Estructura de Datos

Maria C. Torres

Maria C. Torres

Ing. Electrónica (UNAL)
M.E. Ing. Eléctrica (UPRM)
Ph.D. Ciencias e Ingeniería de la Computación y la Información (UPRM)
Profesora asociada
Dpto. Ciencias de la Computación y la Decisión

mctorresm@unal.edu.co

HORARIO DE ATENCIÓN: Martes 10:00 am a 12:00 m – Oficina 313 M8A

Contenido del Curso

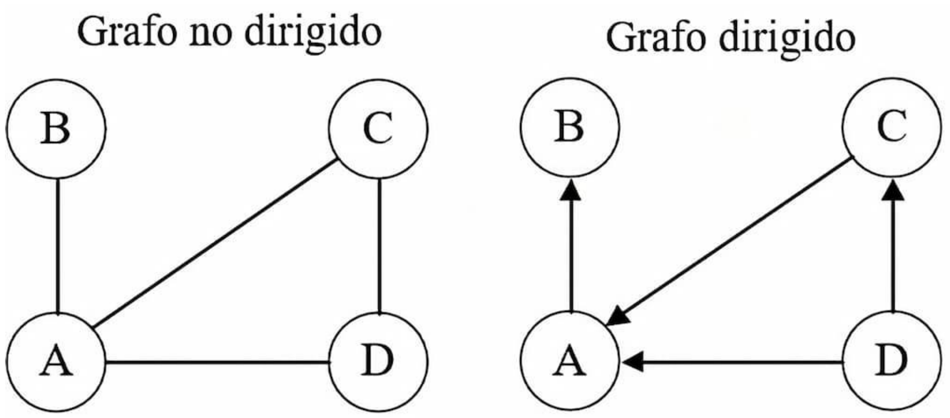
- Introducción: revisión fundamentos y POO
- Análisis de complejidad
- Arreglos
- Listas enlazadas
- Pilas y colas
- Heap
- Arboles binarios
- Tablas hash
- **Grafos**

Grafos

- ❑ Listas y matrices de adyacencia
- ❑ Recorrido
- ❑ Árbol de expansión mínima
- ❑ **Algoritmos del camino más corto**

Grafos

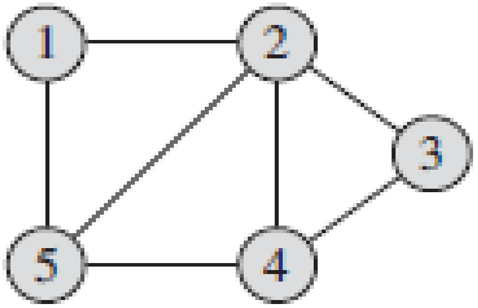
- Un grafo $G = (V, E)$ se define por:
  - Un conjunto de n vértices V, a los cuales se hace referencia por sus índices
  - Un conjunto de m aristas E, que conectan vértices entre sí.
  - Una arista es un par de vértices, indicados de la forma $< i, j >$, que indica que el vértice i está conectado al vértice j.
- Dependiendo de si el orden de los vértices en las aristas importa o no tenemos dos clases de grafos:
  - Grafo dirigido: El orden importa, es decir $< i, j > \neq < j, i >$. Si el vértice i está conectado al vértice j, no implica que el vértice j está conectado al vértice i.
  - Grafo no dirigido: El orden no importa, $< i, j > = < j, i >$.

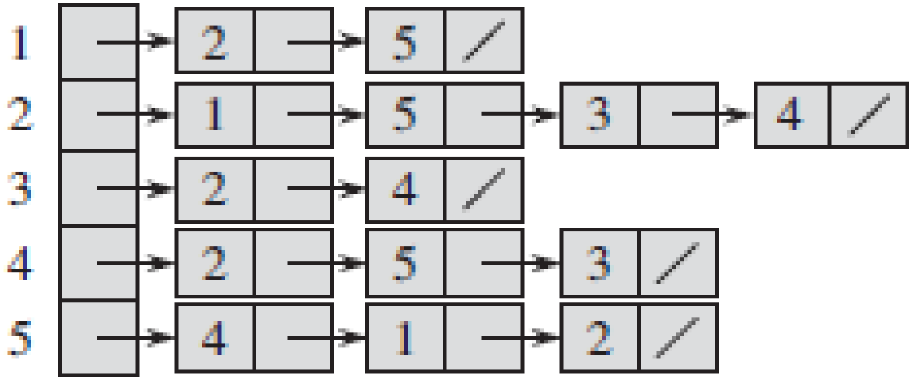


Grafos

Representación

- **<u>Listas de adyacencia</u>**: proporciona un camino compacto para representar grafos esparcidos, es decir grafos con pocas aristas.
  - Esta representación consiste en un arreglo Adj con $|V|$ listas, una para cada vértice.
  - Para cada $u \in V$, la lista de adyacencia $Adj[u]$ contiene todos los vértices v tal que exista la arista $< u, v >$.
  - Si G es un grafo dirigido, la suma de las longitudes de todas las listas de adyacencia es $|E|$.
  - Si G es un grafo no dirigido, la suma de las longitudes de todas las listas de adyacencia es $2|E|$.



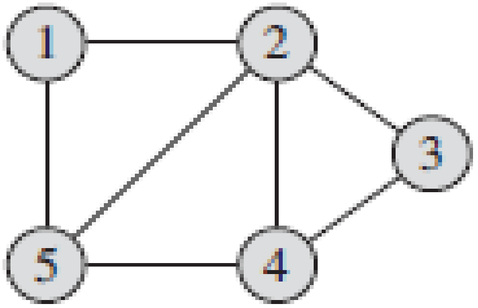


Grafos

Representación

- **<u>Matriz de adyacencia</u>**: Cuando el grafo es denso, es decir tiene muchas aristas, o deseamos saber rápidamente si dos nodos están conectados, la representación más apropiada es la matriz de adyacencia.
- Para esta representación de un grafo $G = (V, E)$ asumimos que los vértices están enumerados de $1 \ldots |V|$
- entonces la matriz de adyacencia es una matriz $|V| * |V|$ tal que:

$$a_{ij} = \begin{cases} 1 \quad si < i, j > \in E \\ 0 \; en \; otro \; caso \end{cases}$$

