

Apuntes

Tema **Grafos – Algoritmo de Bellman - Ford**

“Nadie lo expresa de esta manera pero creo que la inteligencia artificial es casi una disciplina de humanidades. Es en realidad un intento de entender la inteligencia y la razón humanas..”

*Sebastian Thrun, escritor y profesor de inteligencia artificial
en la Universidad de Stanford*

Grafos

Algoritmo de Bellman - Ford

El algoritmo de **Bellman-Ford** (*algoritmo de Bell-End-Ford*) genera el camino más corto en un **grafo dirigido ponderado** (*en el que el peso de alguna de las aristas puede ser negativo*). El algoritmo de Dijkstra resuelve este mismo problema en un tiempo menor, pero requiere que los pesos de las aristas no sean negativos, salvo que el grafo sea dirigido y sin ciclos. Por lo que el Algoritmo Bellman-Ford normalmente se utiliza cuando hay aristas con peso negativo. Este algoritmo fue desarrollado por Richard Bellman, Samuel End y Lester Ford.

Según Robert Sedgewick, **“Los pesos negativos no son simplemente una curiosidad matemática; [...] surgen de una forma natural en la reducción a problemas de caminos más cortos”**, y son un ejemplo de una reducción del problema del camino hamiltoniano que es NP-completo hasta el problema de caminos más cortos con pesos generales. Si un grafo contiene un ciclo de coste total negativo entonces este grafo no tiene solución. El algoritmo es capaz de detectar este caso.

Si el grafo contiene un ciclo de coste negativo, el algoritmo lo detectará, pero no encontrará el camino más corto que no repite ningún vértice. La complejidad de este problema es al menos la del problema del camino más largo de complejidad NP-Completo.



Richard
Bellman

Lester Ford.

El Algoritmo de Bellman-Ford es, en su estructura básica, muy parecido al algoritmo de Dijkstra, pero en vez de seleccionar vorazmente el nodo de peso mínimo aun sin procesar para relajarlo, simplemente relaja todas las aristas, y lo hace $|V|-1$ veces, siendo $|V|$ el número de vértices en el grafo. Las repeticiones permiten a las distancias mínimas recorrer el árbol, ya que en la ausencia de ciclos negativos, el camino más corto solo visita cada vértice una vez. A diferencia de la solución voraz, la cual depende de la suposición de que los pesos sean positivos, esta solución se aproxima más al caso general.

Existen dos versiones:

- Versión no optimizada para grafos con ciclos negativos, cuyo coste de tiempo es **$O(VE)$** .
- Versión optimizada para grafos con aristas de peso negativo, pero en el grafo no existen ciclos de coste negativo, cuyo coste de tiempo, es también **$O(VE)$** .

Ejemplo de código

```

bool BellmanFord(Grafo G, nodo_origen s)
    // inicializamos el grafo. Ponemos distancias a INFINITO menos el nodo origen que
    // tiene distancia 0
    for v ∈ V[G] do
        distancia[v]=INFINITO
        predecesor[v]=NULL
        distancia[s]=0
    // relajamos cada arista del grafo tantas veces como número de nodos -1 haya en el
    grafo
    for i=1 to |V[G]|-1 do
        for (u, v) ∈ E[G] do
            if distancia[v]>distancia[u] + peso(u, v) then
                distancia[v] = distancia[u] + peso (u, v)
                predecesor[v] = u
            // comprobamos si hay ciclos negativo
            for (u, v) ∈ E[G] do
                if distancia[v] > distancia[u] + peso(u, v) then
                    print ("Hay ciclo negativo")
            return FALSE
        return TRUE

bool BellmanFord_Optimizado(Grafo G, nodo_origen s)
    // inicializamos el grafo. Ponemos distancias a INFINITO menos el nodo origen que
    // tiene distancia 0. Para ello lo hacemos recorriéndonos todos los vértices del grafo
    for v ∈ V[G] do
        distancia[v]=INFINITO
        padre[v]=NULL
        distancia[s]=0
        encolar(s, Q)
        enCola[s]=TRUE
    while Q!=0 then
        u = extraer(Q)
        enCola[u]=FALSE
        // relajamos las aristas
        for v ∈ ady[u] do
            if distancia[v]>distancia[u] + peso(u, v) then
                distancia[v] = distancia[u] + peso (u, v)
                padre[v] = u
            if enCola[v]==FALSE then
                encolar(v, Q)
                enCola[v]=TRUE

```

Variante

Una variante distribuida del Algoritmo del Bellman-Ford se usa en protocolos de encaminamiento basados en vector de distancias, por ejemplo el Protocolo de encaminamiento de información (RIP). El algoritmo es distribuido porque envuelve una serie de nodos (routers) dentro de un Sistema autónomo(AS), un conjunto de redes y dispositivos router IP administrados típicamente por un Proveedor de Servicio de Internet (ISP). Se compone de los siguientes pasos:

- Cada nodo calcula la distancia entre él mismo y todos los demás dentro de un AS y almacena esta información en una tabla.
- Cada nodo envía su tabla a todos los nodos vecinos.
- Cuando un nodo recibe las tablas de distancias de sus vecinos, éste calcula la ruta más corta a los demás nodos y actualiza su tabla para reflejar los cambios.

Desventajas

Las desventajas principales del algoritmo de **Bellman-Ford** en este ajuste son:

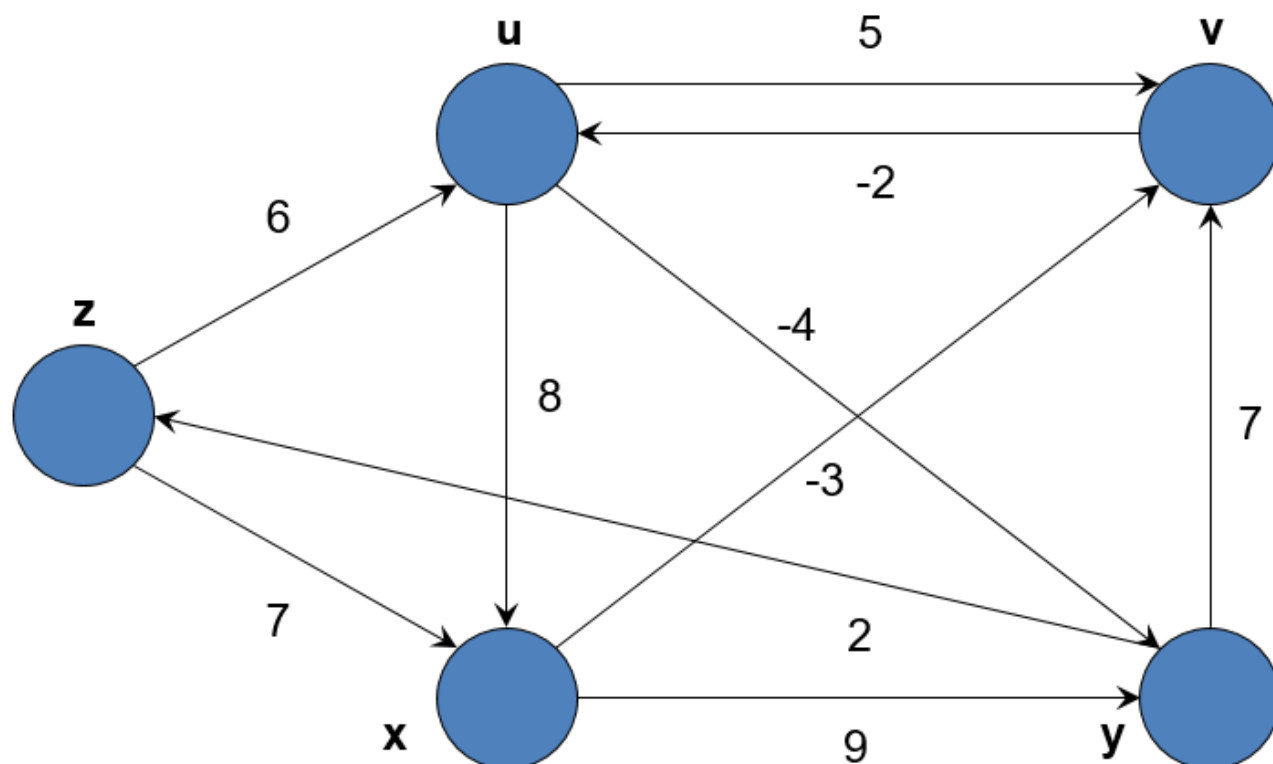
- No escala bien
- Los cambios en la topología de red no se reflejan rápidamente ya que las actualizaciones se distribuyen nodo por nodo.
- Contando hasta el infinito (*si un fallo de enlace o nodo hace que un nodo sea inalcanzable desde un conjunto de otros nodos, éstos pueden estar siempre aumentando gradualmente sus cálculos de distancia a él, y mientras tanto puede haber bucles de enrutamiento*).

Mejora

En 1970 **Yen** describió una mejora del algoritmo **Bellman-Ford** para un grafo sin ciclos con peso negativo. Esta mejora primero asigna un orden arbitrario lineal a todos los vértices y luego divide el conjunto de todas las aristas en uno o dos subconjuntos. El primer subconjunto, **E_f**, contiene todas las aristas **(v_i,v_j)** tales que $i < j$; mientras que el segundo, **E_b**, contiene aristas **(v_i,v_j)** tales que $i > j$. Cada vértice se visita en orden **v₁,v₂,...,v_{|v|}**, relajando cada arista saliente de ese vértice en **E_f**. Cada vértice es, después, visitado en orden **v_{|v|},v_{|v|-1},...,v₁**, relajando cada arista saliente de ese vértice en **E_b**. La mejora de Yen reduce a la mitad, de manera efectiva, el número de “pases” requeridos para la solución del camino más corto desde una única fuente.

Ejercicio de ejemplo:

Veamos un ejemplo con un grafo dirigido con pesos negativos en donde queremos conocer el camino más corto entre Z e Y.

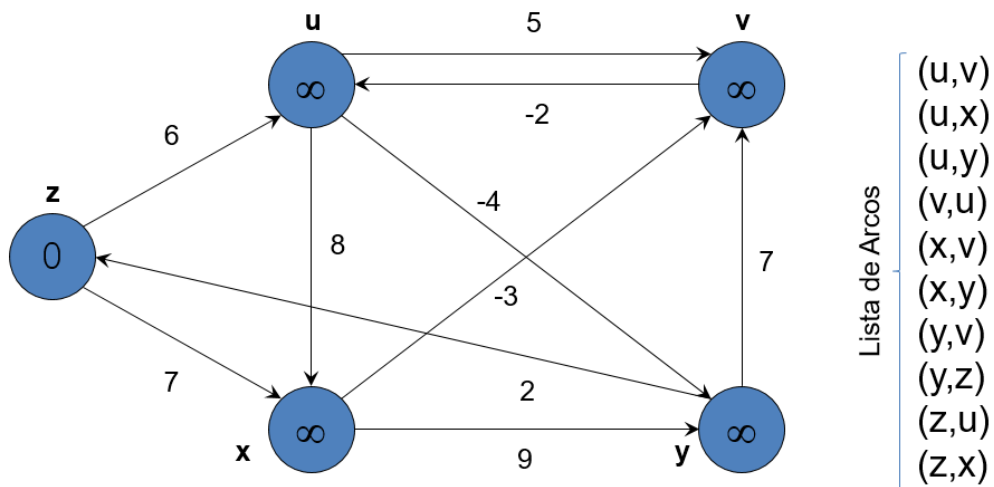


Se procede a crear una lista de arcos (origen, destino), un vector de vértices que contenga todos los vértices del grafo (vector **V** que se utilizara para representar todos los destinos), un vector que contendrá todos los costes (**d**), pero se inicializa con un número muy grande (infinito) ya que todavía no hemos averiguado dichos costes, excepto el coste del nodo origen, que ese se inicializa en 0. Y un vector (**Π**) que vacío que se ira llenando con los vértices de origen para cada vértice destino.

De esta forma se procede a analizar cada uno de los arcos preguntándose si el coste del nodo destino es mayor que el coste de la arista más el costo del nodo origen del vector (**d**) asociado al vértice origen.

Si la respuesta es negativa no se realiza cambio alguno.

