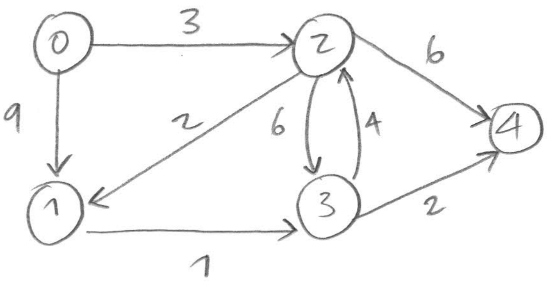
NOMBRE: JHON JAIRO MUESES Q.

CODIGO: 2879355.

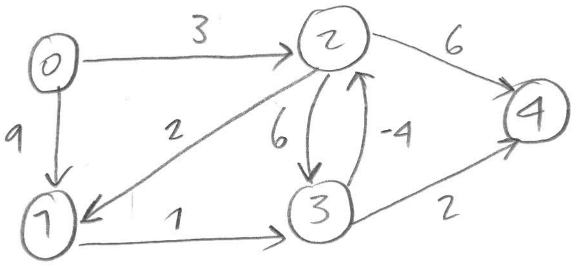
PROFESOR: GERMAN HERNANDEZ.

Considere los siguientes GRAFOS:

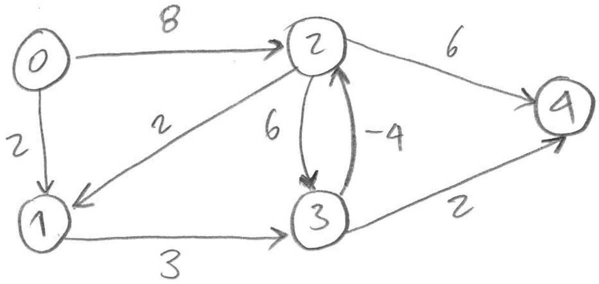
(a)



(b)



(c)



(1) para los 3 grafos, calcule (por inspección) la longitud de los caminos mas cortos desde el nodo 0 hacia el resto de nodos.

Grafo (a)		longitud	peso.
0 — 1	$0 \xrightarrow{3} 2 \xrightarrow{2} 1$	2	5
0 — 2	$0 \xrightarrow{3} 2$	1	3

grafo (a)   longitud   peso.

0 — 3 $0 \xrightarrow{3} 2 \xrightarrow{2} 1 \xrightarrow{1} 3$ 3   6

0 — 4 $0 \xrightarrow{3} 2 \xrightarrow{2} 1 \xrightarrow{1} 3 \xrightarrow{2} 4$ 4   8.

grafo (b)   longitud   peso.

0 — 1.

$$\begin{tikzcd} 0 \arrow[r, "3"] & 2 \arrow[r, "2"] & 1 \arrow[r, "1"] & 3 \arrow[ll, bend left, "-4"] \end{tikzcd}$$

∞ $-\infty$.

0 – 2

$$\begin{tikzcd} 0 \arrow[r, "3"] & 2 \arrow[r, "2"] & 1 \arrow[r, "1"] & 3 \arrow[ll, bend left, "-4"] \end{tikzcd}$$

∞ $-\infty$

0 – 3

$$\begin{tikzcd} 0 \arrow[r, "3"] & 2 \arrow[r, "2"] & 1 \arrow[r, "1"] & 3 \arrow[ll, bend left, "-4"] \end{tikzcd}$$

∞ $-\infty$

0 – 4

$$\begin{tikzcd} 0 \arrow[r, "3"] & 2 \arrow[r, "2"] & 1 \arrow[r, "1"] & 3 \arrow[ll, bend left, "-4."] \arrow[r, "2"] & 4 \end{tikzcd}$$

∞ $-\infty$.

grafo (c)

0 — 1 $0 \xrightarrow{2} 1$ 1   2

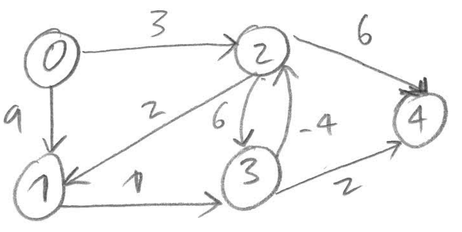
0 — 2 $0 \xrightarrow{2} 1 \xrightarrow{3} 3 \xrightarrow{-4} 2$ 3   1

