

Práctica 4

Pablo Gutiérrez Aguirre
pgutierrez2018@udec.cl

18 de abril 2022

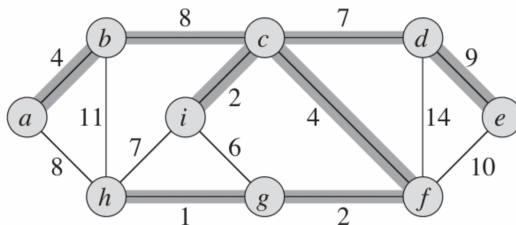
Tabla de contenidos

- 1 Repaso
 - Árbol de cobertura mínima
 - Relajación
 - Camino más corto
- 2 Ejercicio 1a
 - Kruskal
 - Prim
- 3 Ejercicio 3a
 - Bellman-Ford
- 4 Ejercicio 4
 - Bellman-Ford

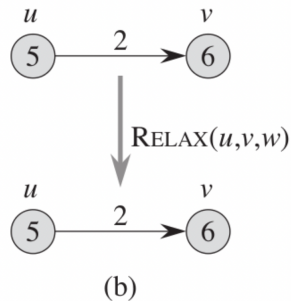
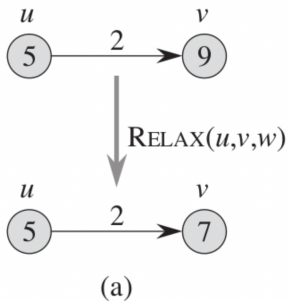
Repaso: Árbol de cobertura mínima

Definición

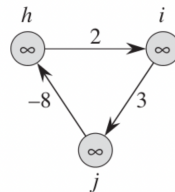
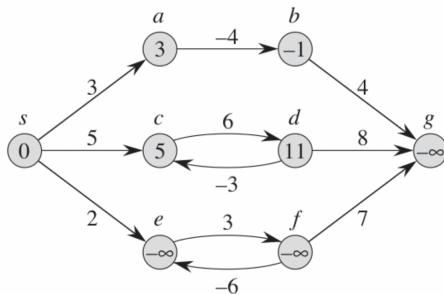
Un árbol de cobertura para un grafo no dirigido conexo $G = (V, E)$ es un subgrafo de G que es un árbol no dirigido y contiene todos los vértices de G . En un grafo ponderado o con pesos $G = (V, E, W)$, el peso de un subgrafo es la suma de los pesos de las aristas incluidas en ese subgrafo. Un árbol cobertura mínima para un grafo ponderado es un árbol cobertura cuyo peso es mínima.



Repaso: Relajación

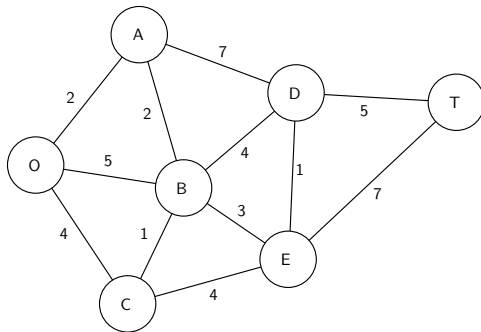


Repaso: Camino más corto



Ejercicio 1a: Kruskal

En una determinada ciudad, se requiere establecer los arcos por los que se requiere pasar líneas telefónicas, para conectar todos los sectores (nodos) con longitud total mínima de cable. Los sectores y distancias entre éstos, se representan en la siguiente red:



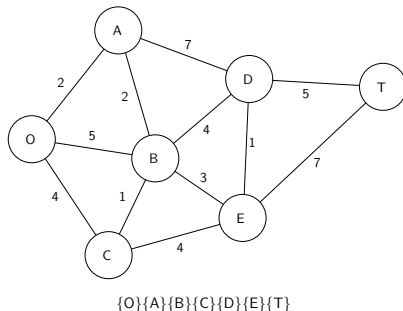
Se pide resolver mediante **algoritmo de Kruskal** y algoritmo de Prim.

Ejercicio 1a: Kruskal

Algorithm: KRUSKAL

Entrada : G: Grafo, w: pesos aristas**Salida** : A: aristas del MTS

```
1  $A \leftarrow \emptyset$ 
2 for vértice  $u \in G.V$  do
3   |  $\text{MakeSet}(u)$ 
4 end
5 Ordenar aristas por peso w creciente
6 for arista  $(u,v) \in G.E$  do
7   | if  $\text{Find}(U) \neq \text{Find}(V)$  then
8   |   |  $A \leftarrow A \cup \{(u,v)\}$ 
9   |   |  $\text{Union}(u,v)$ 
10  | end
11 end
12
```



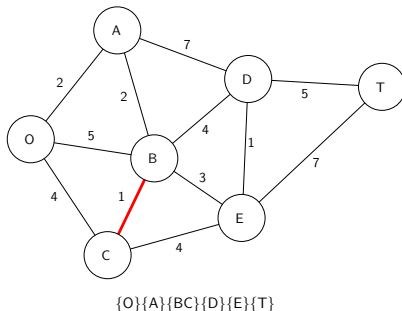
Ejercicio 1a: Kruskal

Algorithm: KRUSKAL

Entrada : G: Grafo, w: pesos aristas

Salida : A: aristas del MTS

```
1 A ← ∅
2 for vértice u ∈ G.V do
3   | MakeSet(u)
4 end
5 Ordenar aristas por peso w creciente
6 for arista (u,v) ∈ G.E do
7   | if Find(U) ≠ Find(V) then
8     |   A ← A ∪ {(u,v)}
9     |   Union(u,v)
10  | end
11 end
12
```

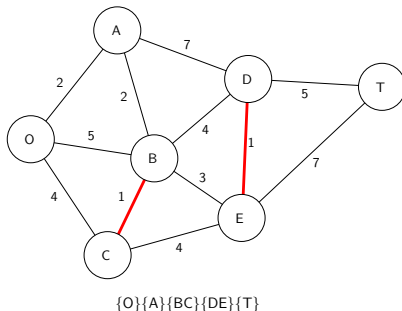


Ejercicio 1a: Kruskal

Algorithm: KRUSKAL

Entrada : G: Grafo,w: pesos aristas**Salida** : A: aristas del MTS

```
1 A ← ∅
2 for vértice u ∈ G.V do
3   | MakeSet(u)
4 end
5 Ordenar aristas por peso w creciente
6 for arista (u,v) ∈ G.E do
7   | if Find(U) ≠ Find(V) then
8     |   A ← A ∪ {(u,v)}
9     |   Union(u,v)
10  | end
11 end
12
```