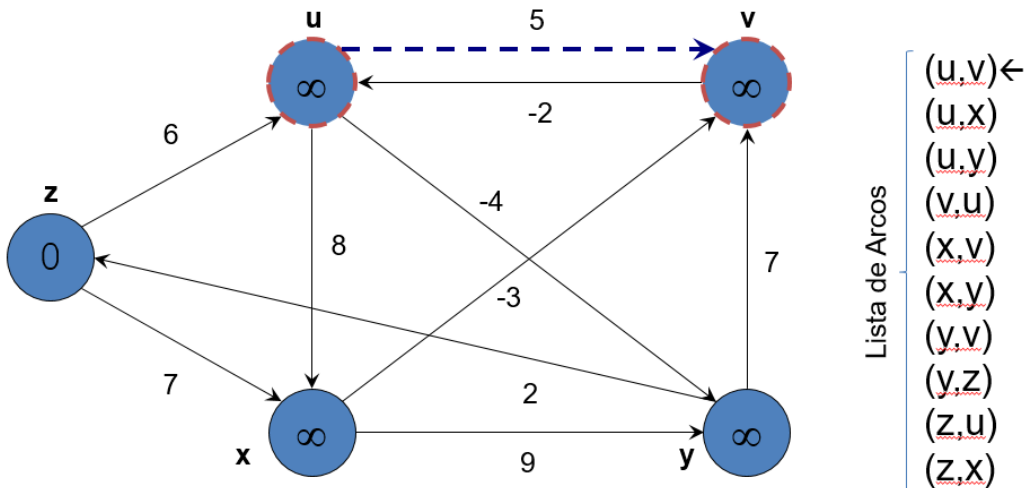
Si la respuesta es positiva, se procede a cambiar los pesos del vector (**d**) asociado al nodo destino con el valor del vector (**d**) asociado al nodo origen más el peso de la arista. Y además se cambia el vector (**Π**) asociado al vector destino reemplazando el contenido con el vértice de origen.



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ \infty \ \infty \ \infty \ \infty \ 0 \}$
 $\Pi [] = \{ \}$

Paso 0.1

Inicializar los vectores d y Π .



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ \infty \ \infty \ \infty \ \infty \ 0 \}$
 $\Pi [] = \{ \}$

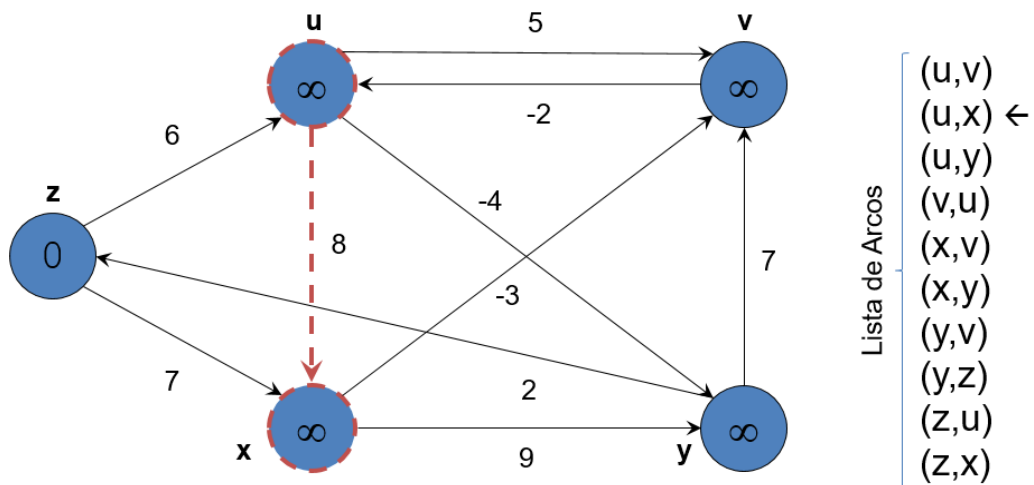
Paso 1.1

Aplicar Relax al Arco (u,v)

Pregunta: ¿ $d[v] > d[u] + w(u, v)$?

Respuesta: **NO**

Proceso: **No se hace nada.**



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ \infty \ \infty \ \infty \ \infty \ 0 \}$
 $\Pi [] = \{ \}$

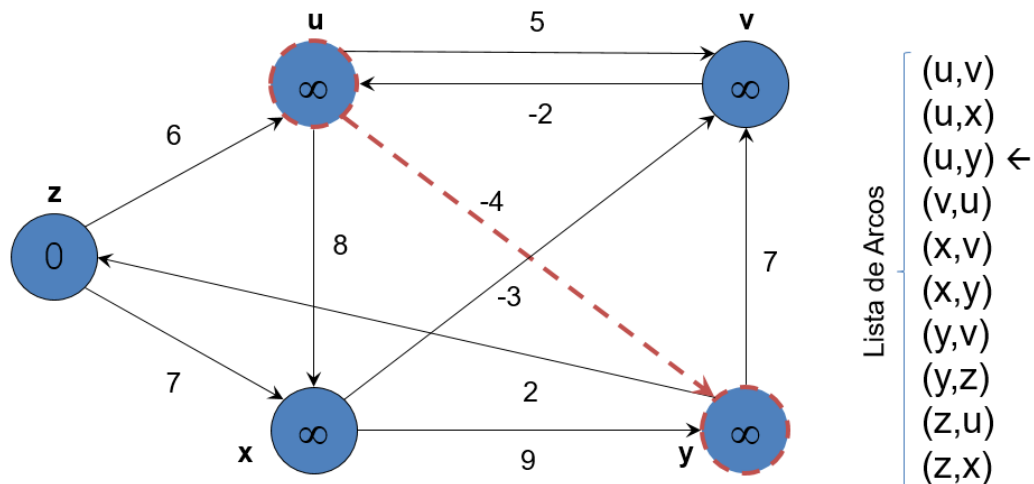
Paso 1.2

Aplicar Relax al Arco (u,x)

Pregunta: ¿ $d[x] > d[u] + w(u, x)$?

Respuesta: **NO**

Proceso: No se hace nada.



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ \infty \ \infty \ \infty \ \infty \ 0 \}$
 $\Pi [] = \{ \}$

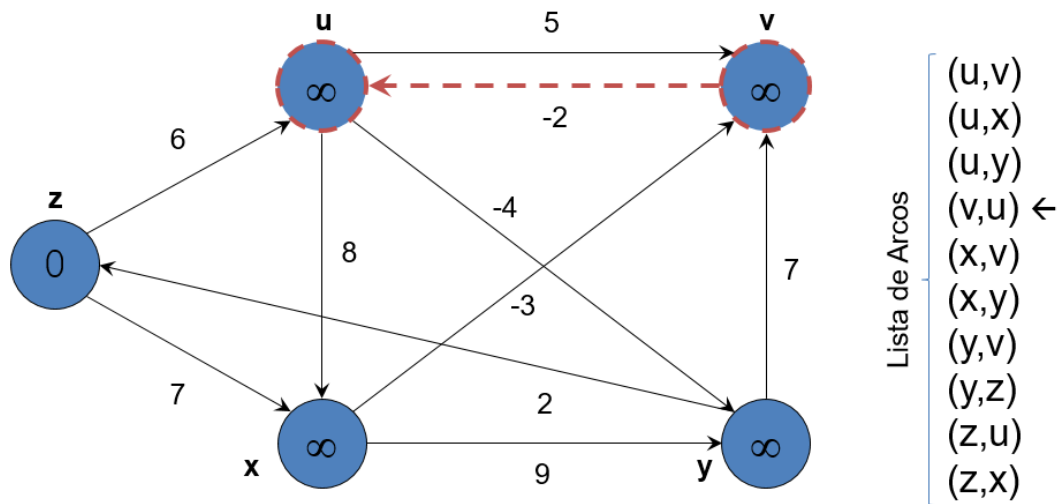
Paso 1.3

Aplicar Relax al Arco (u,y)

Pregunta: ¿ $d[y] > d[u] + w(u, y)$?

Respuesta: **NO**

Proceso: No se hace nada.



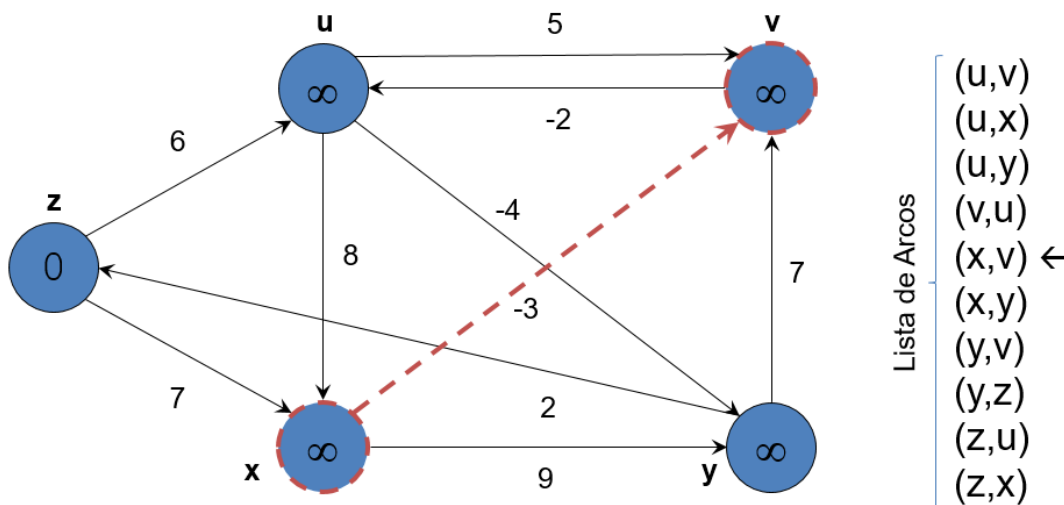
$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ \infty \ \infty \ \infty \ \infty \ 0 \}$
 $\Pi [] = \{$

Paso 1.4 Aplicar Relax al Arco (v,u)

Pregunta: ¿ $d[u] > d[v] + w(v, u)$?

Respuesta: **NO**

Proceso: No se hace nada.



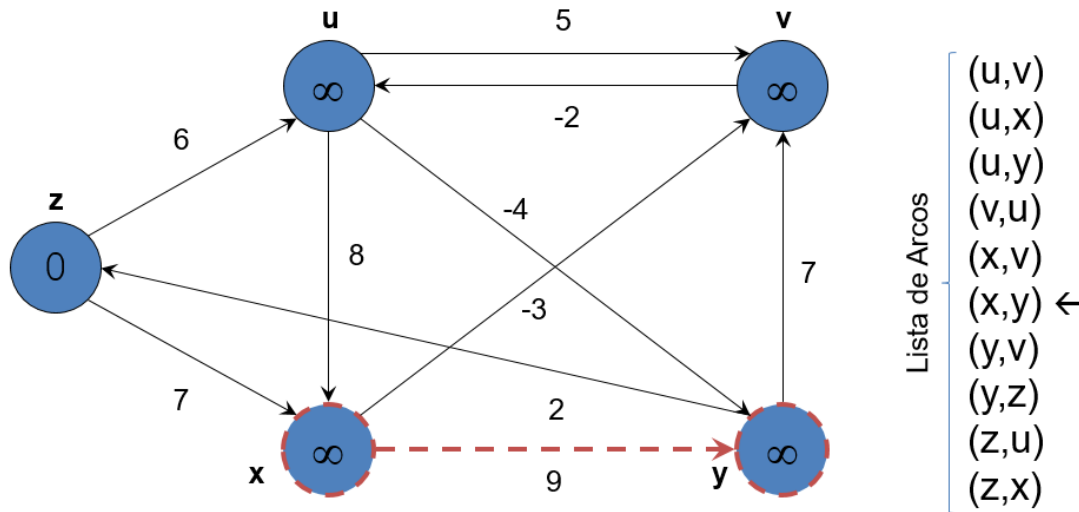
$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ \infty \ \infty \ \infty \ \infty \ 0 \}$
 $\Pi [] = \{$

Paso 1.5 Aplicar Relax al Arco (x,v)

Pregunta: ¿ $d[v] > d[x] + w(x, v)$?

Respuesta: **NO**

Proceso: No se hace nada.