0 — 3. $0 \xrightarrow{2} 1 \xrightarrow{3} 3$ 2   5

0 — 4 $0 \xrightarrow{2} 1 \xrightarrow{3} 3 \xrightarrow{2} 4$ 3   7

$0 \xrightarrow{2} 1 \xrightarrow{3} 3 \xrightarrow{-4} 2 \xrightarrow{6} 4$ 4   7

② para el grafo b, ¿ puede encontrar un camino desde $0 \to 1$ de costo 0? explique.



Se tiene que al terminar el ciclo $2 \xrightarrow{2} 1 \xrightarrow{1} 3 \xrightarrow{-4} 2$ el peso del nodo 2 disminuye en 1., Posteriormente llega el momento en que su peso será -2 y al pasar al nodo 1, el peso sera de $-2+2=0$.

Camino de costo 0.

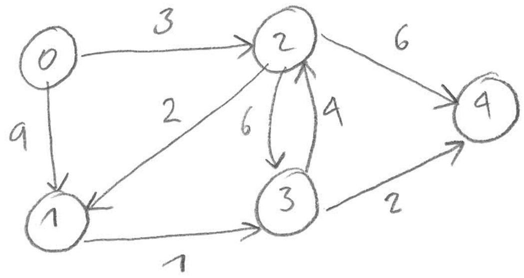
$$0 \xrightarrow{3} 2 \xrightarrow{2} 1 \xrightarrow{1} 3 \xrightarrow{-4} 2 \xrightarrow{2} 1 \xrightarrow{1} 3 \xrightarrow{-4} 2$$

$$\xrightarrow{2} 1 \xrightarrow{1} 3 \xrightarrow{-4} 2 \xrightarrow{2} 1 \xrightarrow{1} 3 \xrightarrow{-4} 2 \xrightarrow{2} 1$$

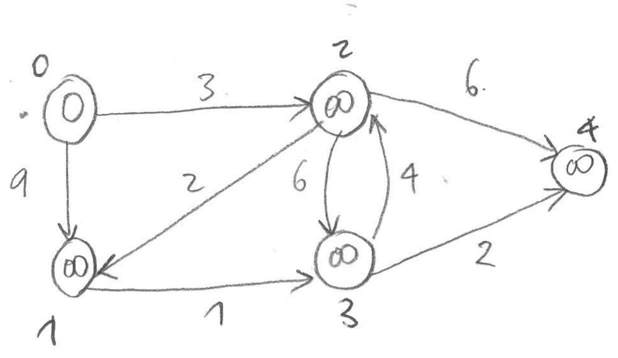
$$\xrightarrow{1} 3 \xrightarrow{-4} 2 \xrightarrow{2} 1$$

③ Aplique el algoritmo de Bellman-Ford para encontrar caminos mas cortos a los grafos a, b, c. En cada caso muestre el estado de los grafos (valores de v.d para cada nodo, puede mostrarlos dentro de cada nodo omitiendo la etiqueta del nodo) despues de cada iteración del for en la linea 2.

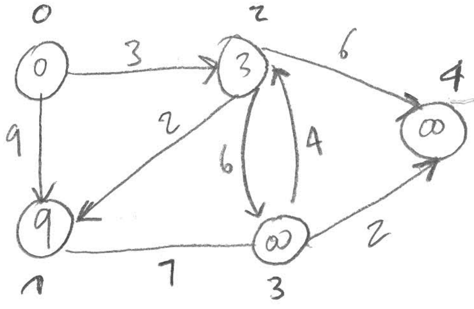
(a)