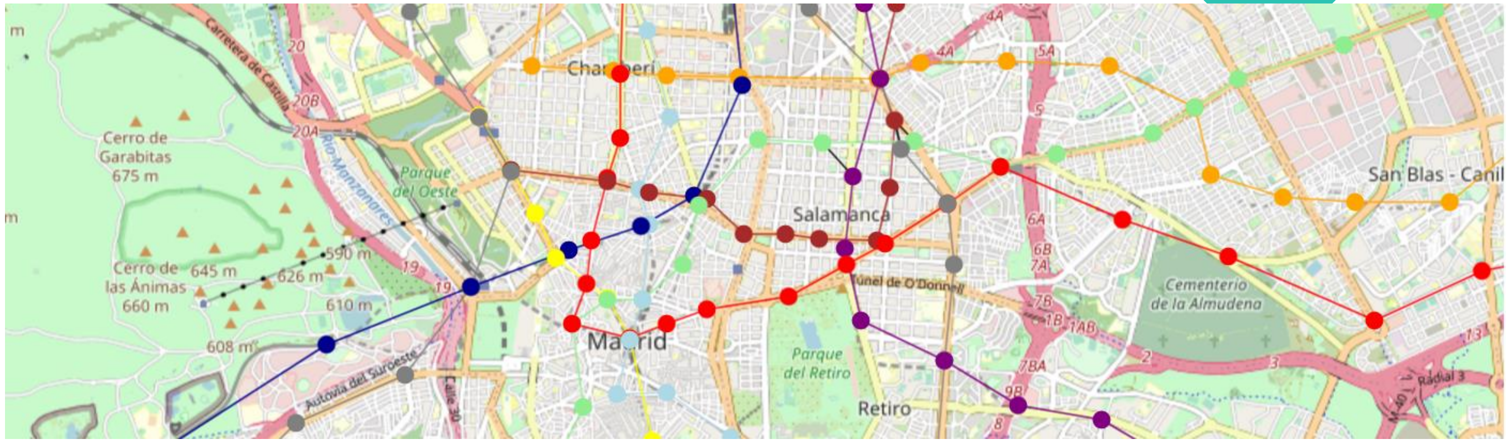


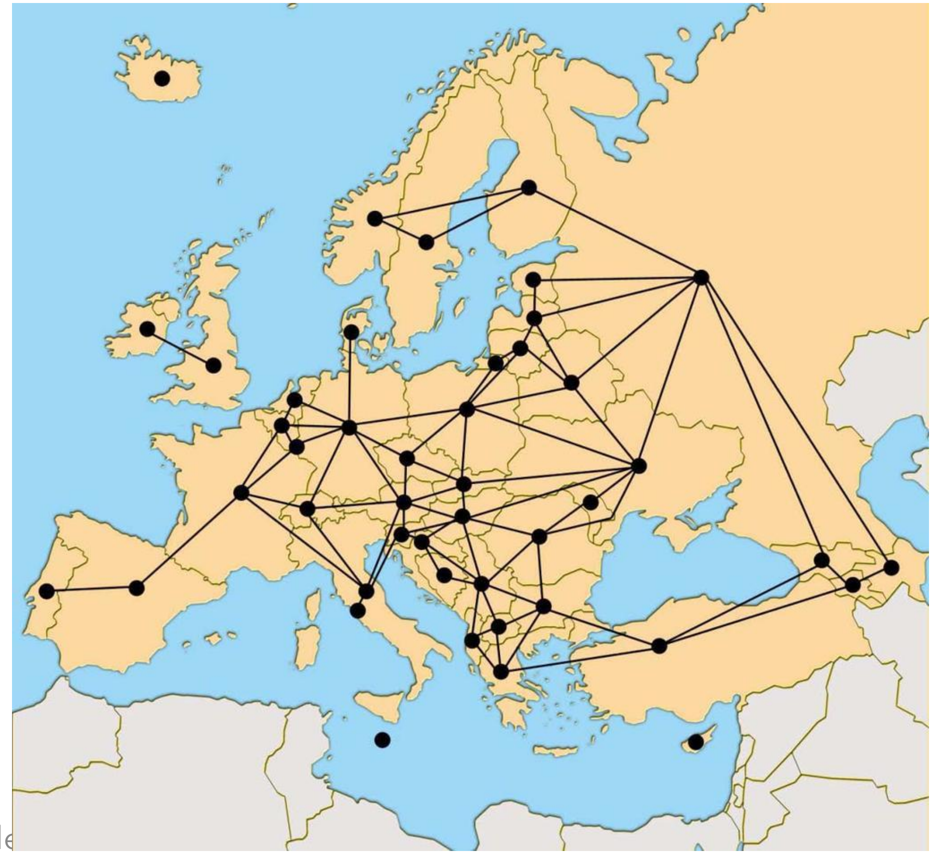
	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

Grafos

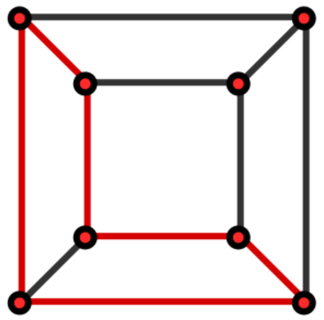
Algunas aplicaciones:

Camino más corto



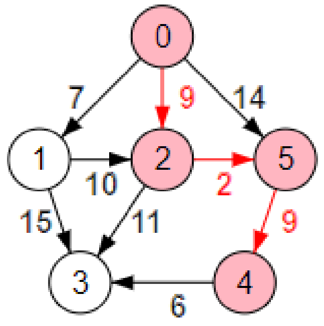


Grafos



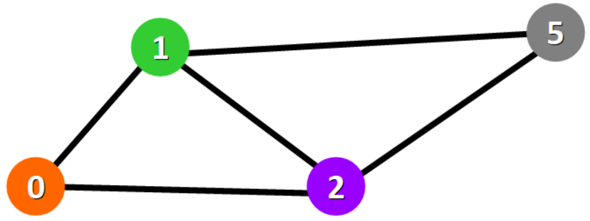
Ciclo

- Un camino simple donde el vértice inicial y el final son el mismo.



Costo de un camino

- suma de los costos o pesos de las aristas que recorre el camino, si el grafo no es ponderado cada arista tiene un peso igual a 1.