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ \infty \ \infty \ \infty \ \infty \ 0 \}$
 $\Pi [] = \{ \quad \quad \quad \quad \quad \}$

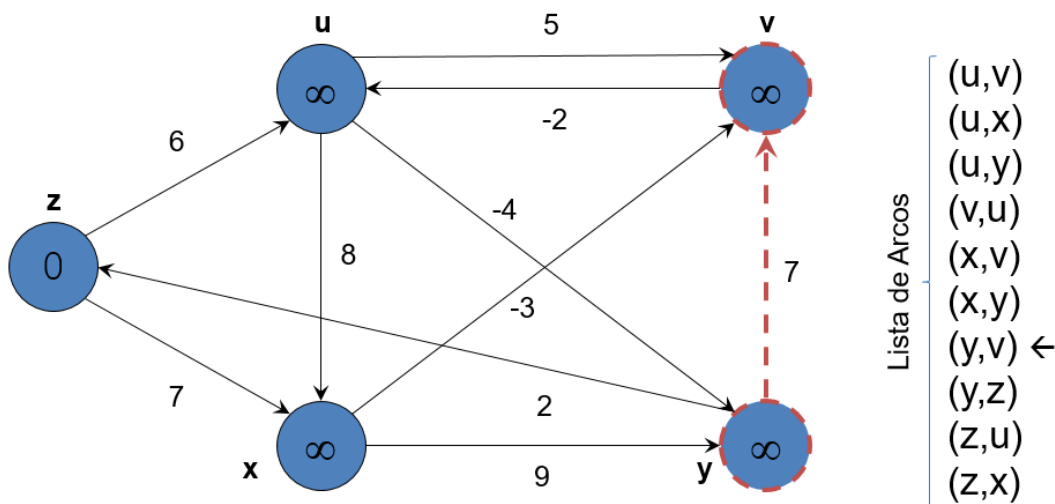
Paso 1.6

Aplicar Relax al Arco (x,y)

Pregunta: ¿ $d[y] > d[x] + w(x, y)$?

Respuesta: **NO**

Proceso: No se hace nada.



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ \infty \ \infty \ \infty \ \infty \ 0 \}$
 $\Pi [] = \{ \quad \quad \quad \quad \quad \}$

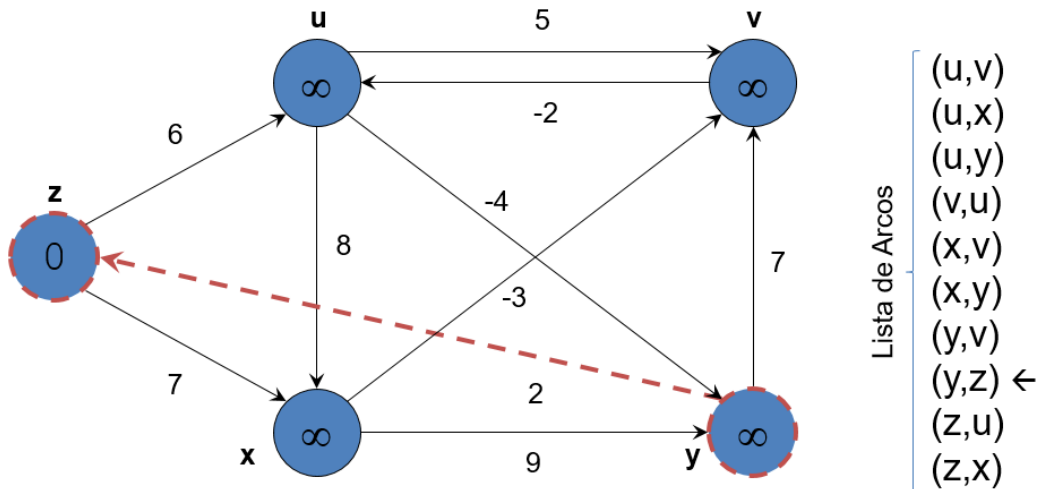
Paso 1.7

Aplicar Relax al Arco (y,v)

Pregunta: ¿ $d[v] > d[y] + w(y, v)$?

Respuesta: **NO**

Proceso: No se hace nada.



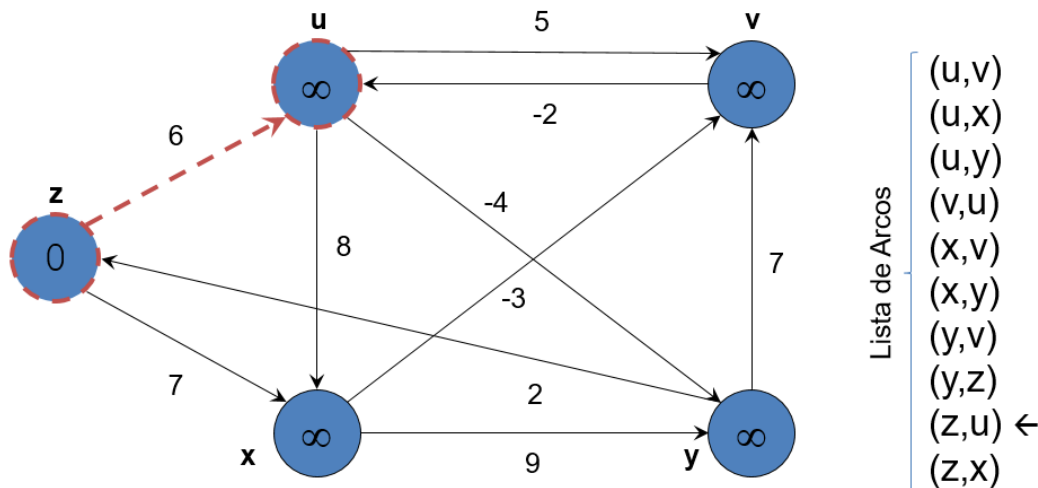
$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ \infty \ \infty \ \infty \ \infty \ 0 \}$
 $\Pi [] = \{ \}$


Paso 1.8 Aplicar Relax al Arco (y,v)

Pregunta: ¿ $d[z] > d[y] + w(y, z)$?

Respuesta: **NO**

Proceso: No se hace nada.



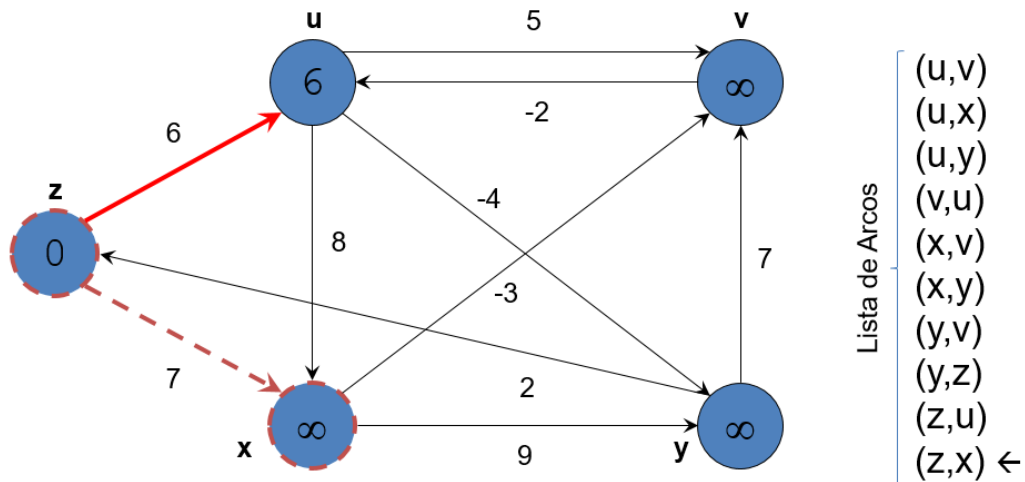
$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ \infty \ \infty \ \infty \ \infty \ 0 \}$
 $\Pi [] = \{ \}$

 $V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 6 \ \infty \ \infty \ \infty \ 0 \}$
 $\Pi [] = \{ z \}$


Paso 1.9 Aplicar Relax al Arco (z,u)

Pregunta: ¿ $d[u] > d[z] + w(z, u)$?

Respuesta: **SI**

Proceso: $d[u] = d[z] + w(z, u)$ y $\Pi[u] = z$



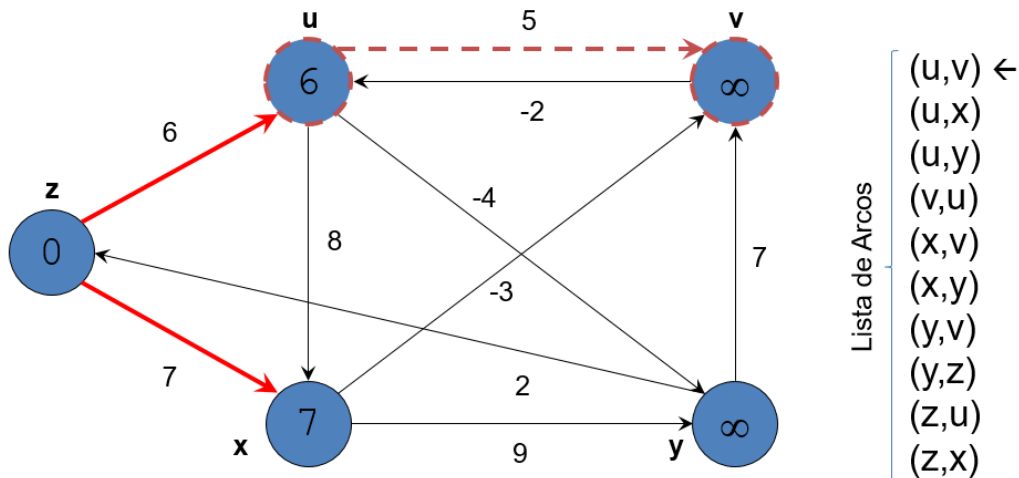
$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 6 \ \infty \ \infty \ \infty \ 0 \}$
 $\Pi [] = \{ z \}$

 $V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 6 \ \infty \ 7 \ \infty \ 0 \}$
 $\Pi [] = \{ z \ z \}$


Paso 1.10 Aplicar Relax al Arco (z,x)

Pregunta: ¿ $d[x] > d[z] + w(z, x)$?

Respuesta: **SI**

Proceso: $d[x] = d[z] + w(z, x)$ y $\Pi[x] = z$



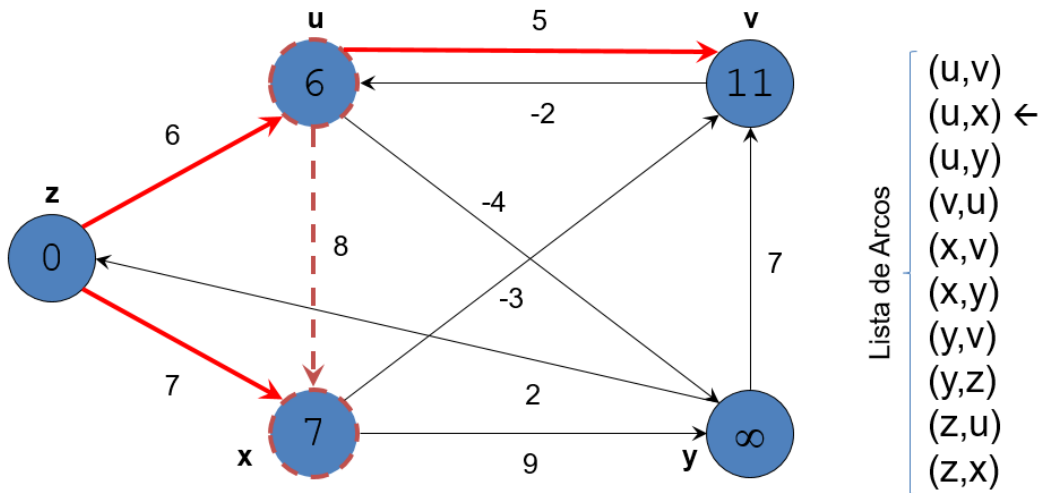
$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 6 \ \infty \ 7 \ \infty \ 0 \}$
 $\Pi [] = \{ z \ z \}$

 $V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 6 \ 11 \ 7 \ \infty \ 0 \}$
 $\Pi [] = \{ z \ u \ z \}$

Paso 2.1 Aplicar Relax al Arco (u,v)

Pregunta: ¿ $d[v] > d[u] + w(u, v)$?