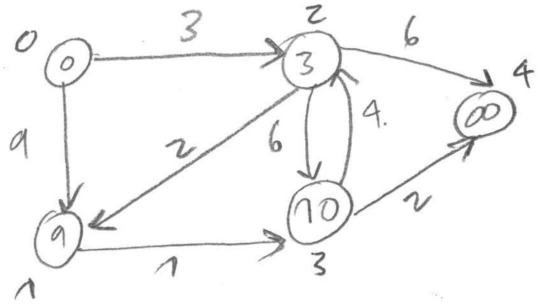
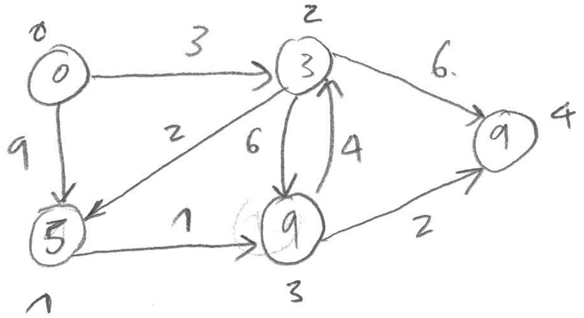


$\boxed{i=1}$



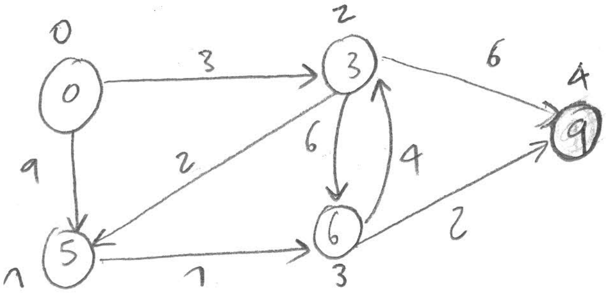
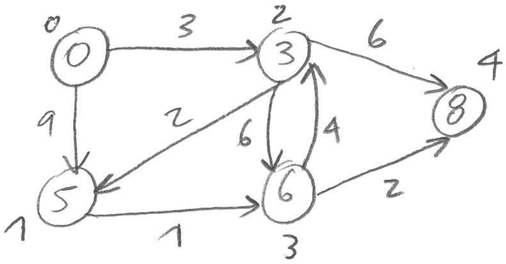
$\Rightarrow$



$\Rightarrow$  $\Rightarrow$ 

v				
0	1	2	3	4
0	5	3	9	9

$\boxed{i=2}$

 $\Rightarrow$ 

$\Rightarrow i = 3$

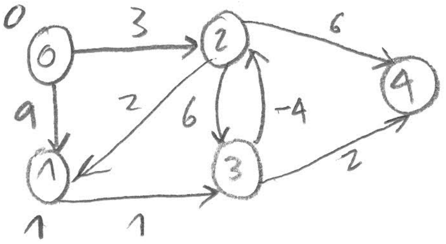
$i = 4$

$\vdots$

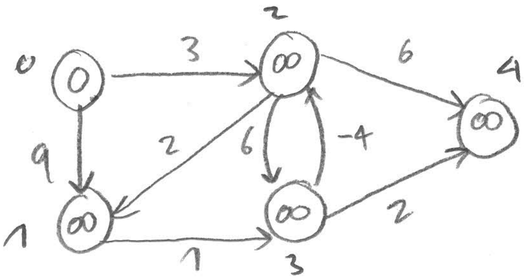
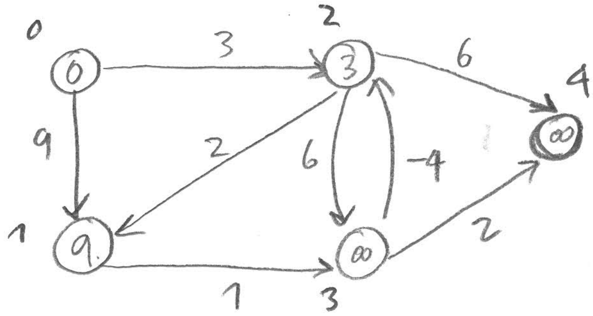
$i = |G.V| - 1 = 7$