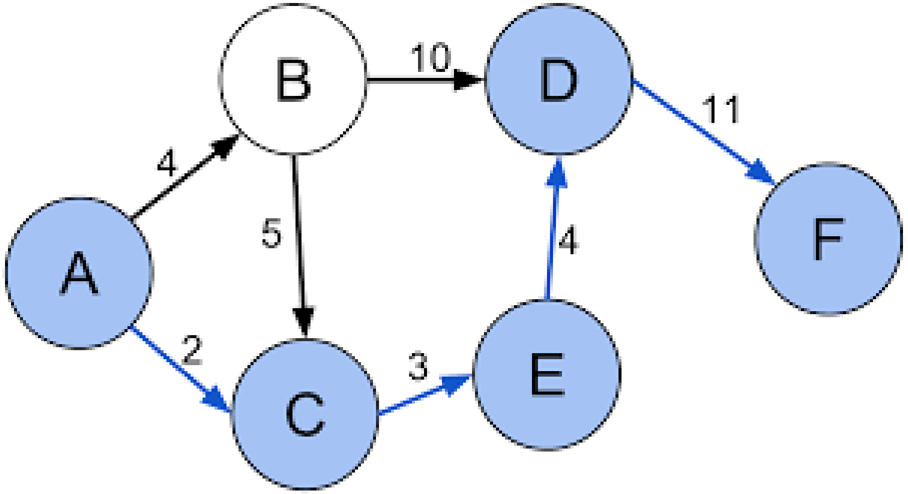
Ruta optima

- Camino de costo mínimo

Grafos

Descripción del problema

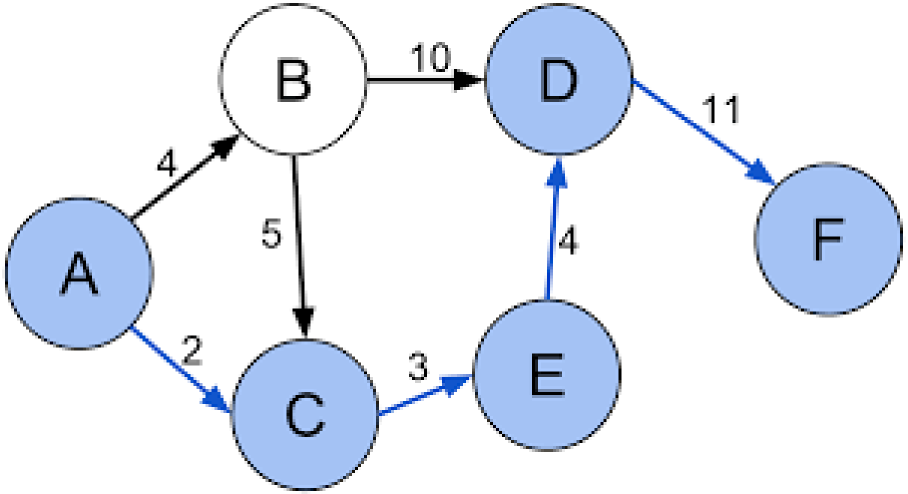
- Dado un grafo dirigido pesado encuentre
  - El camino más corto desde un vértice s a todos los demás vértices del grado (*Single source shortest paths*)
  - El camino más corto entre todos los pares del vértice del grafo (*All pairs shortest paths*)
- La longitud del camino corresponde a la suma de los pesos



Grafos

Descripción del problema

- El camino más corto desde un vértice s a todos los demás vértices del grado:
  - Bellman-Ford
  - Dijkstra
- El camino más corto entre todos los pares del vértice del grafo
  - Floyd-Warshall
  - Johnson



Grafos

Single source shortest paths

- $\delta(u, v)$: longitud del camino más corto entre los vértices u y v
- Para cada vértice se mantiene $d[u]$ que es la estimación de $\delta(s, u)$
- $d[u]$ se actualiza cuando se descubre un camino más corto, por tanto:

$$d[u] \geq \delta(u, v)$$

- $\pi(u)$ mantiene el predecesor de u en el camino más corto, permitiendo reconstruir toda la trayectoria

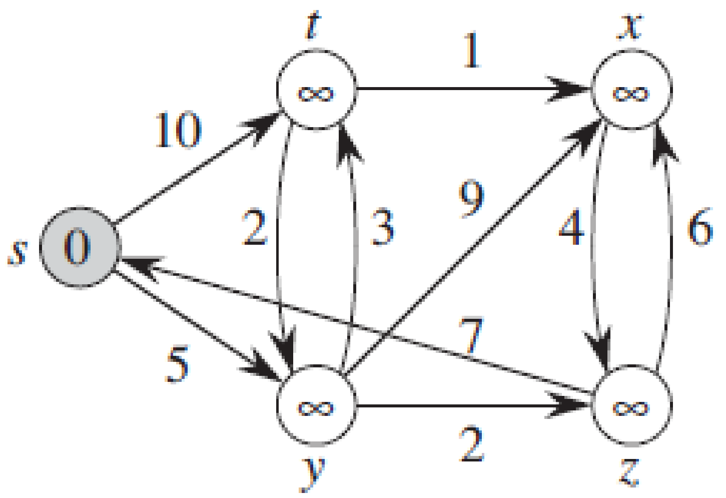
Grafos

Single source shortest paths - Inicialización

- Los dos algoritmos que vamos a estudiar emplean una técnica denomina relajación, encarga de actualizar $d[u]$ y $\pi(u)$ cuando se descubre un camino más corto
- Para hacer uso de esta operación, debemos inicializar $d[u]$ y $\pi(u)$ para todos los nodos u de nuestro grafo de la siguiente forma:

INITIALIZE_SINGLE_SOURCE(G,s)

1. FOR each $u \in G$
2. $\qquad d[u] = \infty$
3. $\qquad \pi(u) = null$
4. $d[u] = 0$



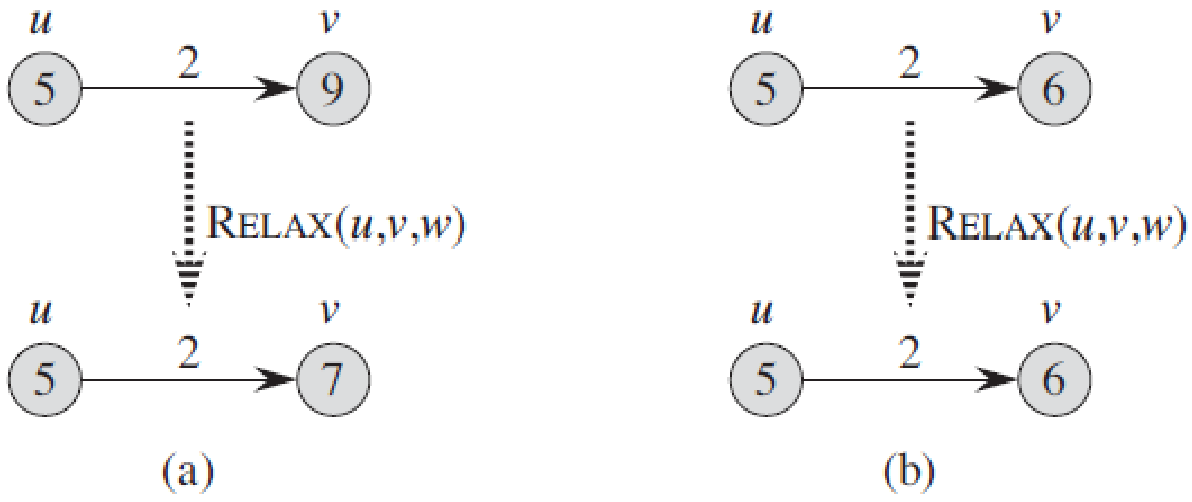
Grafos

Single source shortest paths – Relajación

- Relajar una arista (u, v) consiste en probar si podemos mejorar el camino mas cortos hasta v pasando a través de u

RELAX(u,v,W)