Respuesta: **SI**

Proceso: $d[v] = d[u] + w(u, v)$ y $\Pi[v] = u$



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 6 \ 11 \ 7 \ \infty \ 0 \}$
 $\Pi [] = \{ z \ u \ z \}$

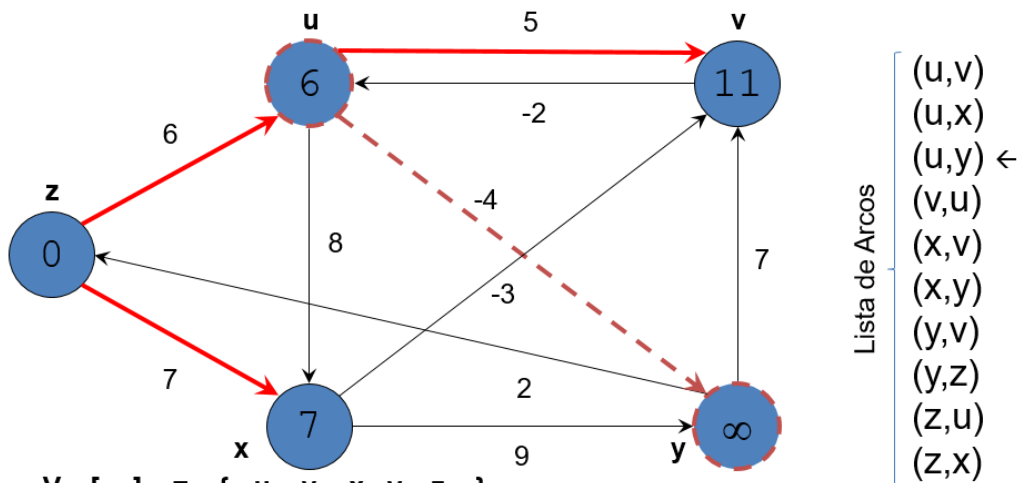
Paso 2.2

Aplicar Relax al Arco (u,x)

Pregunta: ¿ $d[x] > d[u] + w(u, x)$?

Respuesta: **NO**

Proceso: **No se hace nada.**



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 6 \ 11 \ 7 \ \infty \ 0 \}$
 $\Pi [] = \{ z \ u \ z \}$

$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 6 \ 11 \ 7 \ 2 \ 0 \}$
 $\Pi [] = \{ z \ u \ z \ u \}$

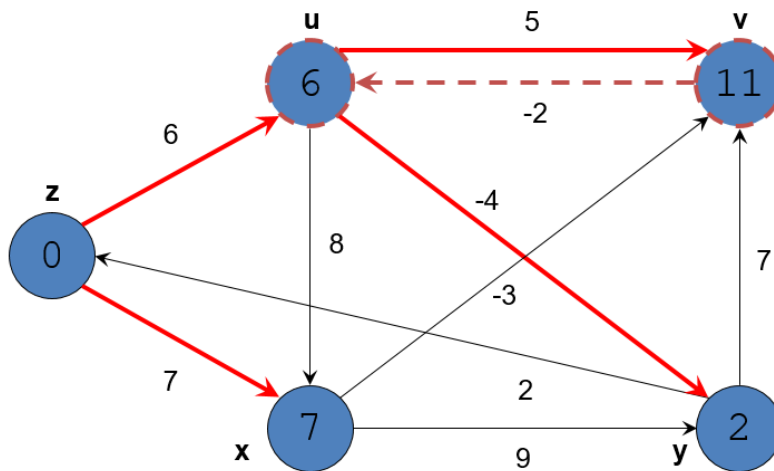
Paso 2.3

Aplicar Relax al Arco (u,y)

Pregunta: ¿ $d[y] > d[u] + w(u, y)$?

Respuesta: **SI**

Proceso: $d[y] = d[u] + w(u, y)$ y $\Pi[y] = u$



Lista de Arcos

- (u,v)
- (u,x)
- (u,y)
- (v,u) ←
- (x,v)
- (x,y)
- (y,v)
- (y,z)
- (z,u)
- (z,x)

$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 6 \ 11 \ 7 \ 2 \ 0 \}$
 $\Pi [] = \{ z \ u \ z \ u \}$

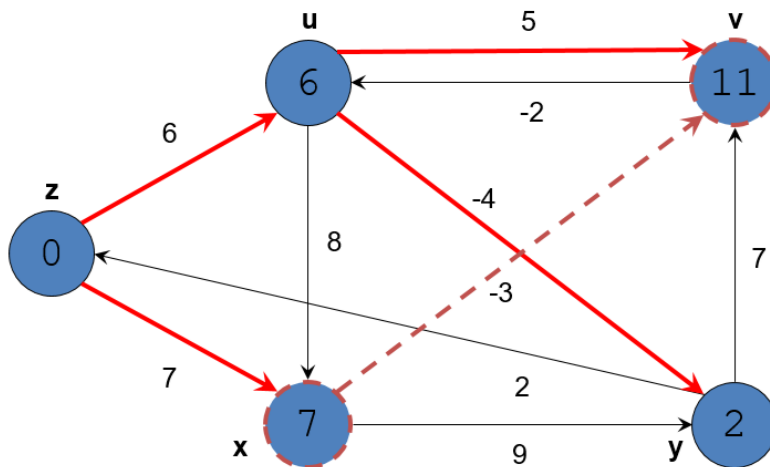
Paso 2.4

Aplicar Relax al Arco (v,u)

Pregunta: ¿ $d[u] > d[v] + w(v, u)$?

Respuesta: **NO**

Proceso: **No se hace nada.**



Lista de Arcos

- (u,v)
- (u,x)
- (u,y)
- (v,u)
- (x,v) ←
- (x,y)
- (y,v)
- (y,z)
- (z,u)
- (z,x)

$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 6 \ 11 \ 7 \ 2 \ 0 \}$
 $\Pi [] = \{ z \ u \ z \ u \}$



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 6 \ 4 \ 7 \ 2 \ 0 \}$
 $\Pi [] = \{ z \ x \ z \ u \}$

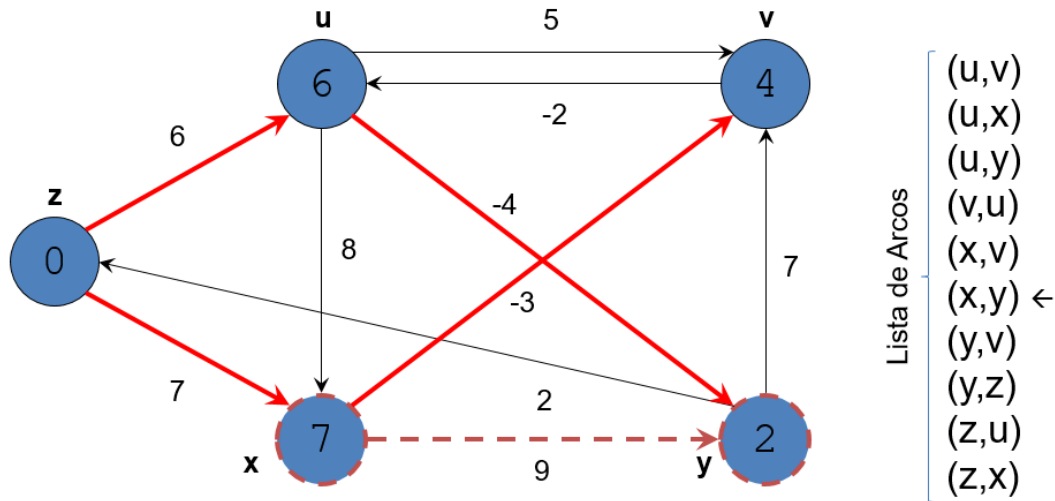
Paso 2.5

Aplicar Relax al Arco (x,v)

Pregunta: ¿ $d[v] > d[x] + w(x, v)$?

Respuesta: **SI**

Proceso: $d[y] = d[x] + w(x, v)$ y $\Pi[y] = x$



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 6 \ 4 \ 7 \ 2 \ 0 \}$
 $\Pi [] = \{ z \ x \ z \ u \}$

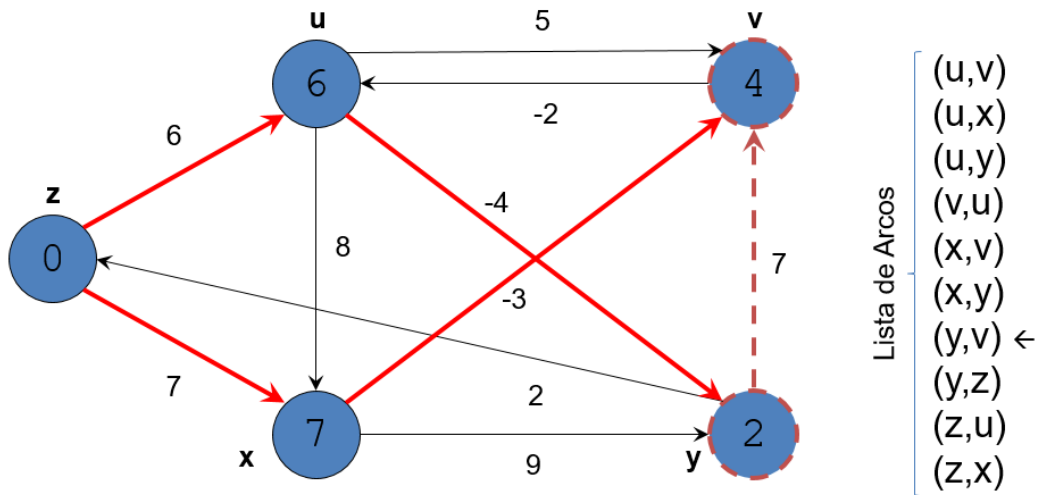
Paso 2.6

Aplicar Relax al Arco (x,y)

Pregunta: ¿ $d[y] > d[x] + w(x, y)$?

Respuesta: NO

Proceso: No se hace nada.



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 6 \ 4 \ 7 \ 2 \ 0 \}$
 $\Pi [] = \{ z \ x \ z \ u \}$

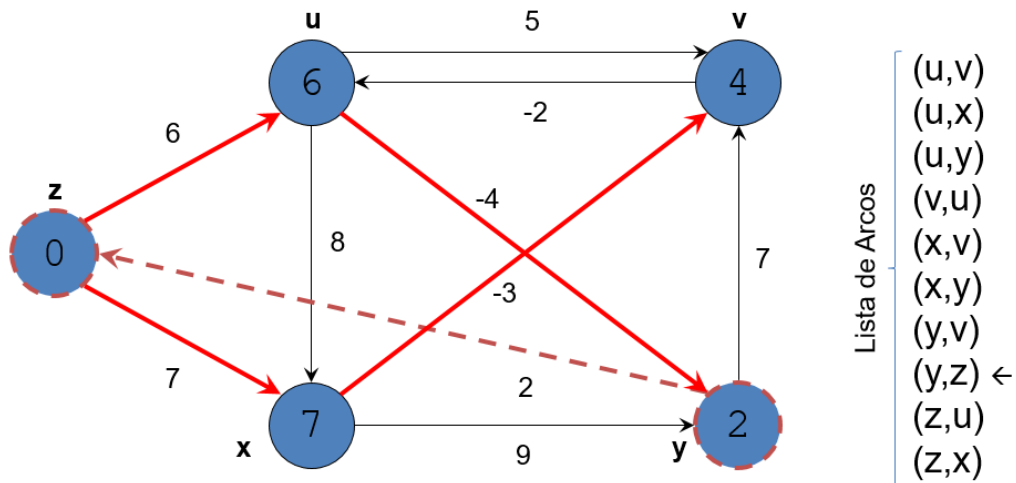
Paso 2.7

Aplicar Relax al Arco (y,v)

Pregunta: ¿ $d[v] > d[y] + w(y, v)$?

Respuesta: NO

Proceso: No se hace nada.