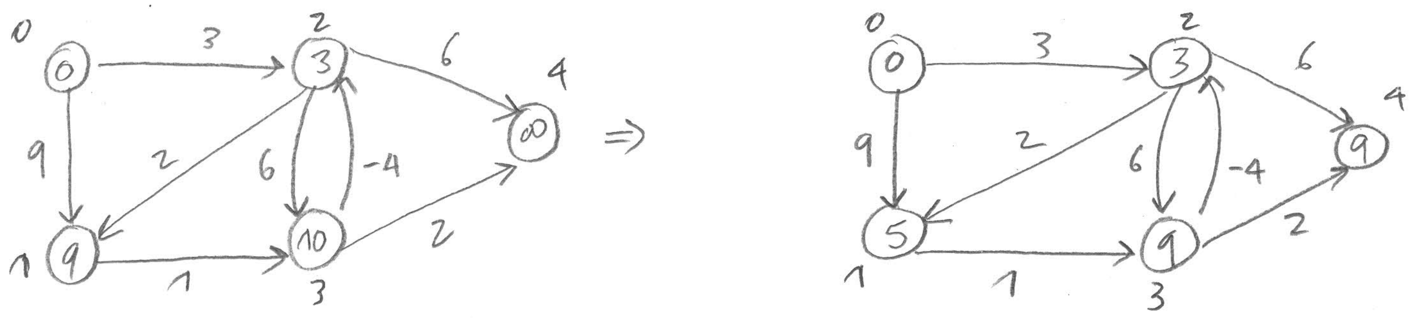
No cambia

0	1	2	3	4
0	5	3	6	8

(b) 