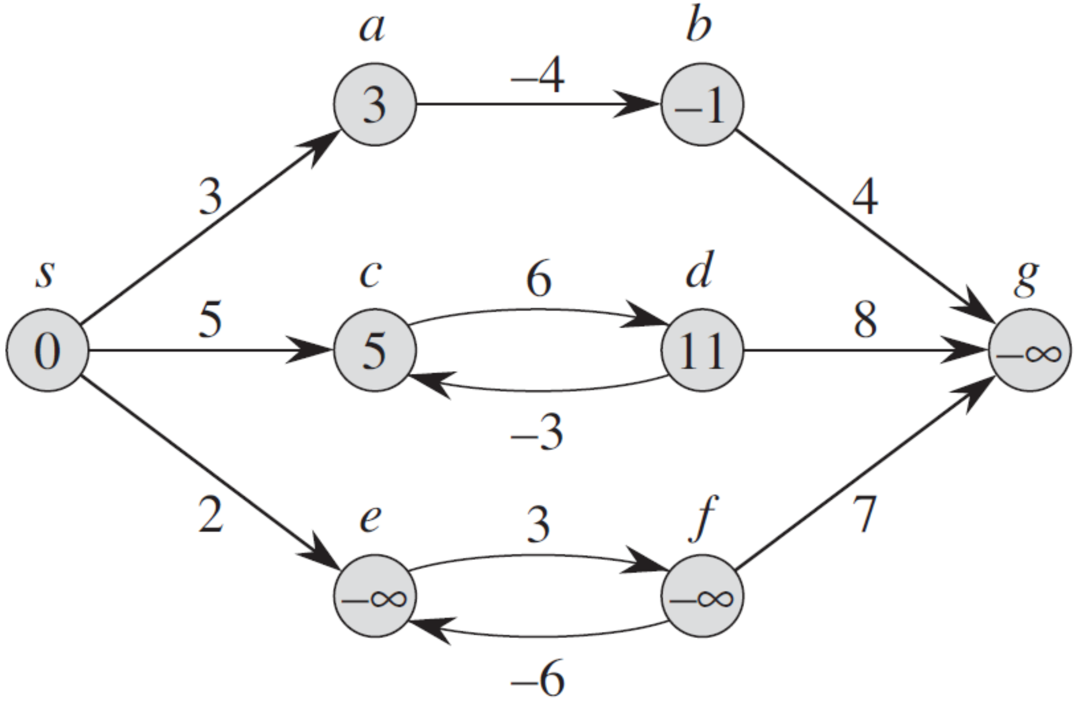
1. IF $d[v] > d[u] + W(u, v)$
2. $\quad d[v] = d[u] + W(u,$
3. $\quad \pi(v) = v$



Grafos

Single source shortest paths – Bellman-Ford

- Soluciona el problema SSSP considerando pesos negativos
- Este algoritmo permite determinar si hay un ciclo negativo en el grafo, en cuyo caso el problema del SSSP no tiene solución



Grafos

Single source shortest paths – Bellman-Ford

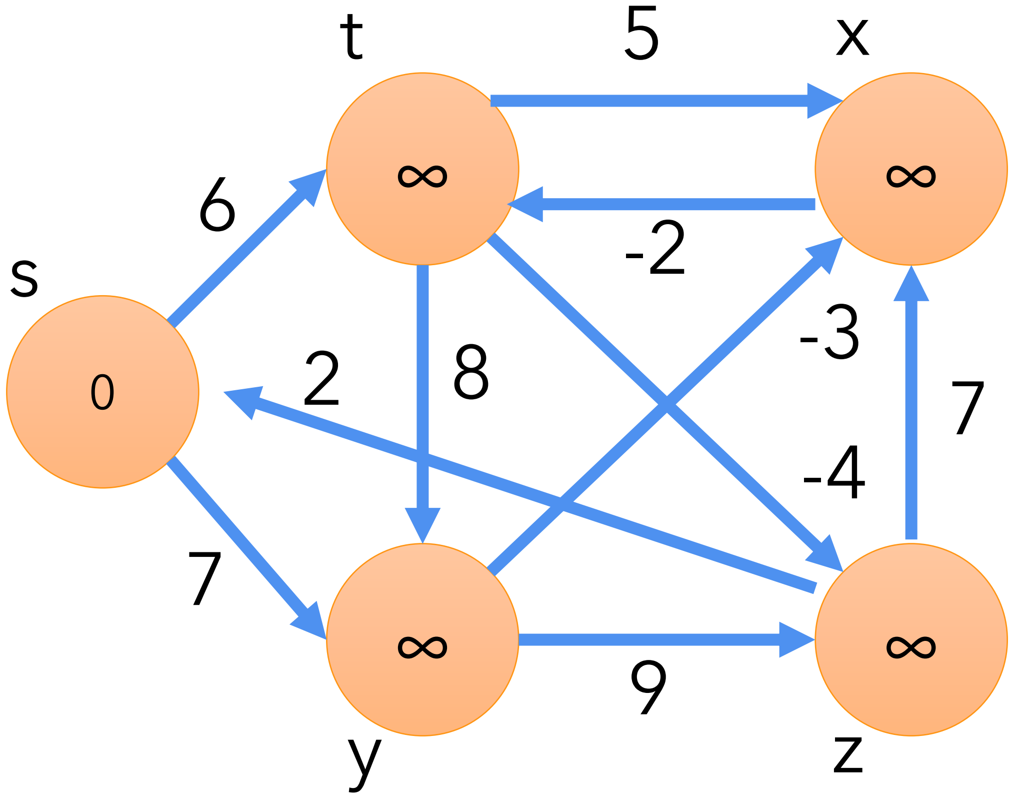
- Considere el grafo pesado $G = (V, E)$, donde los pesos están representados en una matriz W y un vértice para iniciar el camino s

```
BELLMAN-FORD(G, W, s)
1. FOR i = 1 TO |V|-1
2.      FOR each (u,v)
3.           RELAX(u,v,W)
4. FOR each (u,v)
5.      IF d[v] > d[u] + W(u,v)
6.           RETURN false
7. RETURN true
```

O(VE)