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 6 \ 4 \ 7 \ 2 \ 0 \}$
 $\Pi [] = \{ z \ x \ z \ u \}$

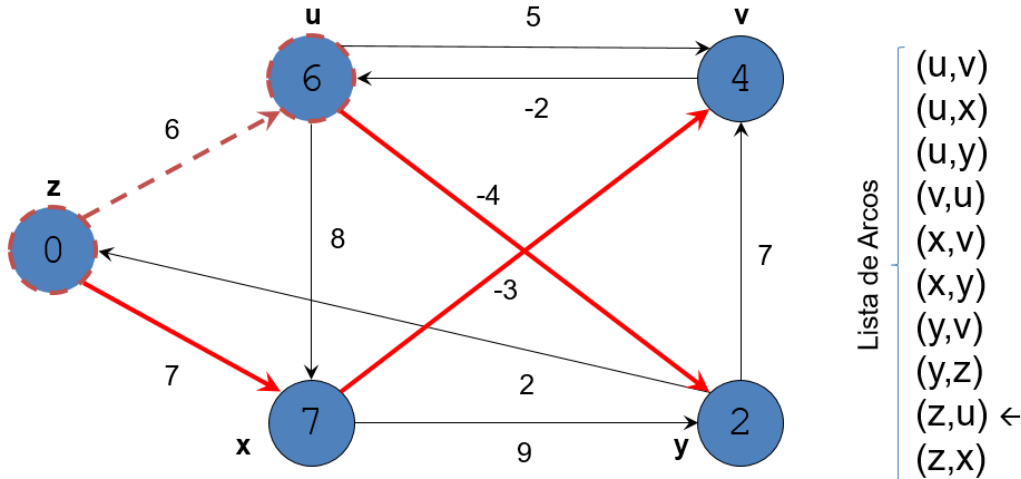
Paso 2.8

Aplicar Relax al Arco (y,z)

Pregunta: ¿ $d[z] > d[y] + w(y, z)$?

Respuesta: NO

Proceso: No se hace nada.



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 6 \ 4 \ 7 \ 2 \ 0 \}$
 $\Pi [] = \{ z \ x \ z \ u \}$

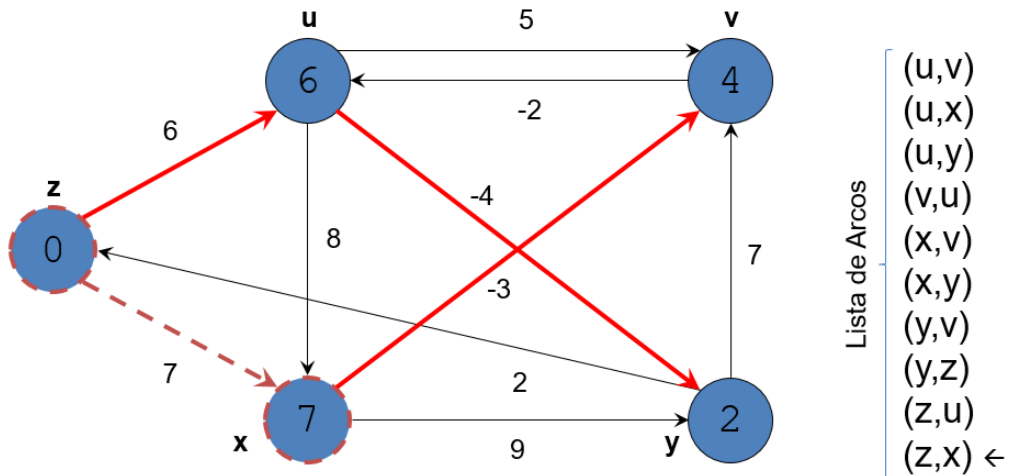
Paso 2.9

Aplicar Relax al Arco (z,u)

Pregunta: ¿ $d[u] > d[z] + w(z, u)$?

Respuesta: NO

Proceso: No se hace nada.



V [] = { u v x y z }

d [] = { 6 4 7 2 0 }

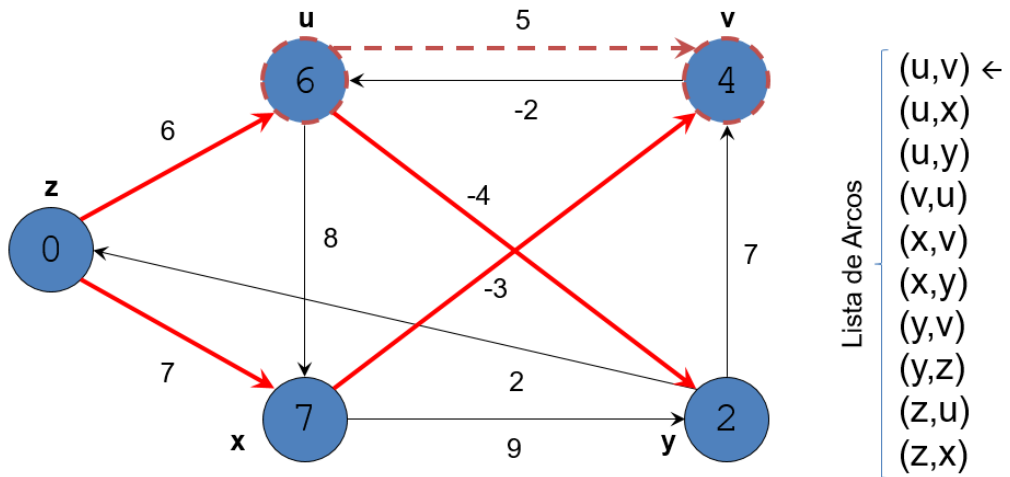
Π [] = { z x z u }

Paso 2.10 Aplicar Relax al Arco (z,x)

Pregunta: ¿ $d[x] > d[z] + w(z, x)$?

Respuesta: NO

Proceso: No se hace nada.



V [] = { u v x y z }

d [] = { 6 4 7 2 0 }

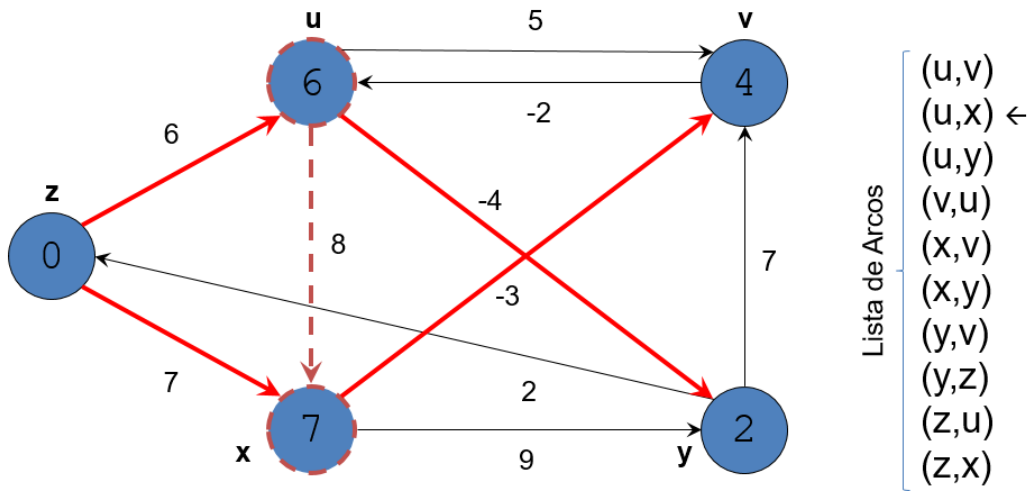
Π [] = { z x z u }

Paso 3.1 Aplicar Relax al Arco (u,v)

Pregunta: ¿ $d[v] > d[u] + w(u, v)$?

Respuesta: NO

Proceso: No se hace nada.



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 6 \ 4 \ 7 \ 2 \ 0 \}$
 $\Pi [] = \{ z \ x \ z \ u \}$

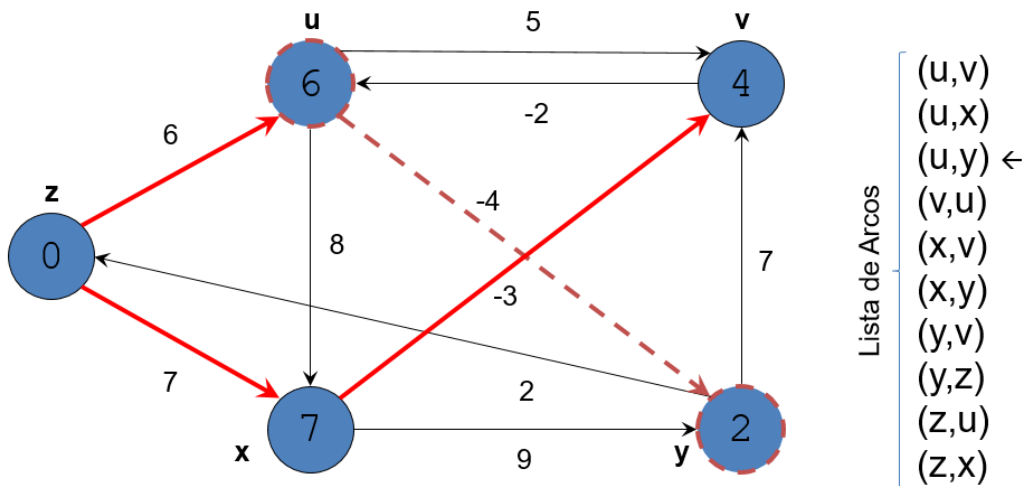
Paso 3.2

Aplicar Relax al Arco (u,x)

Pregunta: ¿ $d[x] > d[u] + w(u, x)$?

Respuesta: **NO**

Proceso: **No se hace nada.**



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 6 \ 4 \ 7 \ 2 \ 0 \}$
 $\Pi [] = \{ z \ x \ z \ u \}$

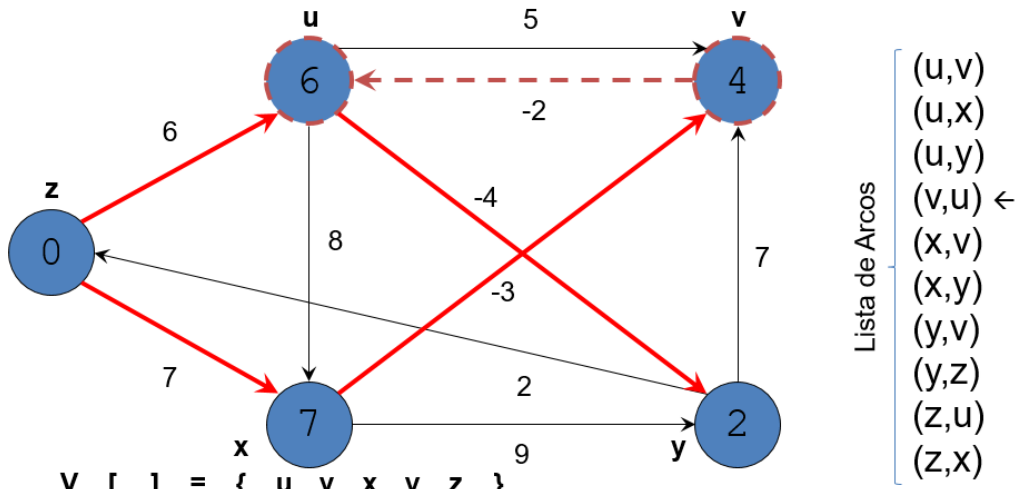
Paso 3.3

Aplicar Relax al Arco (u,y)

Pregunta: ¿ $d[y] > d[u] + w(u, y)$?