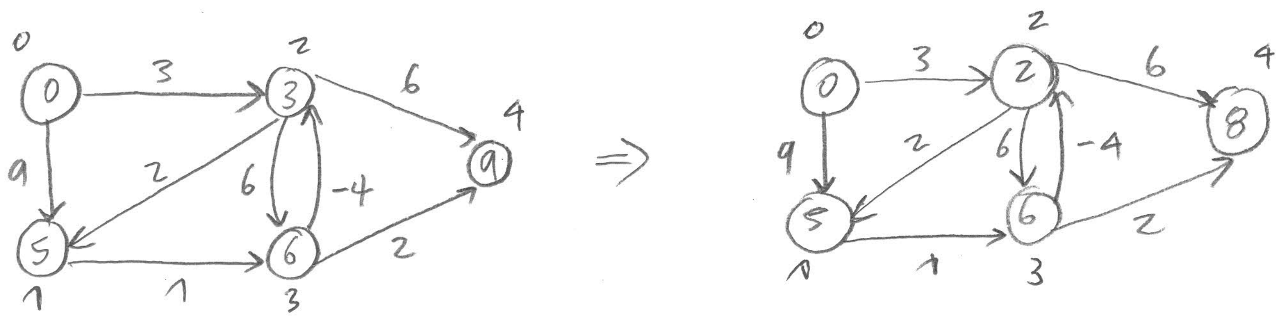
$\boxed{i=1}$

 $\Rightarrow$ 



0
1
2
3
4
3
9
2
6
-4
6
2
1
0
9
3
10
∞
⇒
0
5
3
9
9

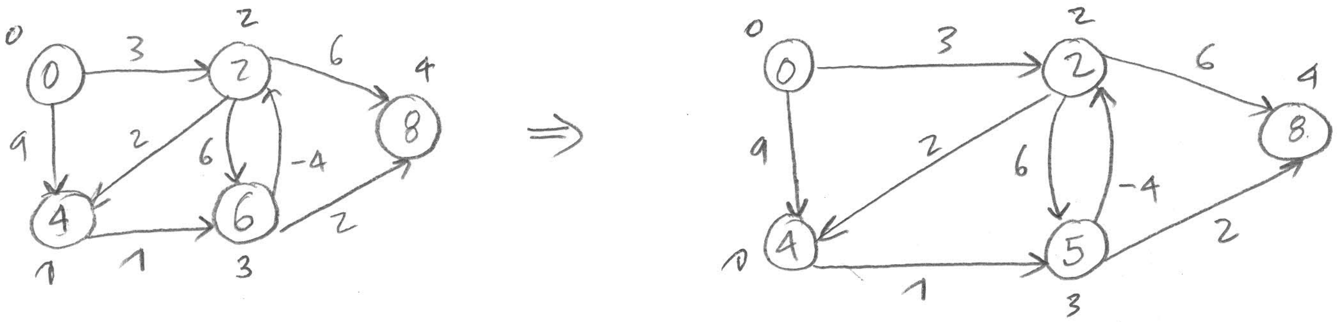
i=2



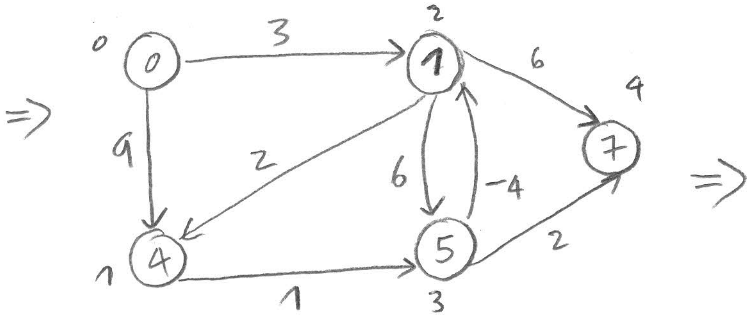
0
1
2
3
4
3
9
2
6
-4
6
2
1
0
5
3
6
9
⇒
0
5
2
6
8

i=3

i=4

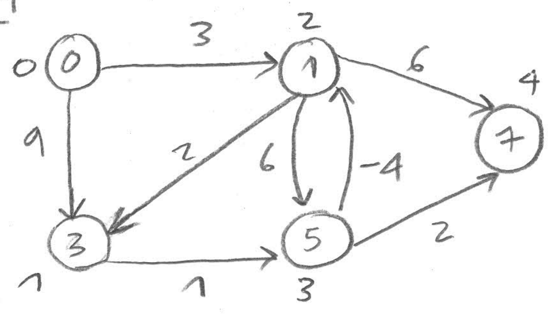


0
1
2
3
4
3
9
2
6
-4
6
2
1
0
4
2
6
8
⇒
0
4
2
5
8



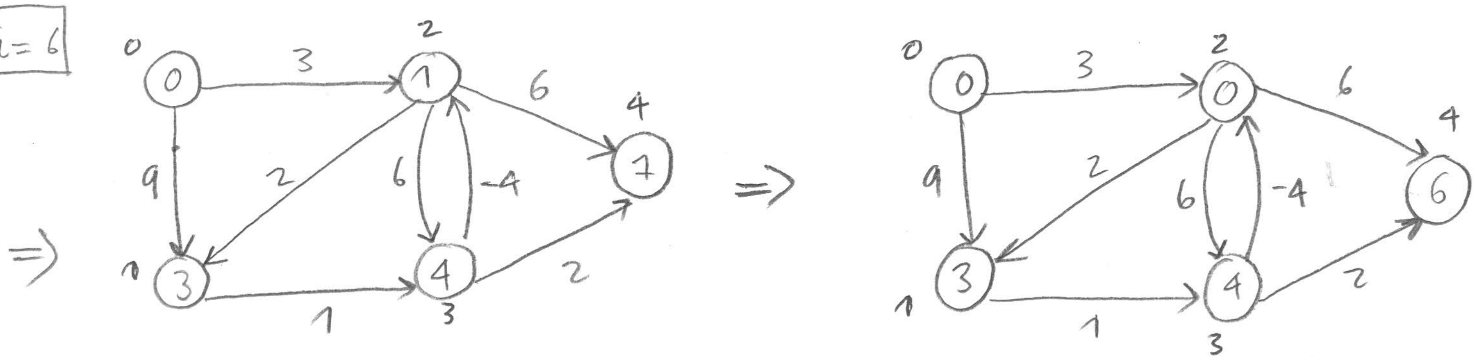
⇒
0
1
2
3
4
3
9
2
6
-4
6
2
1
0
4
1
5
7
⇒