Grafos

Single source shortest paths – Bellman-Ford



	s	t	y	x	z
d[ ]	0	∞	∞	∞	∞
π[ ]	NULL	NULL	NULL	NULL	NULL

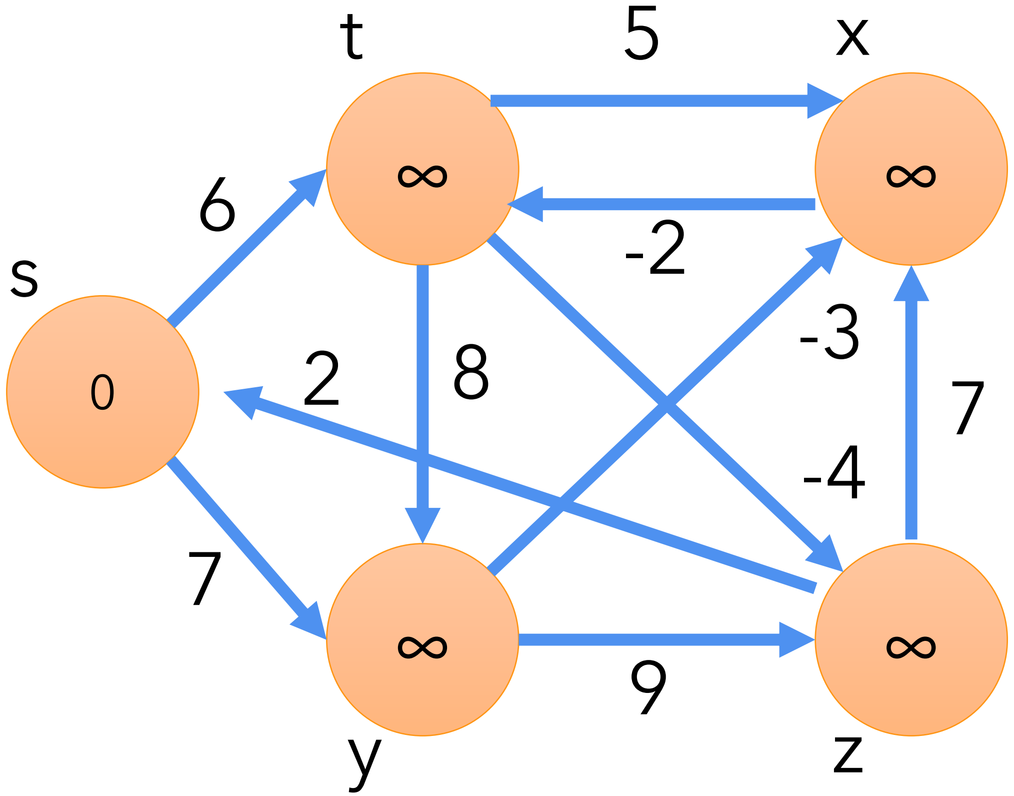
Grafos

Single source shortest paths – Bellman-Ford

i = 1

Relajar todos

los vértices



	s	t	y	x	z
d[ ]	0	∞	∞	∞	∞
π[ ]	NULL	NULL	NULL	NULL	NULL

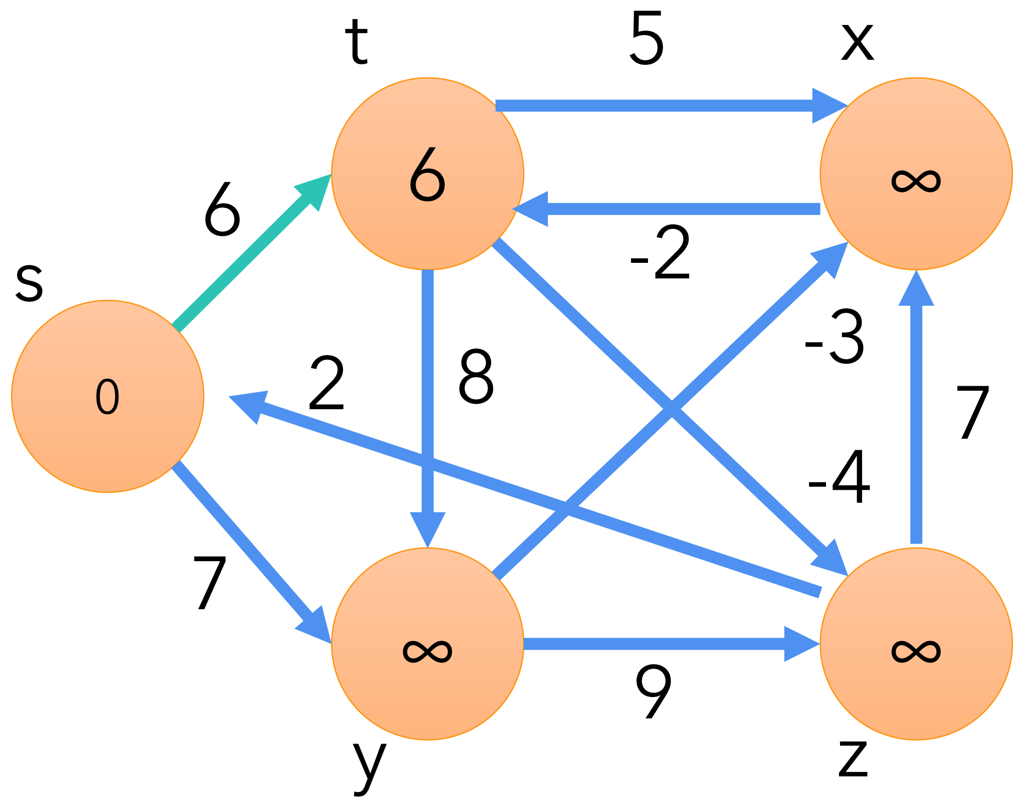
Grafos

Single source shortest paths – Bellman-Ford

i = 1

Relajar todos

los vértices



	s	t	y	x	z
d[ ]	0	6	∞	∞	∞
π[ ]	NULL	s	NULL	NULL	NULL