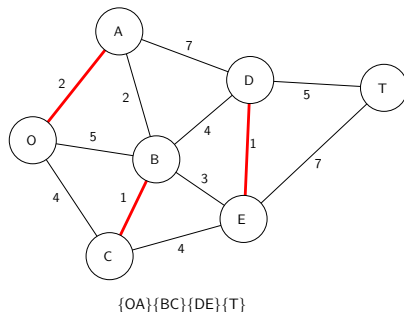


Ejercicio 1a: Kruskal

Algorithm: KRUSKAL

Entrada : G: Grafo,w: pesos aristas**Salida** : A: aristas del MTS

```
1 A ← ∅
2 for vértice u ∈ G.V do
3   | MakeSet(u)
4 end
5 Ordenar aristas por peso w creciente
6 for arista (u,v) ∈ G.E do
7   | if Find(U) ≠ Find(V) then
8     |   A ← A ∪ {(u,v)}
9     |   Union(u,v)
10  | end
11 end
12
```



Ejercicio 1a: Kruskal

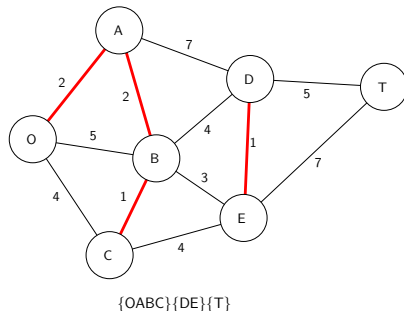
Algorithm: KRUSKAL

Entrada : G: Grafo, w: pesos aristas

Salida : A: aristas del MTS

```

1  A ← ∅
2  for vértice u ∈ G.V do
3    |  MakeSet(u)
4  end
5  Ordenar aristas por peso w creciente
6  for arista (u,v) ∈ G.E do
7    |  if Find(U) ≠ Find(V) then
8    |    |  A ← A ∪ {(u,v)}
9    |    |  Union(u,v)
10   |  end
11 end
12
```



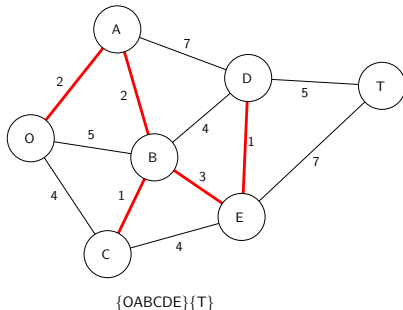
Ejercicio 1a: Kruskal

Algorithm: KRUSKAL

Entrada : G: Grafo, w: pesos aristas

Salida : A: aristas del MTS

```
1 A ← ∅
2 for vértice u ∈ G.V do
3   | MakeSet(u)
4 end
5 Ordenar aristas por peso w creciente
6 for arista (u,v) ∈ G.E do
7   | if Find(U) ≠ Find(V) then
8     |   A ← A ∪ {(u,v)}
9     |   Union(u,v)
10  | end
11 end
12
```



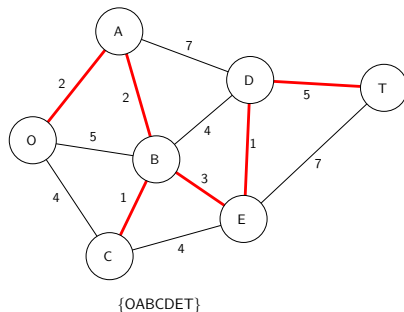
Ejercicio 1a: Kruskal

Algorithm: KRUSKAL

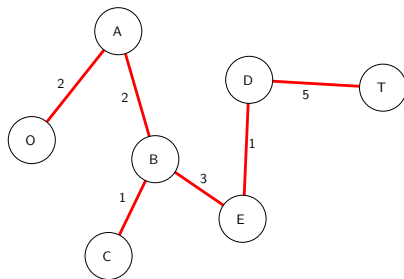
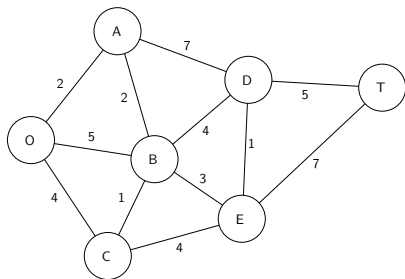
Entrada : G: Grafo,w: pesos aristas

Salida : A: aristas del MTS

```
1 A ← ∅
2 for vértice u ∈ G.V do
3   | MakeSet(u)
4 end
5 Ordenar aristas por peso w creciente
6 for arista (u,v) ∈ G.E do
7   | if Find(U) ≠ Find(V) then
8     |   A ← A ∪ {(u,v)}
9     |   Union(u,v)
10  | end
11 end
12
```

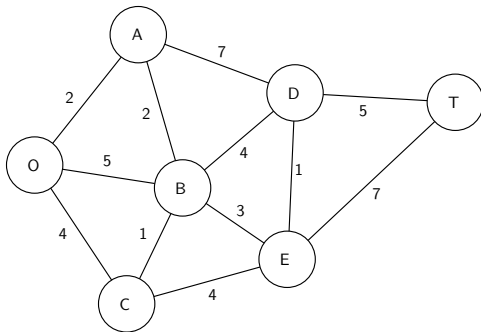


Ejercicio 1a: Kruskal



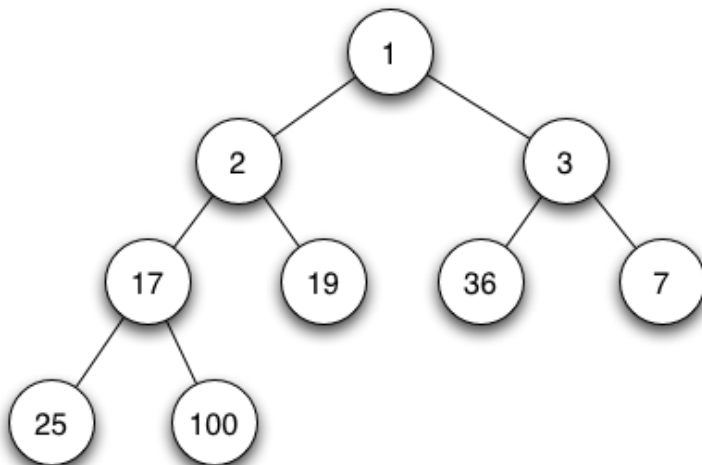
Ejercicio 1a: Prim

En una determinada ciudad, se requiere establecer los arcos por los que se requiere pasar líneas telefónicas, para conectar todos los sectores (nodos) con longitud total mínima de cable. Los sectores y distancias entre éstos, se representan en la siguiente red:



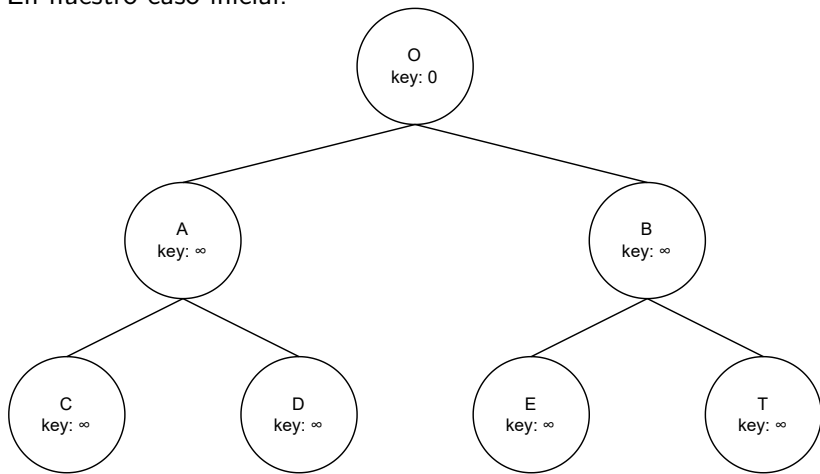
Se pide resolver mediante algoritmo de Kruskal y **algoritmo de Prim**.

Ejercicio 1a: Heap binario



Ejercicio 1a: Heap binario

En nuestro caso inicial:



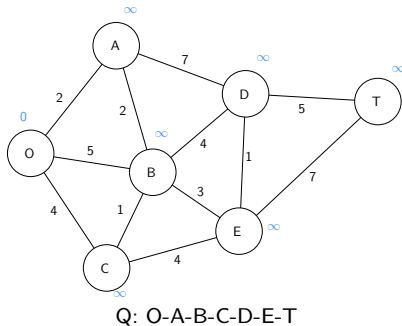
Ejercicio 1a: Prim

Algorithm: Prim

Entrada : G : Grafo, w : pesos aristas,
 u : vertice inicial

Salida : G con key y π en vértices

```
1 for  $v \in G.V$  do
2    $v.key \leftarrow \infty$ 
3    $v.\pi \leftarrow NULL$ 
4 end
5  $u.key \leftarrow 0$ 
6  $Q \leftarrow G.V$ 
7 while  $Q \neq \emptyset$  do
8    $u \leftarrow EXTRAER - MIN(Q)$ 
9   for  $v \in G.adyacentes(u)$  do
10    if  $v \in Q$  y  $w(u, v) < v.key$  then
11       $v.\pi \leftarrow u$ 
12       $v.key \leftarrow w(u, v)$ 
13    end
14  end
15 end
```



Ejercicio 1a: Prim

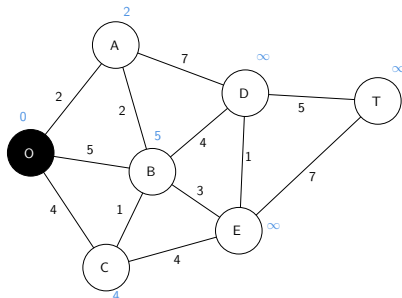
Algorithm: Prim

Entrada : G : Grafo, w : pesos aristas,
 u : vertice inicial

Salida : G con key y π en vértices

```

1 for  $v \in G.V$  do
2    $v.key \leftarrow \infty$ 
3    $v.\pi \leftarrow NULL$ 
4 end
5  $u.key \leftarrow 0$ 
6  $Q \leftarrow G.V$ 
7 while  $Q \neq \emptyset$  do
8    $u \leftarrow EXTRAER - MIN(Q)$ 
9   for  $v \in G.adyacentes(u)$  do
10    if  $v \in Q$  y  $w(u, v) < v.key$  then
11       $v.\pi \leftarrow u$ 
12       $v.key \leftarrow w(u, v)$ 
13    end
14  end
15 end
  
```



Q: A-C-B-D-E-T

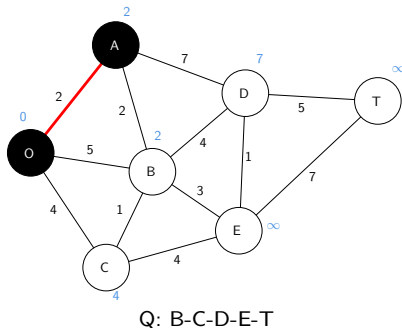
Ejercicio 1a: Prim

Algorithm: Prim

Entrada : G : Grafo, w : pesos aristas,
 u : vertice inicial

Salida : G con key y π en vértices

```
1 for  $v \in G.V$  do
2    $v.key \leftarrow \infty$ 
3    $v.\pi \leftarrow NULL$ 
4 end
5  $u.key \leftarrow 0$ 
6  $Q \leftarrow G.V$ 
7 while  $Q \neq \emptyset$  do
8    $u \leftarrow EXTRAER - MIN(Q)$ 
9   for  $v \in G.adyacentes(u)$  do
10    if  $v \in Q$  y  $w(u, v) < v.key$  then
11       $v.\pi \leftarrow u$ 
12       $v.key \leftarrow w(u, v)$ 
13    end
14  end
15 end
```



Ejercicio 1a: Prim

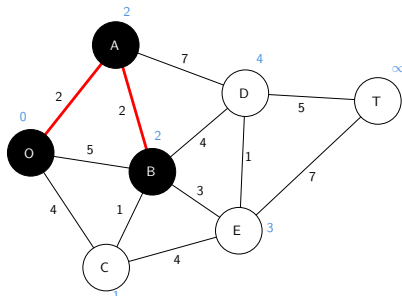
Algorithm: Prim

Entrada : G : Grafo, w : pesos aristas,
 u : vertice inicial

Salida : G con key y π en vértices

```

1 for  $v \in G.V$  do
2    $v.key \leftarrow \infty$ 
3    $v.\pi \leftarrow NULL$ 
4 end
5  $u.key \leftarrow 0$ 
6  $Q \leftarrow G.V$ 
7 while  $Q \neq \emptyset$  do
8    $u \leftarrow EXTRAER - MIN(Q)$ 
9   for  $v \in G.adyacentes(u)$  do
10    if  $v \in Q$  y  $w(u, v) < v.key$  then
11       $v.\pi \leftarrow u$ 
12       $v.key \leftarrow w(u, v)$ 
13    end
14  end
15 end
  