i=5



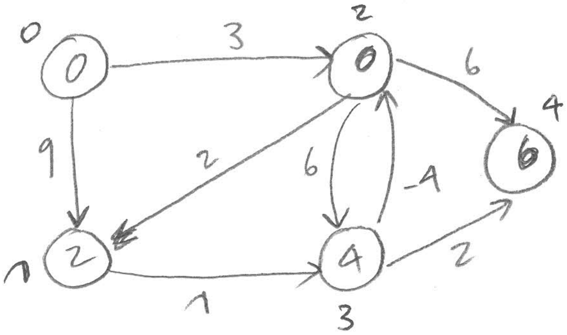
0
1
2
3
4
3
9
2
6
-4
6
2
1
0
3
1
5
7

i=6



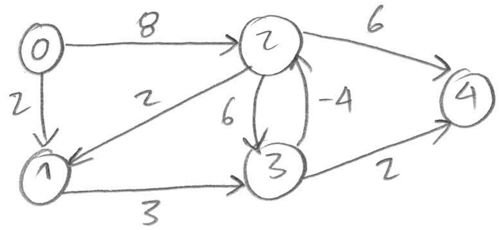
⇒
0
1
2
3
4
3
9
2
6
-4
6
2
1
0
3
1
4
7
⇒
0
3
0
4
6

$\boxed{i=7}$

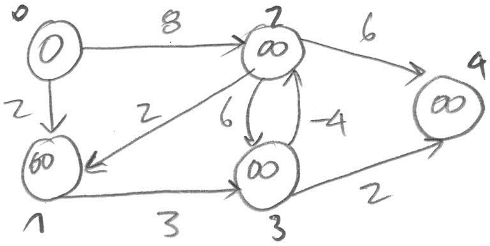
$\Rightarrow$ 

0	1	2	3	4
0	2	0	4	6

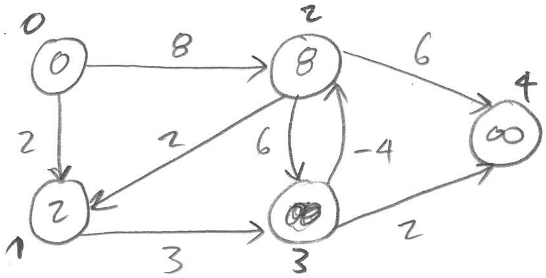
(c)