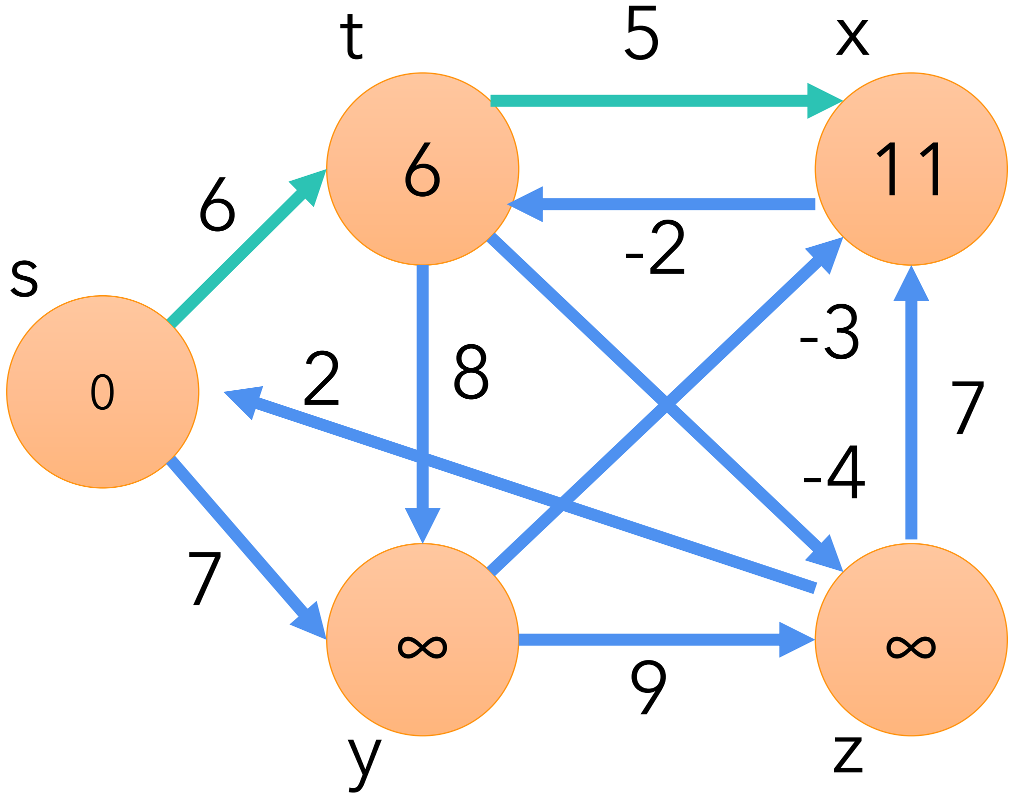
Grafos

Single source shortest paths – Bellman-Ford

i = 1

Relajar todos

los vértices



	s	t	y	x	z
d[ ]	0	6	∞	11	∞
π[ ]	NULL	s	NULL	t	NULL

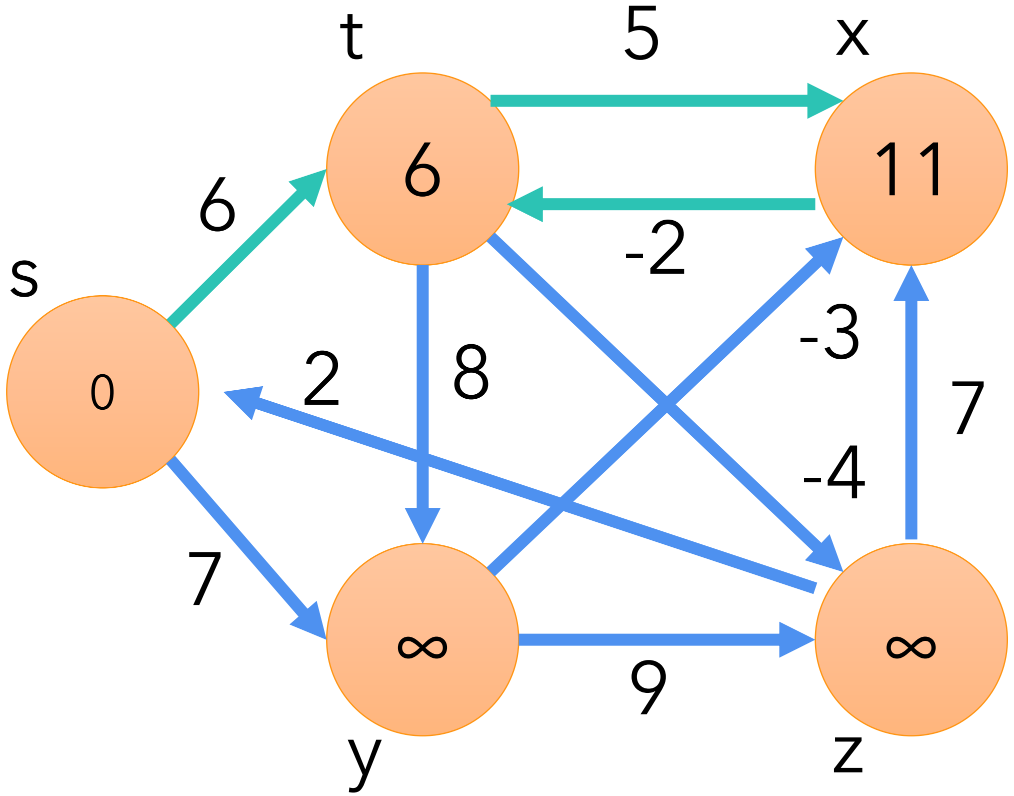
Grafos

Single source shortest paths – Bellman-Ford

i = 1

Relajar todos

los vértices



	s	t	y	x	z
d[ ]	0	6	∞	11	∞
π[ ]	NULL	s	NULL	t	NULL

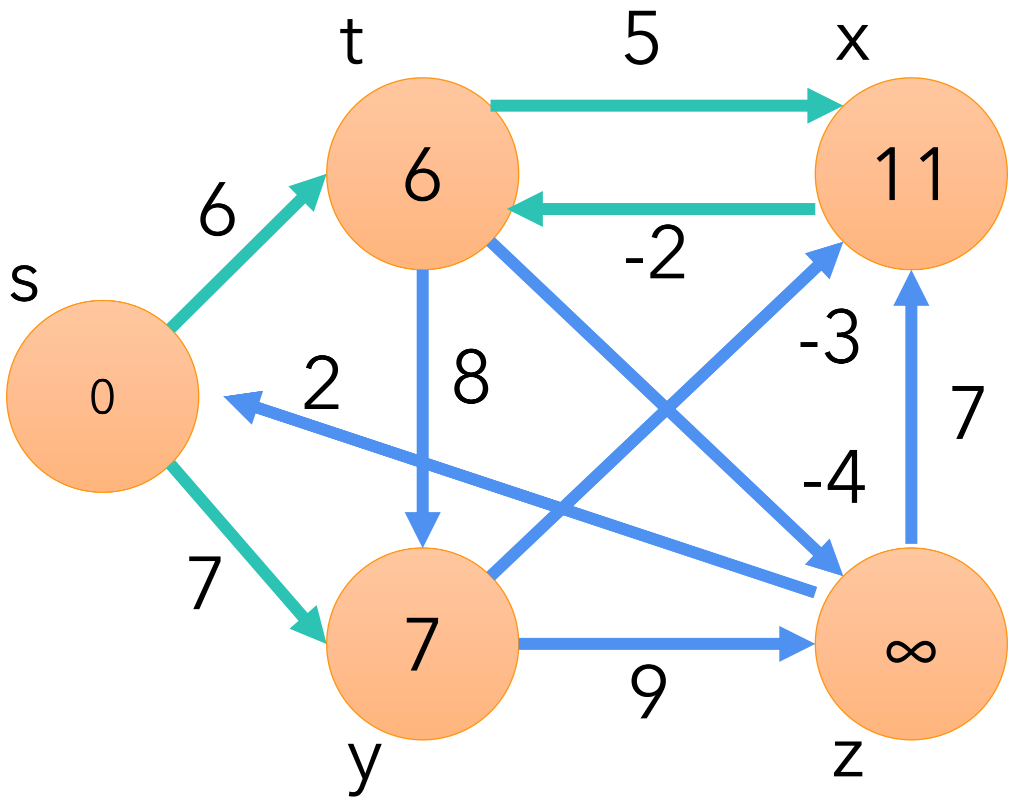
Grafos

Single source shortest paths – Bellman-Ford

i = 1