```



Q: C-E-D-T

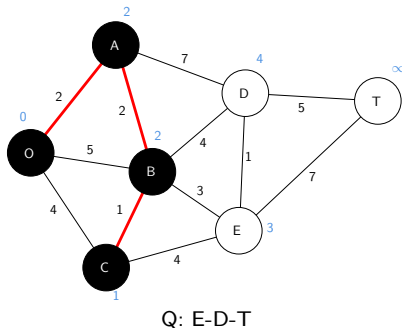
Ejercicio 1a: Prim

Algorithm: Prim

Entrada : G : Grafo, w : pesos aristas,
 u : vertice inicial

Salida : G con key y π en vértices

```
1 for  $v \in G.V$  do
2    $v.key \leftarrow \infty$ 
3    $v.\pi \leftarrow NULL$ 
4 end
5  $u.key \leftarrow 0$ 
6  $Q \leftarrow G.V$ 
7 while  $Q \neq \emptyset$  do
8    $u \leftarrow EXTRAER - MIN(Q)$ 
9   for  $v \in G.adyacentes(u)$  do
10    if  $v \in Q$  y  $w(u, v) < v.key$  then
11       $v.\pi \leftarrow u$ 
12       $v.key \leftarrow w(u, v)$ 
13    end
14  end
15 end
```



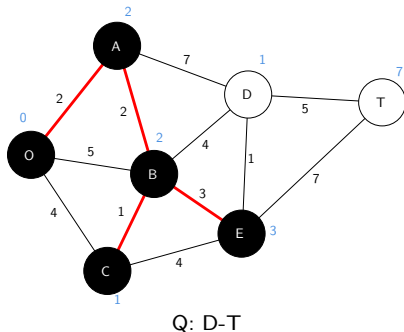
Ejercicio 1a: Prim

Algorithm: Prim

Entrada : G : Grafo, w : pesos aristas,
 u : vertice inicial

Salida : G con key y π en vértices

```
1 for  $v \in G.V$  do
2    $v.key \leftarrow \infty$ 
3    $v.\pi \leftarrow NULL$ 
4 end
5  $u.key \leftarrow 0$ 
6  $Q \leftarrow G.V$ 
7 while  $Q \neq \emptyset$  do
8    $u \leftarrow EXTRAER - MIN(Q)$ 
9   for  $v \in G.adyacentes(u)$  do
10    if  $v \in Q$  y  $w(u, v) < v.key$  then
11       $v.\pi \leftarrow u$ 
12       $v.key \leftarrow w(u, v)$ 
13    end
14  end
15 end
```



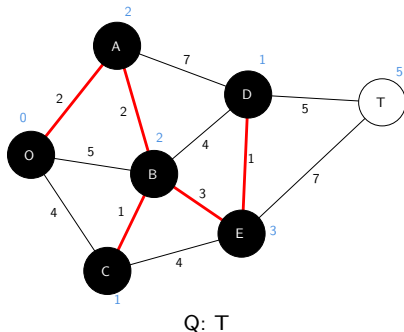
Ejercicio 1a: Prim

Algorithm: Prim

Entrada : G : Grafo, w : pesos aristas,
 u : vertice inicial

Salida : G con key y π en vértices

```
1 for  $v \in G.V$  do
2    $v.key \leftarrow \infty$ 
3    $v.\pi \leftarrow NULL$ 
4 end
5  $u.key \leftarrow 0$ 
6  $Q \leftarrow G.V$ 
7 while  $Q \neq \emptyset$  do
8    $u \leftarrow EXTRAER - MIN(Q)$ 
9   for  $v \in G.adyacentes(u)$  do
10    if  $v \in Q$  y  $w(u, v) < v.key$  then
11       $v.\pi \leftarrow u$ 
12       $v.key \leftarrow w(u, v)$ 
13    end
14  end
15 end
```



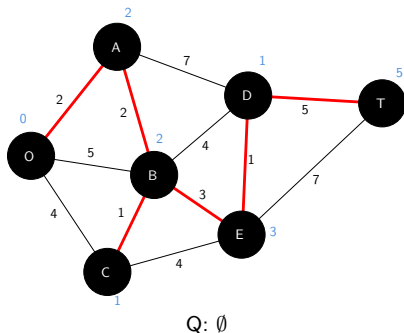
Ejercicio 1a: Prim

Algorithm: Prim

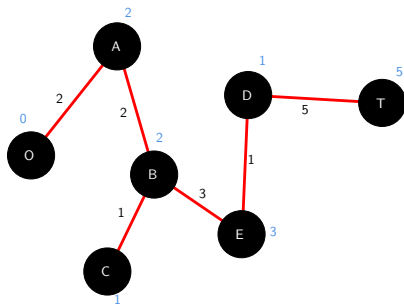
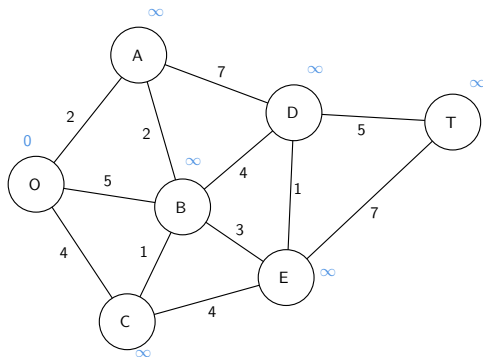
Entrada : G : Grafo, w : pesos aristas,
 u : vertice inicial

Salida : G con key y π en vértices

```
1 for  $v \in G.V$  do
2    $v.key \leftarrow \infty$ 
3    $v.\pi \leftarrow NULL$ 
4 end
5  $u.key \leftarrow 0$ 
6  $Q \leftarrow G.V$ 
7 while  $Q \neq \emptyset$  do
8    $u \leftarrow EXTRAER - MIN(Q)$ 
9   for  $v \in G.adyacentes(u)$  do
10    if  $v \in Q$  y  $w(u, v) < v.key$  then
11       $v.\pi \leftarrow u$ 
12       $v.key \leftarrow w(u, v)$ 
13    end
14  end
15 end
```

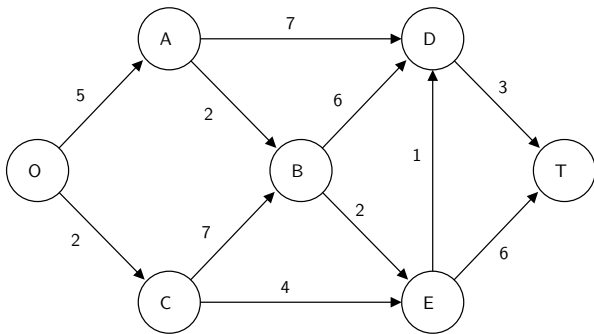


Ejercicio 1a: Prim



Ejercicio 3a

Considere la siguiente red dirigida, donde los valores escritos indican el costo de transitar cada arco.