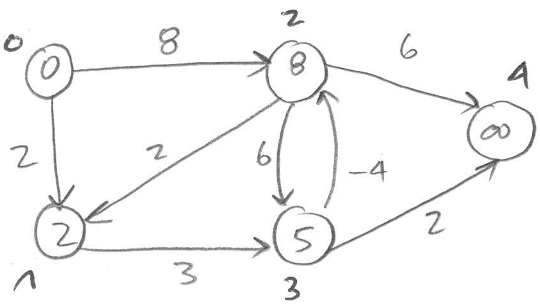


$\boxed{i=1}$

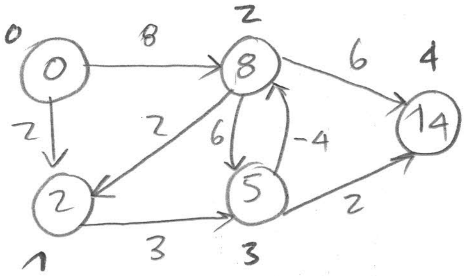


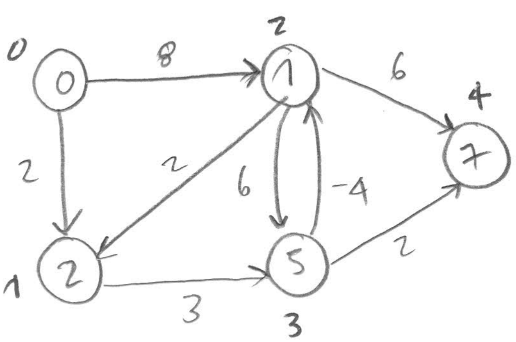
$\Rightarrow$



$\Rightarrow$ 

$\Rightarrow$



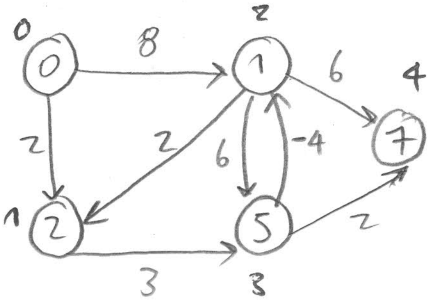
$\Rightarrow$ 

$\Rightarrow$ $i=2$

$i=3$

$\vdots$

$i=7$



No Cambia

(4) Que paso con el grafo 'b'? Explique

En el grafo b, el ciclo de peso negativo de los nodos 1, 2, 3 hace que sus pesos durante Bellman-Ford, disminuyan de forma indefinida. hasta llegar a $i=7$, Por lo tanto con su estructura final se retorna false.

(5) En cada caso, ¿Cuantas llamadas se hicieron a la función Relax? ¿puede hacerse de manera mas eficiente?

a) 48 llamadas

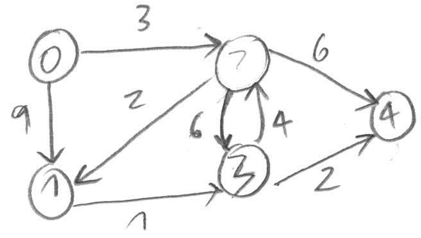
b) 48 llamadas

c) 48 llamadas

para hacerse de forma eficiente se puede dar con la implementación de una cola, la cual inicia con un unico elemento, el cual es el nodo inicial, antes de iniciar RElAX lo quitamos de la cola, y si hay un nodo afectado por RElAX, se agrega a la cola. Asi solo tengo en cuenta los nodos que han sufrido cambios y por ende producen cambros. por lo tanto solo se llama RElAX las veces necesarias para llegar a un grafo que posee los pesos minimos.

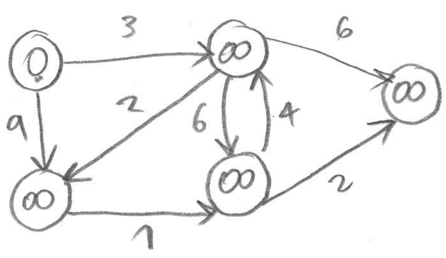
(6) para los grafos a y c, muestre una secuencia de llamadas Relax, que le permita calcular los caminos mas cortos de una manera mas eficiente.

(a)

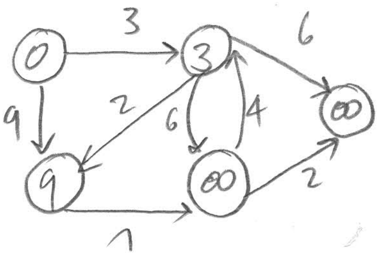


Cola

0



⇒

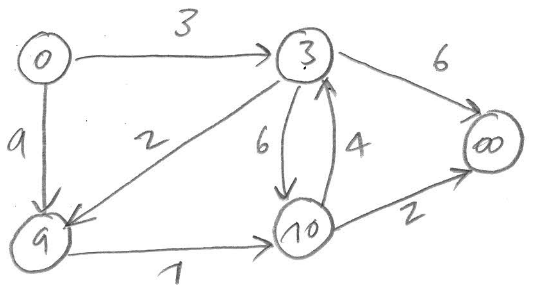


Cola.

2

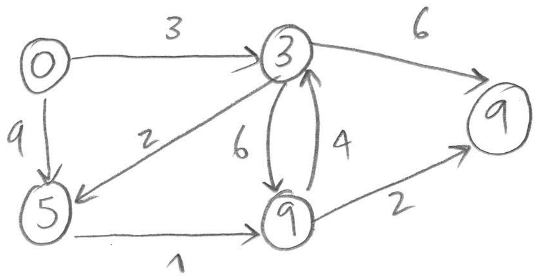
1

~~0~~



Cola

3
2
~~1~~
~~0~~



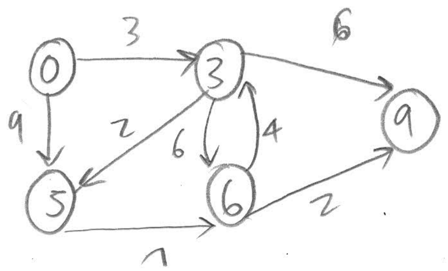
Cola

4
1
3
~~2~~
~~1~~
~~0~~

⇒ grafo continua igual ⇒

Cola.