Relajar todos

los vértices



	s	t	y	x	z
d[ ]	0	6	7	11	∞
π[ ]	NULL	s	s	t	NULL

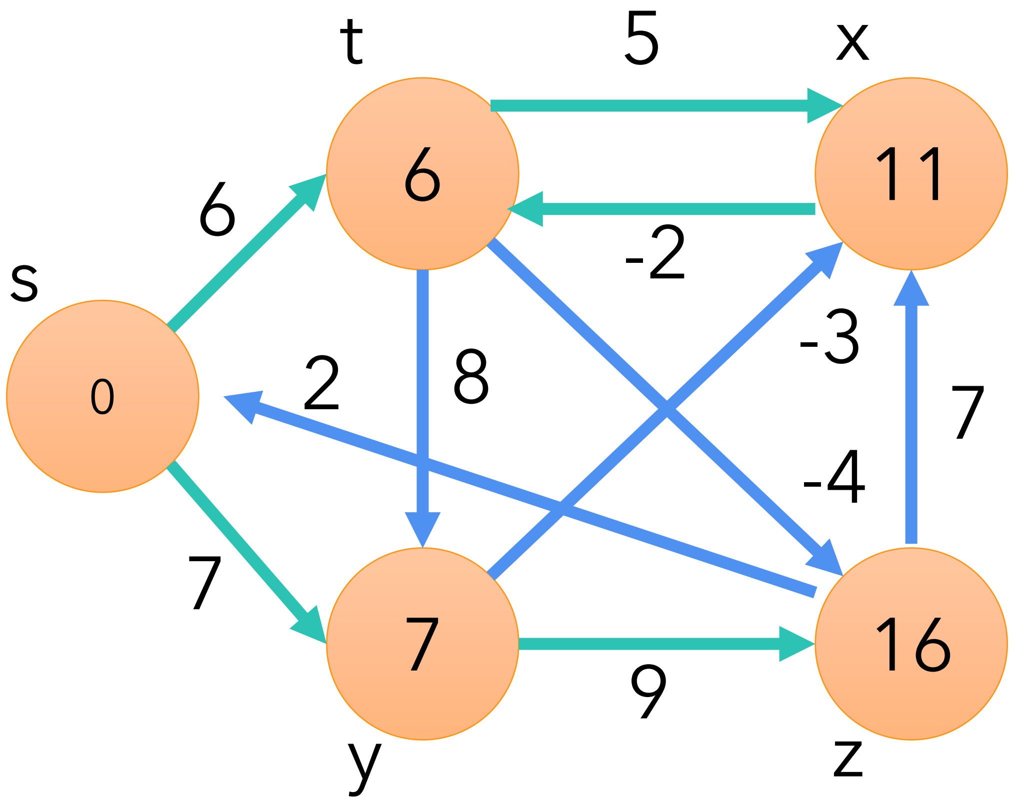
Grafos

Single source shortest paths – Bellman-Ford

i = 1

Relajar todos

los vértices



	s	t	y	x	z
d[ ]	0	6	7	11	16
π[ ]	NULL	s	s	t	y

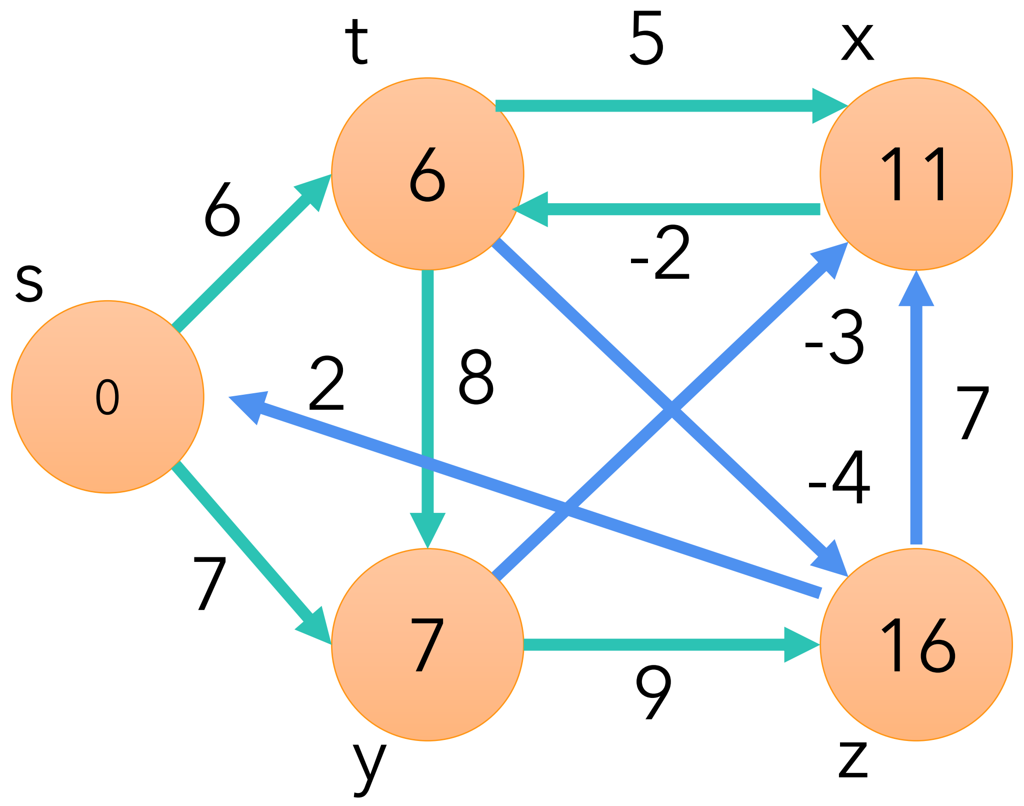
Grafos

Single source shortest paths – Bellman-Ford

i = 1

Relajar todos

los vértices



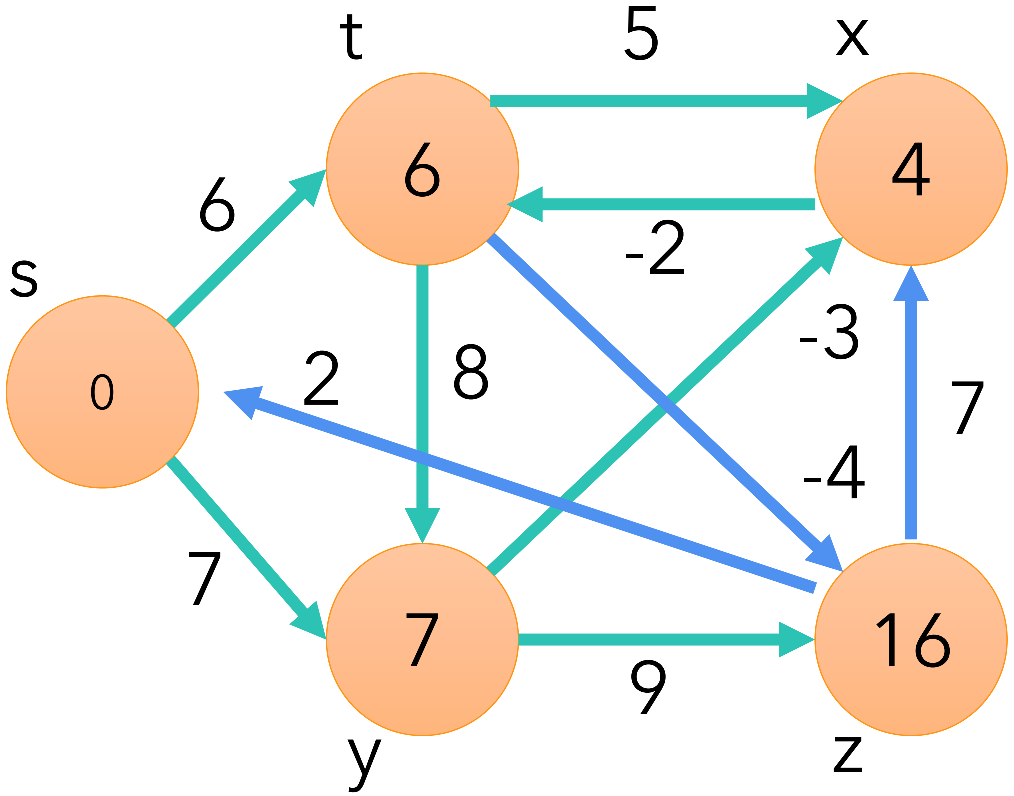
	s	t	y	x	z
d[ ]	0	6	7	11	16
π[ ]	NULL	s	s	t	y