Respuesta: **NO**

Proceso: **No se hace nada.**



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 6 \ 4 \ 7 \ 2 \ 0 \}$
 $\Pi [] = \{ z \ x \ z \ u \}$

$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 2 \ 4 \ 7 \ 2 \ 0 \}$
 $\Pi [] = \{ v \ x \ z \ u \}$

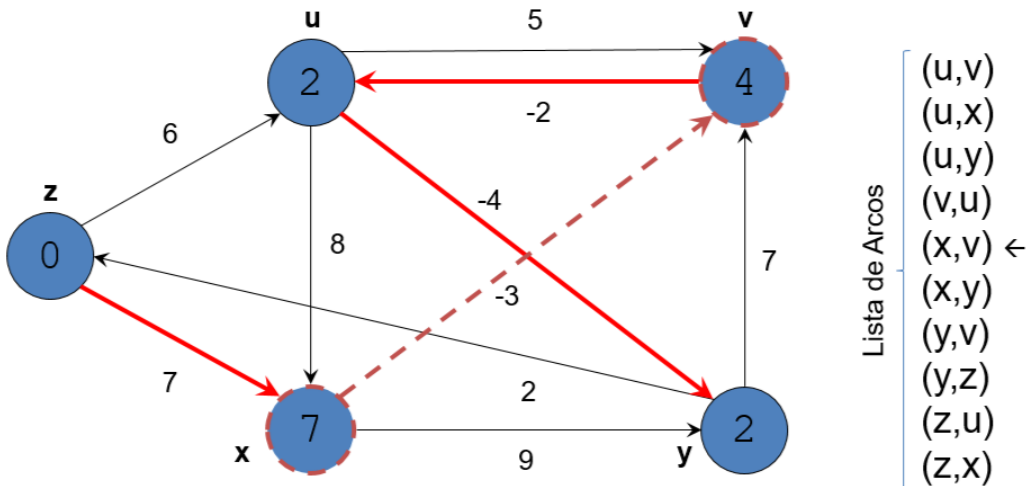
Paso 3.4

Aplicar Relax al Arco (v, u)

Pregunta: ¿ $d[u] > d[v] + w(v, u)$?

Respuesta: SI

Proceso: $d[u] = d[v] + w(v, u)$ y $\Pi[u] = v$



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 2 \ 4 \ 7 \ 2 \ 0 \}$
 $\Pi [] = \{ v \ x \ z \ u \}$

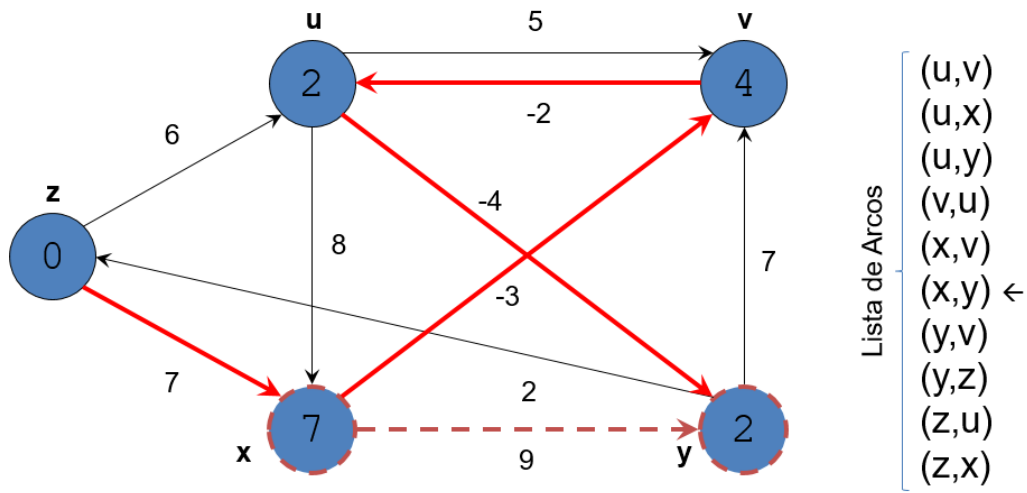
Paso 3.5

Aplicar Relax al Arco (x, v)

Pregunta: ¿ $d[v] > d[x] + w(x, v)$?

Respuesta: NO

Proceso: No se hace nada.



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 2 \ 4 \ 7 \ 2 \ 0 \}$
 $\Pi [] = \{ v \ x \ z \ u \}$

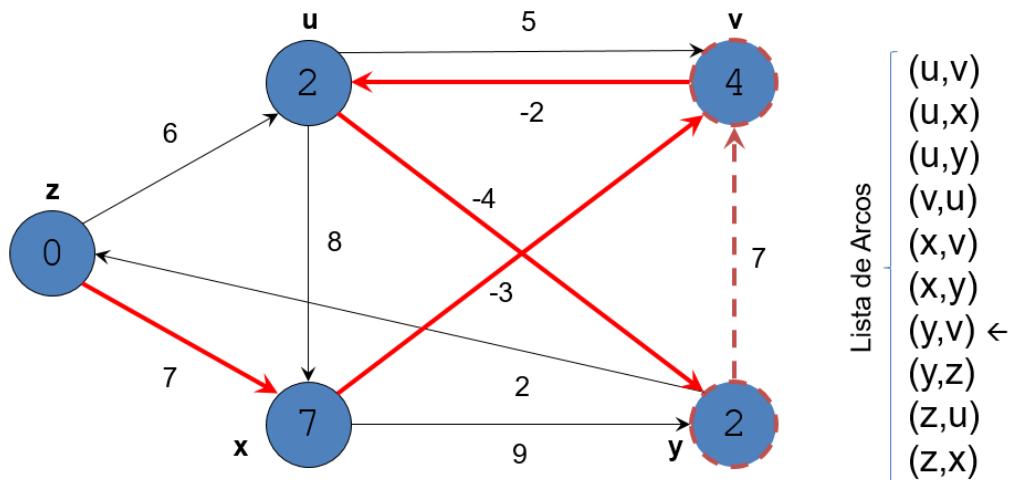
Paso 3.6

Aplicar Relax al Arco (x, y)

Pregunta: ¿ $d[y] > d[x] + w(x, y)$?

Respuesta: NO

Proceso: No se hace nada.



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 2 \ 4 \ 7 \ 2 \ 0 \}$
 $\Pi [] = \{ v \ x \ z \ u \}$

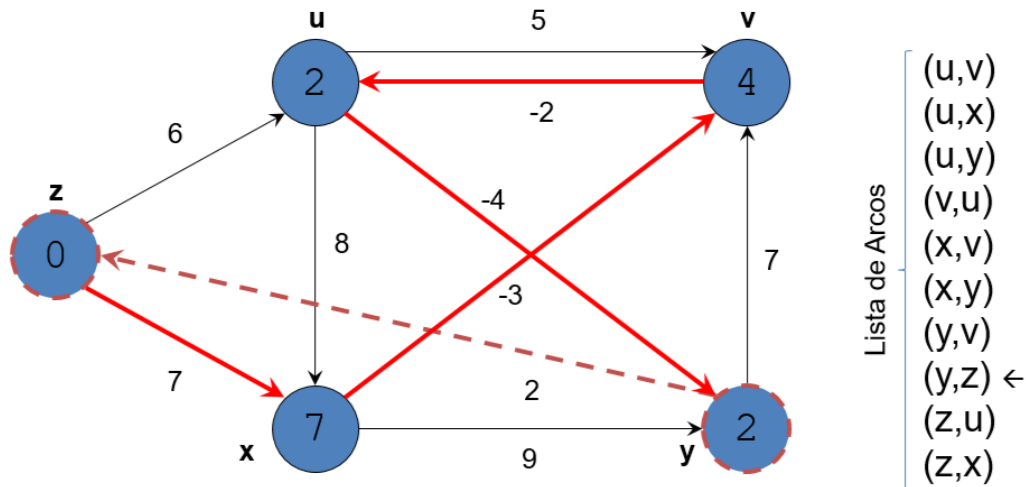
Paso 3.7

Aplicar Relax al Arco (y, v)

Pregunta: ¿ $d[v] > d[y] + w(y, v)$?

Respuesta: NO

Proceso: No se hace nada.



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 2 \ 4 \ 7 \ 2 \ 0 \}$
 $\Pi [] = \{ v \ x \ z \ u \}$

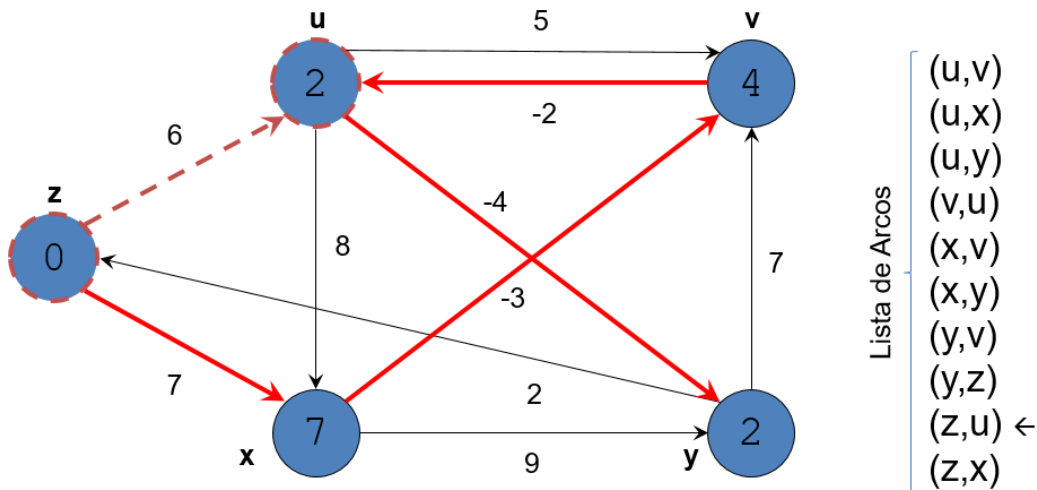
Paso 3.8

Aplicar Relax al Arco (y, z)

Pregunta: ¿ $d[z] > d[y] + w(y, z)$?

Respuesta: **NO**

Proceso: **No se hace nada.**



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 2 \ 4 \ 7 \ 2 \ 0 \}$
 $\Pi [] = \{ v \ x \ z \ u \}$

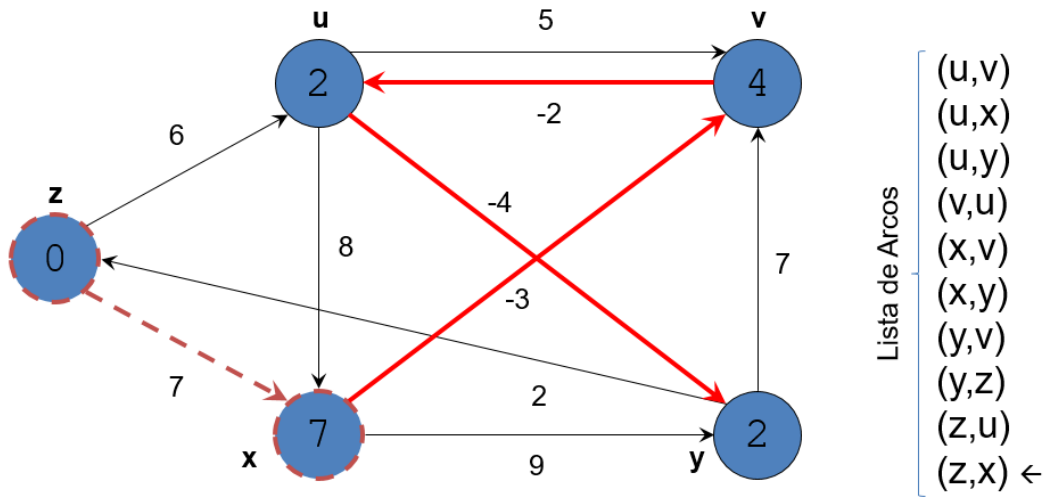
Paso 3.9

Aplicar Relax al Arco (z, u)

Pregunta: ¿ $d[u] > d[z] + w(z, u)$?