4
1
~~3~~
~~2~~
~~1~~
~~0~~

⇒ 

Cola

3
4
~~1~~
~~3~~
~~2~~
~~1~~
~~0~~

⇒ Sin cambios

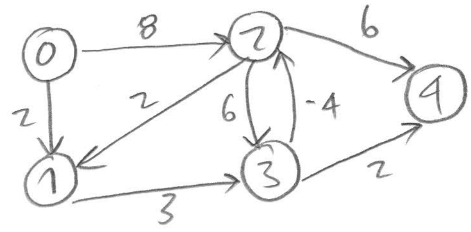
Cola

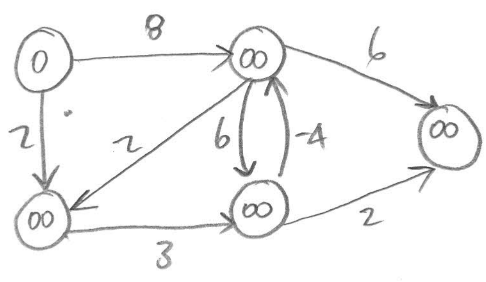
3 ⇒

Cola vacía

$\boxed{\text{Se llamó 11 veces a RELAX}}$

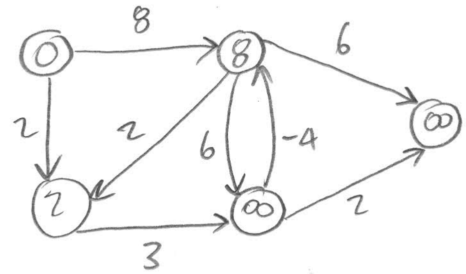
(c)





Cola

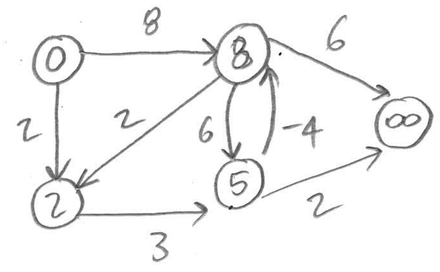
0

⇒ 

Cola

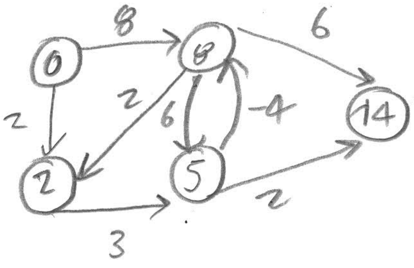
2
1
~~0~~

⇒



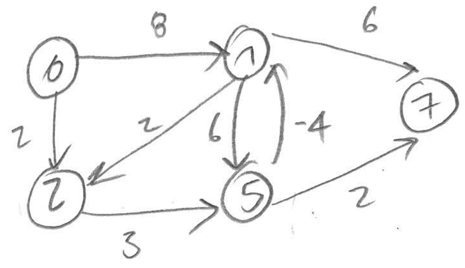
Cola
3
2
~~1~~
~~0~~

⇒



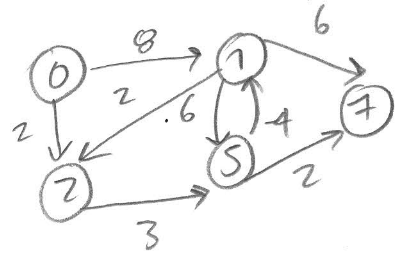
Cola.
4.
3
~~2~~
~~1~~
~~0~~

⇒



Cola
4
2
~~3~~
~~2~~
~~1~~
~~0~~

⇒



Cola
4
~~2~~
~~3~~
~~2~~
~~1~~
~~0~~

⇒ El grafo se mantiene igual ⇒ Cola Vacia.

$\boxed{\text{En este caso se llamo 11 veces a RELAX}}$