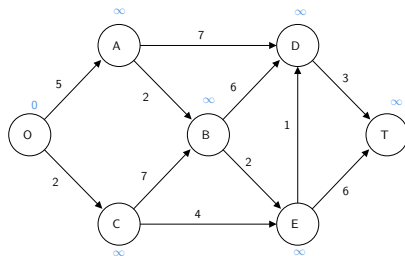
Use el algoritmo de **Bellman-Ford** para determinar el camino mas corto desde el nodo (O) a los demás nodos. Muestre todo el proceso paso a paso.

Ejercicio 3a: Bellman-Ford

Algorithm: Bellman Ford

Entrada : G : Grafo, w : pesos aristas,
 s : vértice inicial

```
1 for  $v \in G.V$  do
2   |  $v.d \leftarrow \infty$ 
3   |  $v.\pi \leftarrow NULL$ 
4 end
5  $s.d \leftarrow 0$ 
6 for  $i \leftarrow 0$  to  $|G.V| - 1$  do
7   | for  $arista (u,v) \in G.E$  do
8     | if  $v.d > u.d + w(u,v)$  then
9       |   |  $v.d \leftarrow u.d + w(u,v)$ 
10      |   |  $v.\pi \leftarrow u$ 
11     |   end
12   | end
13 end
14 for  $arista (u,v) \in G.E$  do
15   | if  $v.d > u.d + w(u,v)$  then
16     |   return False
17   | end
18 end
19 return True
20 1-5: INICIALIZAR-UNICO-ORIGEN
21 8-11: RELAJACIÓN
```



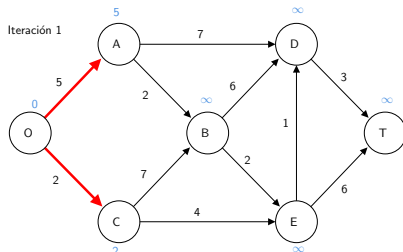
Orden aristas:
OA, OC, AB, AD, CB, CE, BD, BE, DT, ED, ET

Ejercicio 3a: Bellman-Ford

Algorithm: Bellman Ford

Entrada : G: Grafo, w: pesos aristas,
s: vértice inicial

```
1 for v ∈ G.V do
2   | v.d ← ∞
3   | v.π ← NULL
4 end
5 s.d ← 0
6 for i ← 0 to |G.V| - 1 do
7   | for arista (u,v) ∈ G.E do
8     | | if v.d > u.d + w(u,v) then
9       | |   v.d ← u.d + w(u,v)
10      | |   v.π ← u
11     | | end
12   | end
13 end
14 for arista (u,v) ∈ G.E do
15   | if v.d > u.d + w(u,v) then
16     | | return False
17   | end
18 end
19 return True
20 1-5: INICIALIZAR-UNICO-ORIGEN
21 8-11: RELAJACIÓN
```



Orden aristas:

OA, OC, AB, AD, CB, CE, BD, BE, DT, ED, ET

Ejercicio 3a: Bellman-Ford

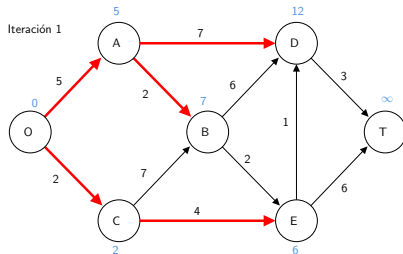
Algorithm: Bellman Ford

Entrada : G: Grafo, w: pesos aristas,
s: vértice inicial

```

1  for  $v \in G.V$  do
2    |  $v.d \leftarrow \infty$ 
3    |  $v.\pi \leftarrow NULL$ 
4  end
5   $s.d \leftarrow 0$ 
6  for  $i \leftarrow 0$  to  $|G.V| - 1$  do
7    | for arista  $(u,v) \in G.E$  do
8    | | if  $v.d > u.d + w(u,v)$  then
9    | | |  $v.d \leftarrow u.d + w(u,v)$ 
10   | | |  $v.\pi \leftarrow u$ 
11   | | end
12   | end
13 end
14 for arista  $(u,v) \in G.E$  do
15   | if  $v.d > u.d + w(u,v)$  then
16   | | return False
17   | end
18 end
19 return True
20 1-5: INICIALIZAR-UNICO-ORIGEN
21 8-11: RELAJACIÓN

```



Orden aristas:
OA, OC, AB, AD, CB, CE, BD, BE, DT, ED, ET

Ejercicio 3a: Bellman-Ford

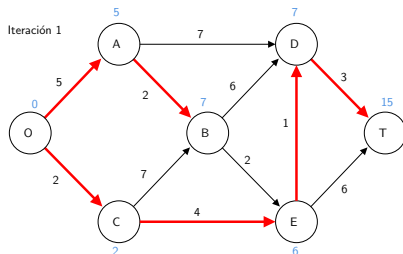
Algorithm: Bellman Ford

Entrada : G: Grafo, w: pesos aristas,
s: vértice inicial

```

1 for  $v \in G.V$  do
2   |  $v.d \leftarrow \infty$ 
3   |  $v.\pi \leftarrow NULL$ 
4 end
5  $s.d \leftarrow 0$ 
6 for  $i \leftarrow 0$  to  $|G.V| - 1$  do
7   | for arista  $(u,v) \in G.E$  do
8     | if  $v.d > u.d + w(u,v)$  then
9       |   |  $v.d \leftarrow u.d + w(u,v)$ 
10      |   |  $v.\pi \leftarrow u$ 
11     |   end
12   | end
13 end
14 for arista  $(u,v) \in G.E$  do
15   | if  $v.d > u.d + w(u,v)$  then
16     |   return False
17   | end
18 end
19 return True
20 1-5: INICIALIZAR-UNICO-ORIGEN
21 8-11: RELAJACIÓN