Respuesta: **NO**

Proceso: **No se hace nada.**



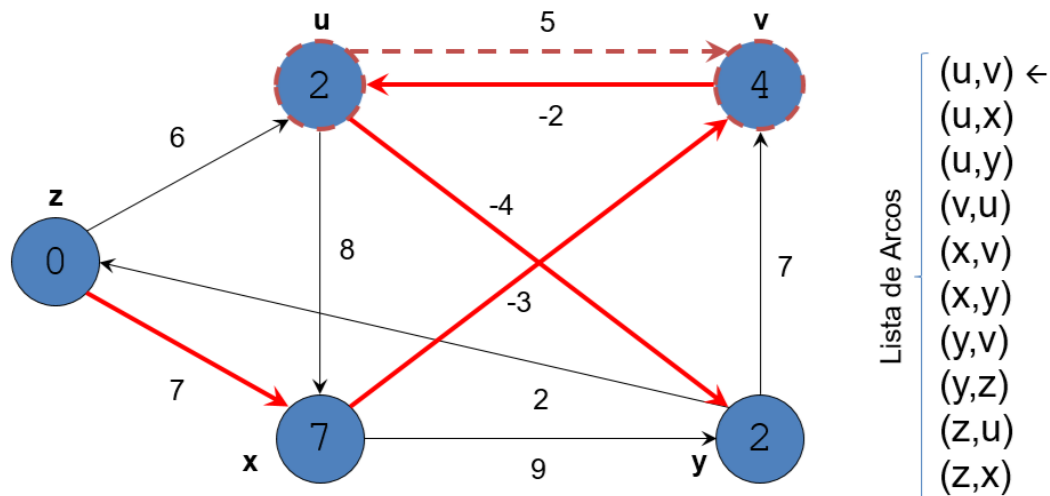
$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 2 \ 4 \ 7 \ 2 \ 0 \}$
 $\Pi [] = \{ v \ x \ z \ u \}$

Paso 3.10 Aplicar Relax al Arco (z, x)

Pregunta: ¿ $d[x] > d[z] + w(z, x)$?

Respuesta: **NO**

Proceso: **No se hace nada.**



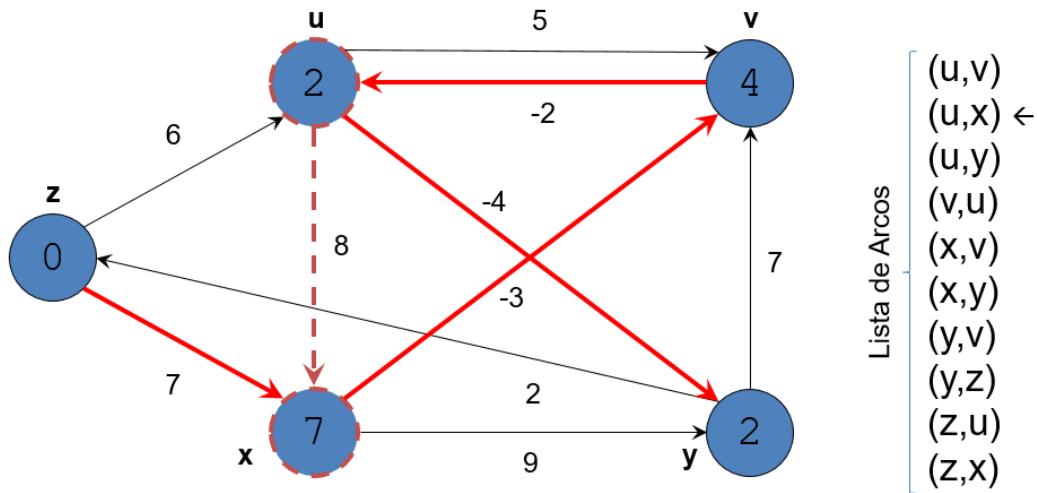
$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 2 \ 4 \ 7 \ 2 \ 0 \}$
 $\Pi [] = \{ v \ x \ z \ u \}$

Paso 4.1 Aplicar Relax al Arco (u, v)

Pregunta: ¿ $d[v] > d[u] + w(u, v)$?

Respuesta: **NO**

Proceso: **No se hace nada.**



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 2 \ 4 \ 7 \ 2 \ 0 \}$
 $\Pi [] = \{ v \ x \ z \ u \}$

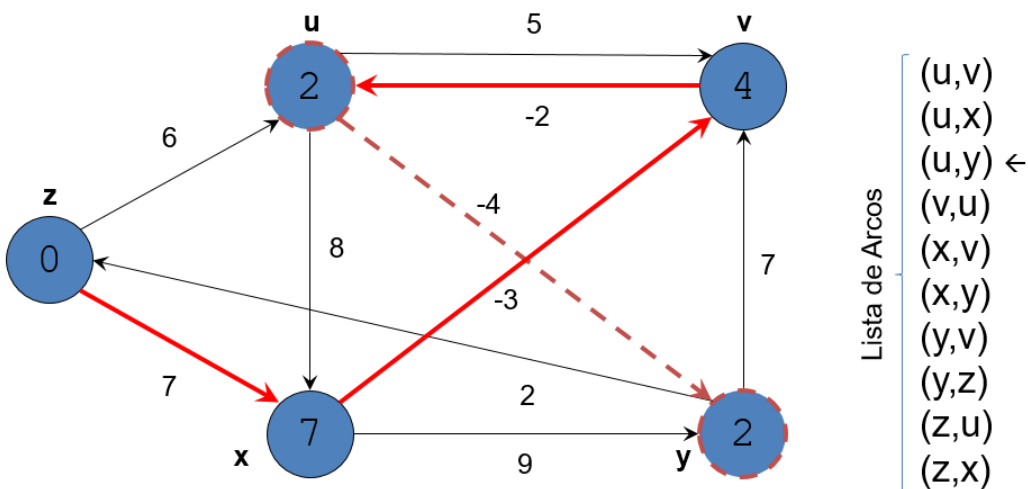
Paso 4.2

Aplicar Relax al Arco (u, x)

Pregunta: ¿ $d[x] > d[u] + w(u, x)$?

Respuesta: **NO**

Proceso: **No se hace nada.**



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 2 \ 4 \ 7 \ 2 \ 0 \}$
 $\Pi [] = \{ v \ x \ z \ u \}$
 $V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 2 \ 4 \ 7 \ -2 \ 0 \}$
 $\Pi [] = \{ v \ x \ z \ u \}$

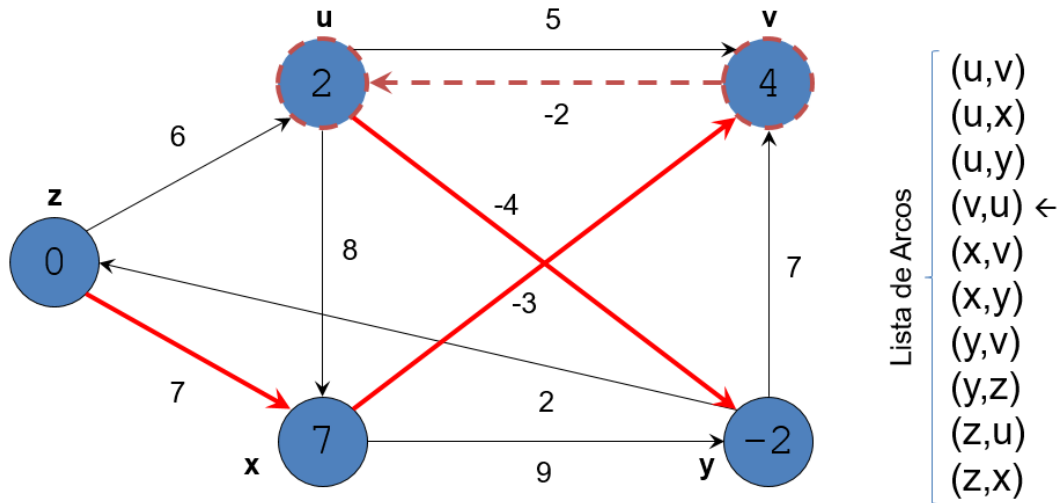
Paso 4.3

Aplicar Relax al Arco (u, y)

Pregunta: ¿ $d[y] > d[u] + w(u, y)$?

Respuesta: **SI**

Proceso: $d[y] = d[u] + w(u, y)$ y $\Pi[y] = u$



V [] = { u v x y z }

d [] = { 2 4 7 -2 0 }

Π [] = { v x z u }

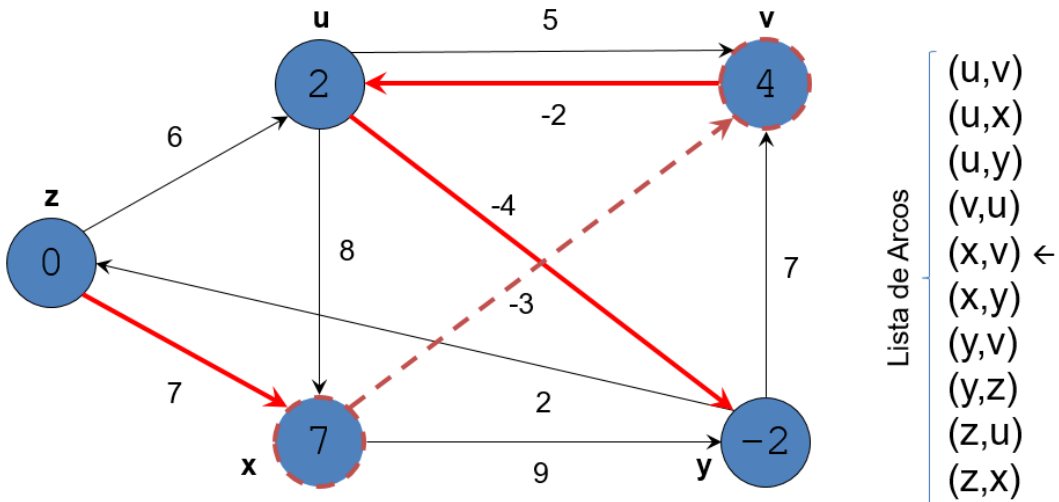
Paso 4.4

Aplicar Relax al Arco (v, u)

Pregunta: ¿ $d[u] > d[v] + w(v, u)$?

Respuesta: NO

Proceso: No se hace nada.



V [] = { u v x y z }

d [] = { 2 4 7 -2 0 }

Π [] = { v x z u }

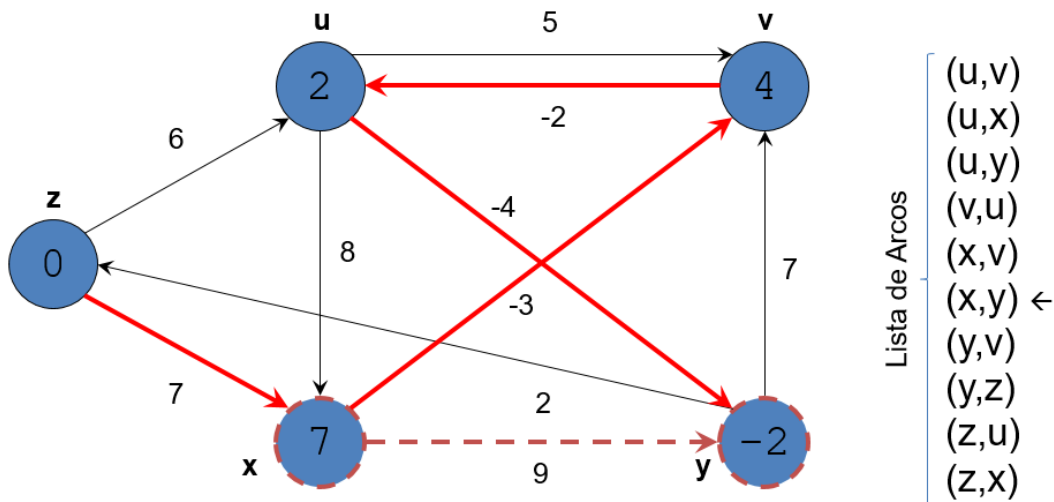
Paso 4.5

Aplicar Relax al Arco (x, v)

Pregunta: ¿ $d[v] > d[x] + w(x, v)$?

Respuesta: NO

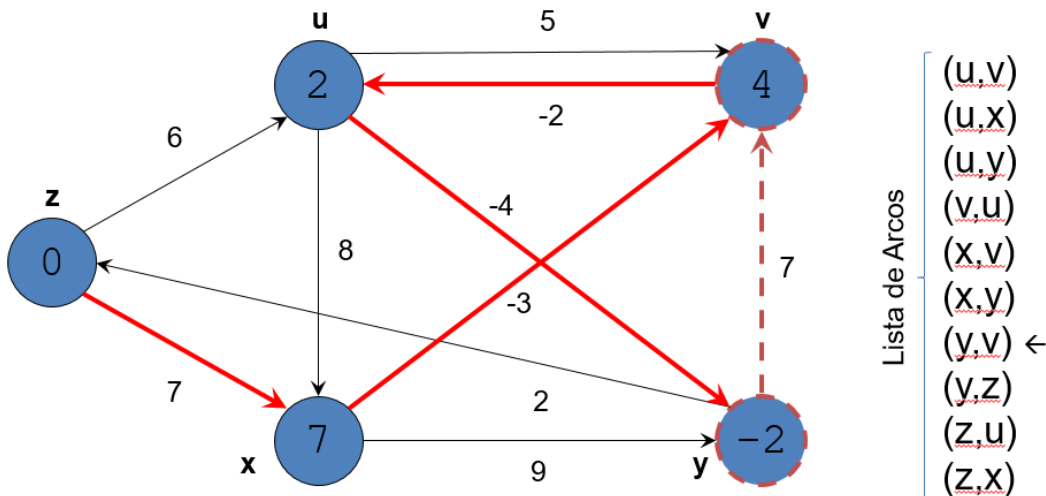
Proceso: No se hace nada.



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 2 \ 4 \ 7 \ -2 \ 0 \}$
 $\Pi [] = \{ v \ x \ z \ u \}$

Paso 4.6

Aplicar Relax al Arco (x, y)

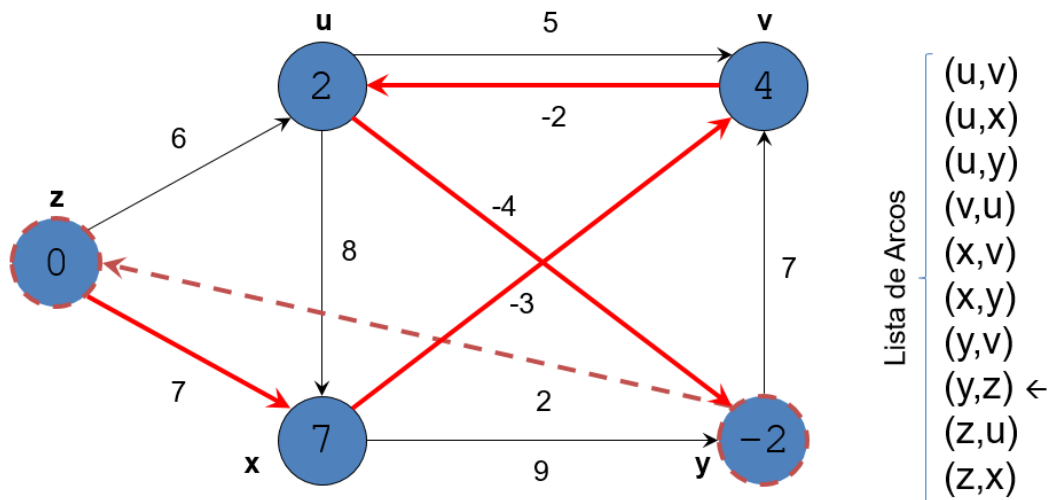
Pregunta: ¿ $d[y] > d[x] + w(x, y)$?Respuesta: **NO**Proceso: **No se hace nada.**

$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 2 \ 4 \ 7 \ -2 \ 0 \}$
 $\Pi [] = \{ v \ x \ z \ u \}$

Paso 4.7

Aplicar Relax al Arco (y, v)

Pregunta: ¿ $d[v] > d[y] + w(y, v)$?Respuesta: **NO**Proceso: **No se hace nada.**



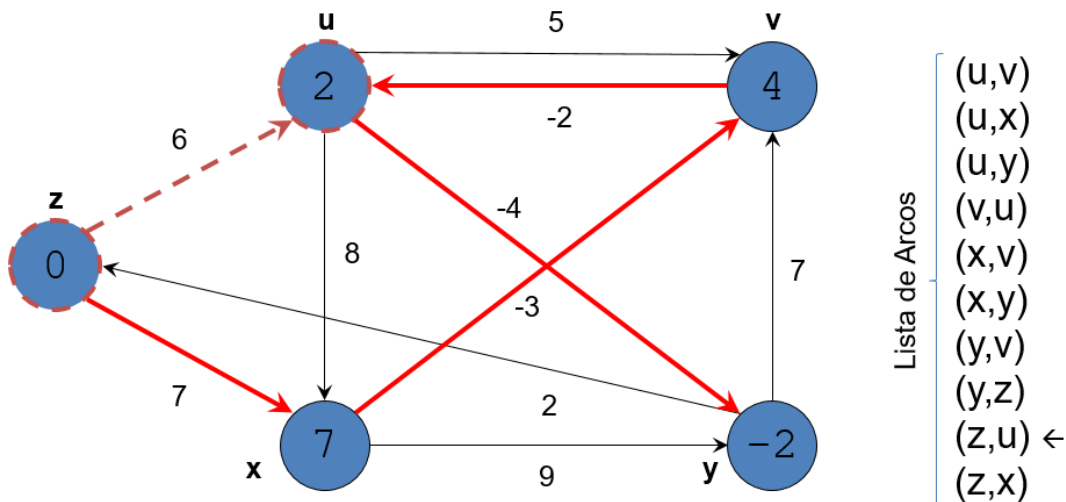
V [] = { u v x y z }
d [] = { 2 4 7 -2 0 }
Π [] = { v x z u }

Paso 4.8 Aplicar Relax al Arco (y, z)

Pregunta: ¿ $d[z] > d[y] + w(y, z)$?

Respuesta: NO

Proceso: No se hace nada.



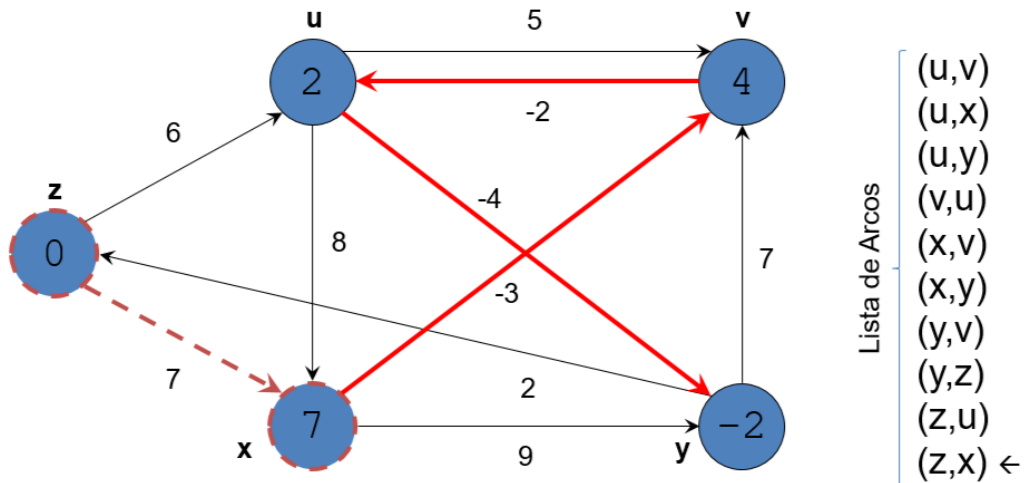
V [] = { u v x y z }
d [] = { 2 4 7 -2 0 }
Π [] = { v x z u }

Paso 4.9 Aplicar Relax al Arco (z, u)

Pregunta: ¿ $d[u] > d[z] + w(z, u)$?

Respuesta: NO

Proceso: No se hace nada.



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 2 \ 4 \ 7 \ -2 \ 0 \}$
 $\Pi [] = \{ v \ x \ z \ u \}$

Paso 4.10

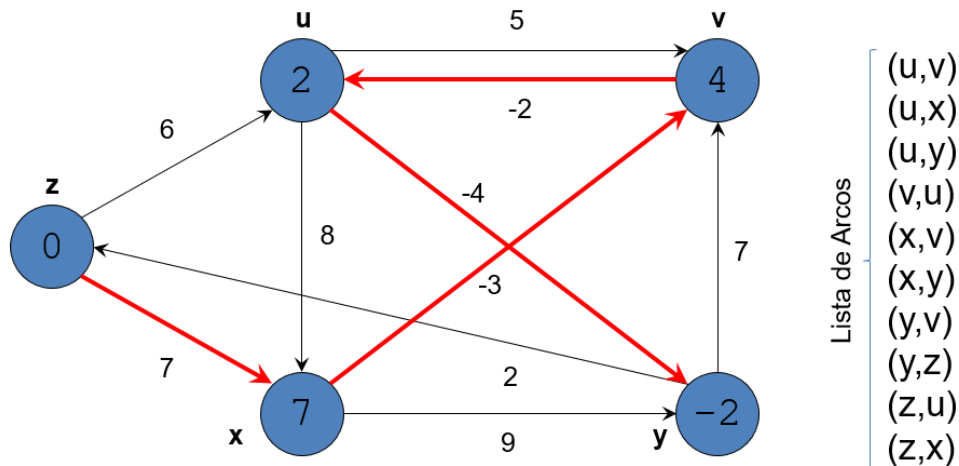
Aplicar Relax al Arco (z, x)

Pregunta: ¿ $d[x] > d[z] + w(z, x)$?

Respuesta: **NO**

Proceso: **No se hace nada.**

Como hubo modificaciones en el paso 4, se procede con el 5to paso que es analizar los 10 arcos. Y en este 5to paso no se realiza ninguna modificación.



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 2 \ 4 \ 7 \ -2 \ 0 \}$
 $\Pi [] = \{ v \ x \ z \ u \}$

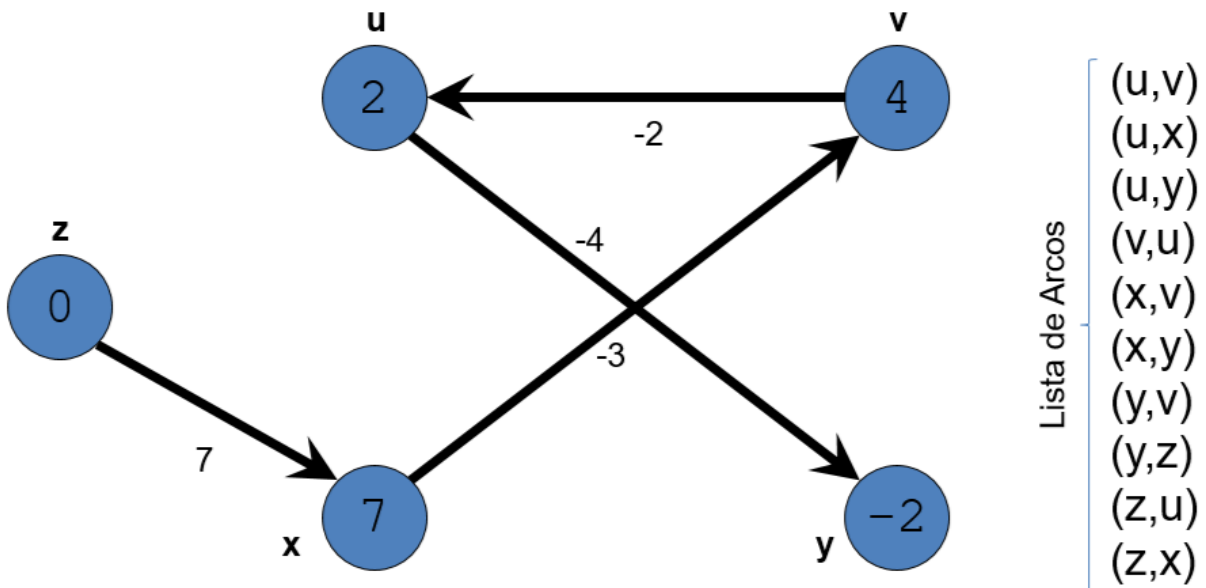
Paso 5.0

Verificar en cada arco que se cumple la condición:

$$d[V_f] \leq d[V_i] + w(V_i, V_f)$$

Si no se cumple:

=> **NO EXISTE SOLUCIÓN.**



$V [] = \{ u \ v \ x \ y \ z \}$
 $d [] = \{ 2 \ 4 \ 7 \ -2 \ 0 \}$
 $\Pi [] = \{ v \ x \ z \ u \ }$

SOLUCIÓN