```



Orden aristas:
OA, OC, AB, AD, CB, CE, **BD, BE, DT, ED, ET**

Ejercicio 3a: Bellman-Ford

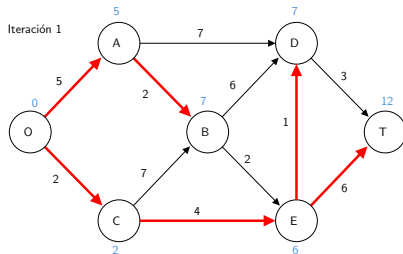
Algorithm: Bellman Ford

Entrada : G: Grafo, w: pesos aristas,
s: vértice inicial

```

1 for  $v \in G.V$  do
2    $v.d \leftarrow \infty$ 
3    $v.\pi \leftarrow NULL$ 
4 end
5  $s.d \leftarrow 0$ 
6 for  $i \leftarrow 0$  to  $|G.V| - 1$  do
7   for arista  $(u,v) \in G.E$  do
8     if  $v.d > u.d + w(u,v)$  then
9        $v.d \leftarrow u.d + w(u,v)$ 
10       $v.\pi \leftarrow u$ 
11    end
12  end
13 end
14 for arista  $(u,v) \in G.E$  do
15   if  $v.d > u.d + w(u,v)$  then
16     return False
17   end
18 end
19 return True
20 1-5: INICIALIZAR-UNICO-ORIGEN
21 8-11: RELAJACIÓN

```



Orden aristas:

OA, OC, AB, AD, CB, CE, BD, BE, DT, ED, **ET**

Ejercicio 3a: Bellman-Ford

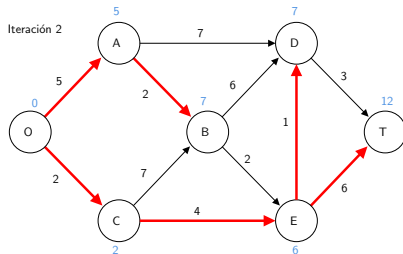
Algorithm: Bellman Ford

Entrada : G : Grafo, w : pesos aristas,
s: vértice inicial

```

1 for  $v \in G.V$  do
2    $v.d \leftarrow \infty$ 
3    $v.\pi \leftarrow NULL$ 
4 end
5  $s.d \leftarrow 0$ 
6 for  $i \leftarrow 0$  to  $|G.V| - 1$  do
7   for arista  $(u,v) \in G.E$  do
8     if  $v.d > u.d + w(u,v)$  then
9        $v.d \leftarrow u.d + w(u,v)$ 
10       $v.\pi \leftarrow u$ 
11    end
12  end
13 end
14 for arista  $(u,v) \in G.E$  do
15   if  $v.d > u.d + w(u,v)$  then
16     return False
17   end
18 end
19 return True
20 1-5: INICIALIZAR-UNICO-ORIGEN
21 8-11: RELAJACIÓN

```



Orden aristas:

OA,OC,AB,AD,CB,CE,BD,BE,DT,ED,ET

Ejercicio 3a: Bellman-Ford

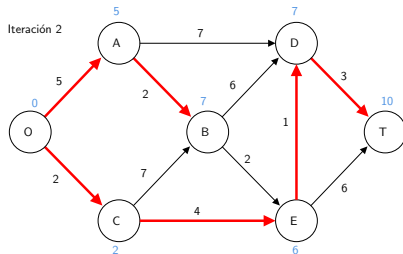
Algorithm: Bellman Ford

Entrada : G: Grafo, w: pesos aristas,
s: vértice inicial

```

1 for  $v \in G.V$  do
2   |  $v.d \leftarrow \infty$ 
3   |  $v.\pi \leftarrow NULL$ 
4 end
5  $s.d \leftarrow 0$ 
6 for  $i \leftarrow 0$  to  $|G.V| - 1$  do
7   | for arista  $(u,v) \in G.E$  do
8     | if  $v.d > u.d + w(u,v)$  then
9       |   |  $v.d \leftarrow u.d + w(u,v)$ 
10      |   |  $v.\pi \leftarrow u$ 
11     |   end
12   | end
13 end
14 for arista  $(u,v) \in G.E$  do
15   | if  $v.d > u.d + w(u,v)$  then
16     |   return False
17   | end
18 end
19 return True
20 1-5: INICIALIZAR-UNICO-ORIGEN
21 8-11: RELAJACIÓN

```



Orden aristas:
OA, OC, AB, AD, CB, CE, BD, BE, **DT, ED, ET**

Ejercicio 3a: Bellman-Ford

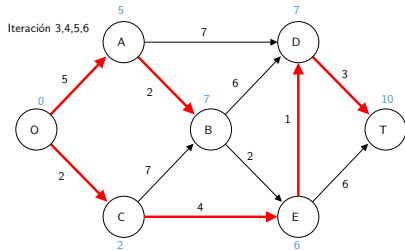
Algorithm: Bellman Ford

Entrada : G: Grafo, w: pesos aristas,
s: vértice inicial

```

1 for  $v \in G.V$  do
2    $v.d \leftarrow \infty$ 
3    $v.\pi \leftarrow NULL$ 
4 end
5  $s.d \leftarrow 0$ 
6 for  $i \leftarrow 0$  to  $|G.V| - 1$  do
7   for arista  $(u,v) \in G.E$  do
8     if  $v.d > u.d + w(u,v)$  then
9        $v.d \leftarrow u.d + w(u,v)$ 
10       $v.\pi \leftarrow u$ 
11    end
12  end
13 end
14 for arista  $(u,v) \in G.E$  do
15   if  $v.d > u.d + w(u,v)$  then
16     return False
17   end
18 end
19 return True
20 1-5: INICIALIZAR-UNICO-ORIGEN
21 8-11: RELAJACIÓN

```



Orden aristas:

OA, OC, AB, AD, CB, CE, BD, BE, DT, ED, ET

Ejercicio 3a: Bellman-Ford

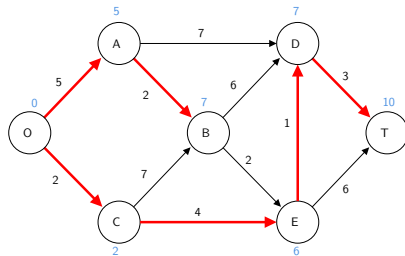
Algorithm: Bellman Ford

Entrada : G: Grafo, w: pesos aristas,
s: vértice inicial

```

1 for  $v \in G.V$  do
2    $v.d \leftarrow \infty$ 
3    $v.\pi \leftarrow NULL$ 
4 end
5  $s.d \leftarrow 0$ 
6 for  $i \leftarrow 0$  to  $|G.V| - 1$  do
7   for arista  $(u,v) \in G.E$  do
8     if  $v.d > u.d + w(u,v)$  then
9        $v.d \leftarrow u.d + w(u,v)$ 
10       $v.\pi \leftarrow u$ 
11    end
12  end
13 end
14 for arista  $(u,v) \in G.E$  do
15   if  $v.d > u.d + w(u,v)$  then
16     return False
17   end
18 end
19 return True
20 1-5: INICIALIZAR-UNICO-ORIGEN
21 8-11: RELAJACIÓN

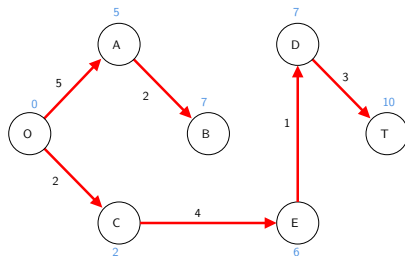
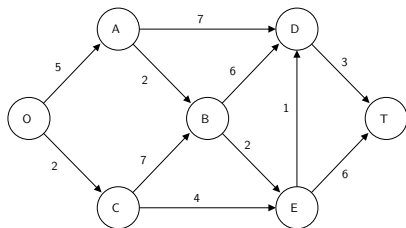
```



Orden aristas:

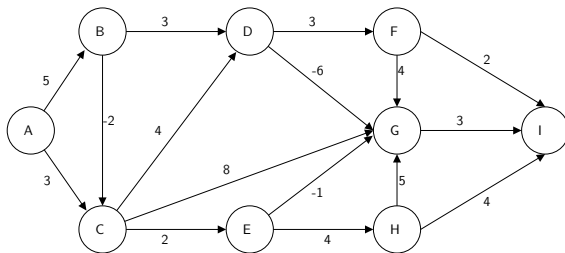
OA, OC, AB, AD, CB, CE, BD, BE, DT, ED, ET

Ejercicio 3a: Bellman-Ford



Ejercicio 4: Bellman-Ford

Encuentre el camino de peso mínimo entre el nodo A y el resto de los nodos de la siguiente red:



Ejercicio 4: Bellman-Ford

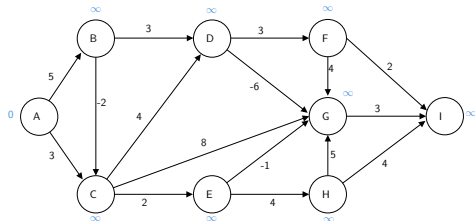
Algorithm: Bellman Ford

Entrada : G: Grafo, w: pesos aristas,
s: vértice inicial

```

1 for  $v \in G.V$  do
2    $v.d \leftarrow \infty$ 
3    $v.\pi \leftarrow NULL$ 
4 end
5  $s.d \leftarrow 0$ 
6 for  $i \leftarrow 0$  to  $|G.V| - 1$  do
7   for  $arista(u,v) \in G.E$  do
8     if  $v.d > u.d + w(u,v)$  then
9        $v.d \leftarrow u.d + w(u,v)$ 
10       $v.\pi \leftarrow u$ 
11    end
12  end
13 end
14 for  $arista(u,v) \in G.E$  do
15   if  $v.d > u.d + w(u,v)$  then
16     return False
17   end
18 end
19 return True
20 1-5: INICIALIZAR-UNICO-ORIGEN
21 8-11: RELAJACIÓN

```



Ejercicio 4: Bellman-Ford

Algorithm: Bellman Ford

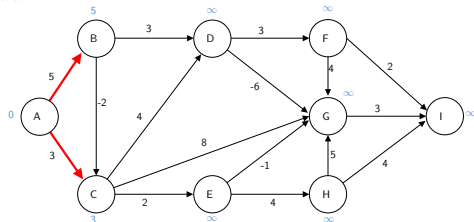
Entrada : G: Grafo, w: pesos aristas,
s: vértice inicial

```

1 for v ∈ G.V do
2   | v.d ← ∞
3   | v.π ← NULL
4 end
5 s.d ← 0
6 for i ← 0 to |G.V| - 1 do
7   | for arista (u,v) ∈ G.E do
8     |   if v.d > u.d + w(u, v) then
9       |     | v.d ← u.d + w(u, v)
10      |     | v.π ← u
11      |     end
12    | end
13 end
14 for arista (u,v) ∈ G.E do
15   | if v.d > u.d + w(u, v) then
16     |   return False
17   | end
18 end
19 return True
20 1-5: INICIALIZAR-UNICO-ORIGEN
21 8-11: RELAJACIÓN

```

Iteración 1



Orden aristas:

AB, AC, BD, BC, CD, CE, CG, DF, DG, EG, EH, FG, FI, GI, HG, HI

Ejercicio 4: Bellman-Ford

Algorithm: Bellman Ford

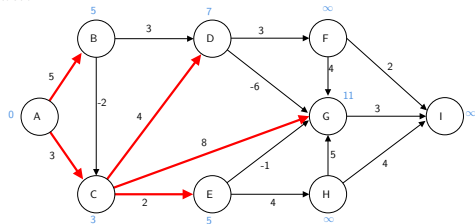
Entrada : G: Grafo, w: pesos aristas,
s: vértice inicial

```

1 for  $v \in G.V$  do
2    $v.d \leftarrow \infty$ 
3    $v.\pi \leftarrow NULL$ 
4 end
5  $s.d \leftarrow 0$ 
6 for  $i \leftarrow 0$  to  $|G.V| - 1$  do
7   for arista  $(u,v) \in G.E$  do
8     if  $v.d > u.d + w(u,v)$  then
9        $v.d \leftarrow u.d + w(u,v)$ 
10       $v.\pi \leftarrow u$ 
11    end
12  end
13 end
14 for arista  $(u,v) \in G.E$  do
15   if  $v.d > u.d + w(u,v)$  then
16     return False
17   end
18 end
19 return True
20 1-5: INICIALIZAR-UNICO-ORIGEN
21 8-11: RELAJACIÓN

```

Iteración 1



Orden aristas:

AB, AC, BD, BC, CD, CE, CG, DF, DG, EG, EH, FG, FI, GI, HG, HI

Ejercicio 4: Bellman-Ford

Algorithm: Bellman Ford

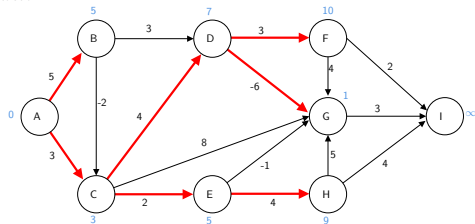
Entrada : G: Grafo, w: pesos aristas,
s: vértice inicial

```

1 for  $v \in G.V$  do
2    $v.d \leftarrow \infty$ 
3    $v.\pi \leftarrow NULL$ 
4 end
5  $s.d \leftarrow 0$ 
6 for  $i \leftarrow 0$  to  $|G.V| - 1$  do
7   for arista  $(u,v) \in G.E$  do
8     if  $v.d > u.d + w(u,v)$  then
9        $v.d \leftarrow u.d + w(u,v)$ 
10       $v.\pi \leftarrow u$ 
11    end
12  end
13 end
14 for arista  $(u,v) \in G.E$  do
15   if  $v.d > u.d + w(u,v)$  then
16     return False
17   end
18 end
19 return True
20 1-5: INICIALIZAR-UNICO-ORIGEN
21 8-11: RELAJACIÓN

```

Iteración 1



Orden aristas:
AB, AC, BD, BC, CD, CE, CG, **DF, DG, EG, EH**, FG, FI, GI, HG, HI

Ejercicio 4: Bellman-Ford

Algorithm: Bellman Ford

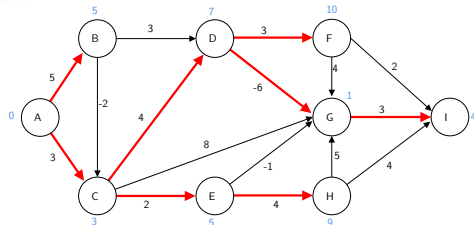
Entrada : G: Grafo, w: pesos aristas,
s: vértice inicial

```

1 for  $v \in G.V$  do
2    $v.d \leftarrow \infty$ 
3    $v.\pi \leftarrow NULL$ 
4 end
5  $s.d \leftarrow 0$ 
6 for  $i \leftarrow 0$  to  $|G.V| - 1$  do
7   for arista  $(u,v) \in G.E$  do
8     if  $v.d > u.d + w(u,v)$  then
9        $v.d \leftarrow u.d + w(u,v)$ 
10       $v.\pi \leftarrow u$ 
11    end
12  end
13 end
14 for arista  $(u,v) \in G.E$  do
15   if  $v.d > u.d + w(u,v)$  then
16     return False
17   end
18 end
19 return True
20 1-5: INICIALIZAR-UNICO-ORIGEN
21 8-11: RELAJACIÓN

```

Iteración 1



Orden aristas:
AB, AC, BD, BC, CD, CE, CG, DF, DG, EG, EH, **FG, FI, GI, HG, HI**

Ejercicio 4: Bellman-Ford

Algorithm: Bellman Ford

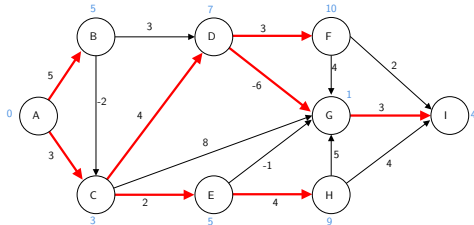
Entrada : G: Grafo, w: pesos aristas,
s: vértice inicial

```

1 for  $v \in G.V$  do
2    $v.d \leftarrow \infty$ 
3    $v.\pi \leftarrow NULL$ 
4 end
5  $s.d \leftarrow 0$ 
6 for  $i \leftarrow 0$  to  $|G.V| - 1$  do
7   for arista  $(u,v) \in G.E$  do
8     if  $v.d > u.d + w(u,v)$  then
9        $v.d \leftarrow u.d + w(u,v)$ 
10       $v.\pi \leftarrow u$ 
11    end
12  end
13 end
14 for arista  $(u,v) \in G.E$  do
15   if  $v.d > u.d + w(u,v)$  then
16     return False
17   end
18 end
19 return True
20 1-5: INICIALIZAR-UNICO-ORIGEN
21 8-11: RELAJACIÓN

```

Iteración 2,3,4,5,6,7,8



Orden aristas:

AB, AC, BD, BC, CD, CE, CG, DF, DG, EG, EH, FG, FI, GI, HG, HI

Ejercicio 4: Bellman-Ford

Algorithm: Bellman Ford

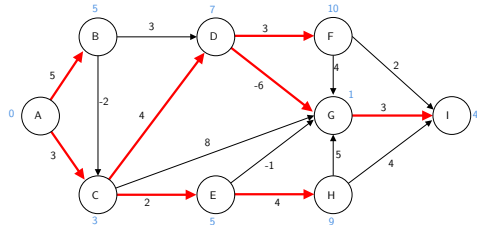
Entrada : G: Grafo, w: pesos aristas,
s: vértice inicial

```

1  for  $v \in G.V$  do
2     $v.d \leftarrow \infty$ 
3     $v.\pi \leftarrow NULL$ 
4  end
5   $s.d \leftarrow 0$ 
6  for  $i \leftarrow 0$  to  $|G.V| - 1$  do
7    for  $arista(u,v) \in G.E$  do
8      if  $v.d > u.d + w(u,v)$  then
9         $v.d \leftarrow u.d + w(u,v)$ 
10        $v.\pi \leftarrow u$ 
11     end
12   end
13 end
14 for  $arista(u,v) \in G.E$  do
15   if  $v.d > u.d + w(u,v)$  then
16     return False
17   end
18 end
19 return True
20 1-5: INICIALIZAR-UNICO-ORIGEN
21 8-11: RELAJACIÓN

```

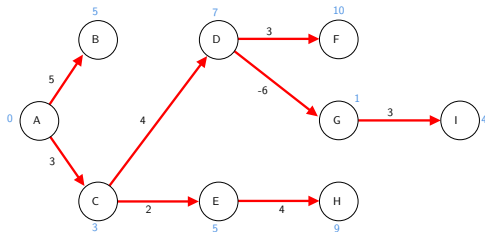
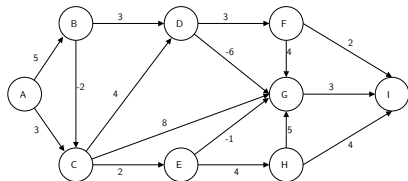
Iteración 2,3,4,5,6,7,8



Orden aristas:

AB, AC, BD, BC, CD, CE, CG, DF, DG, EG, EH, FG, FI, GI, HG, HI

Ejercicio 4: Bellman-Ford



Gracias por su atención
Dudas o consultas:
pgutierrez2018@udec.cl