Grafos

Single source shortest paths – Bellman-Ford

i = 1

Relajar todos los vértices



	s	t	y	x	z
d[ ]	0	6	7	4	16
π[ ]	NULL	s	s	y	y

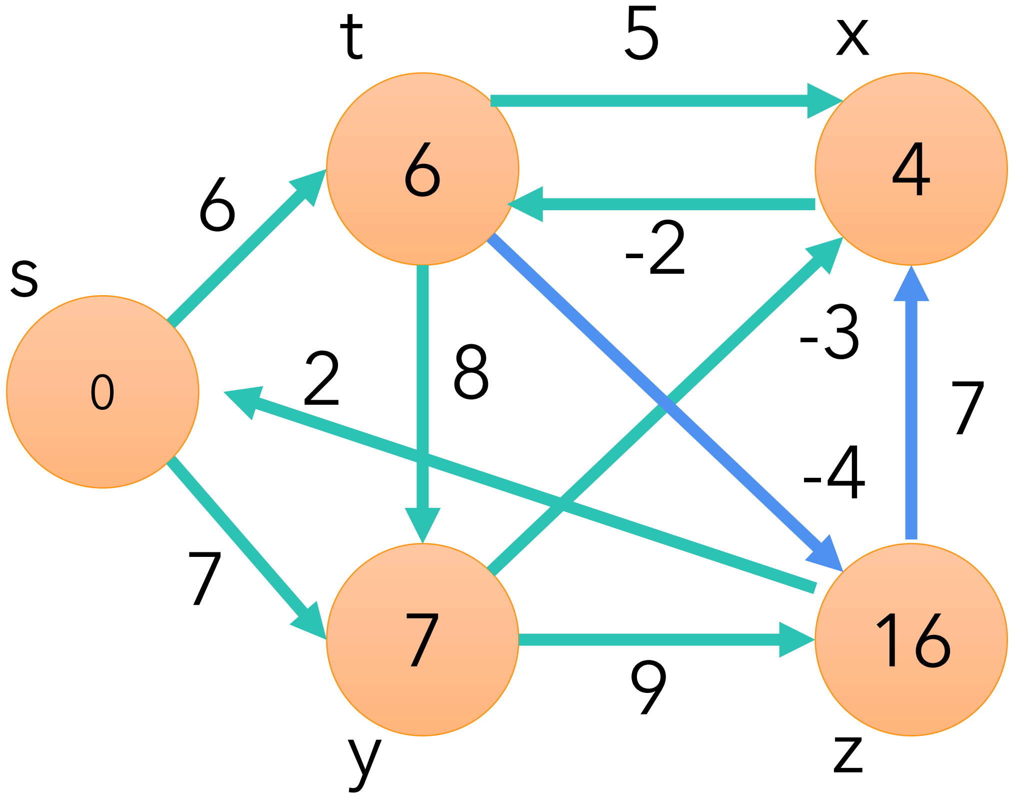
Grafos

Single source shortest paths – Bellman-Ford

i = 1

Relajar todos

los vértices



	s	t	y	x	z
d[ ]	0	6	7	4	16
π[ ]	NULL	s	s	y	y

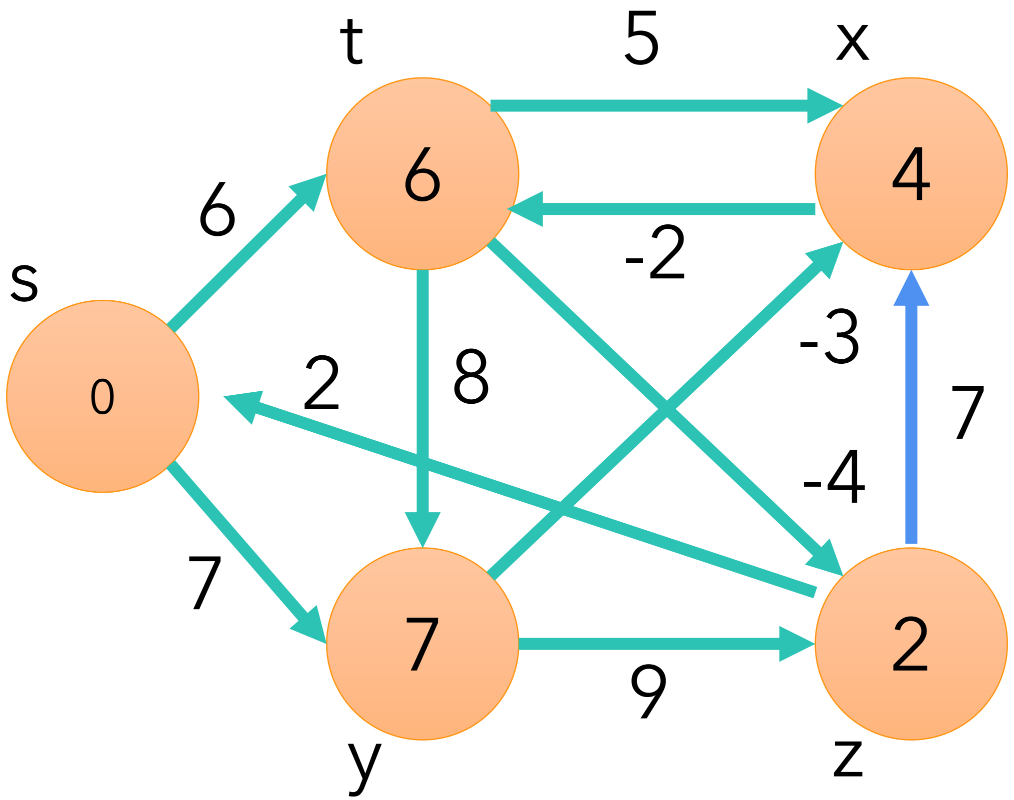
Grafos

Single source shortest paths – Bellman-Ford

i = 1

Relajar todos

los vértices



	s	t	y	x	z
d[ ]	0	6	7	4	2
π[ ]	NULL	s	s	y	t

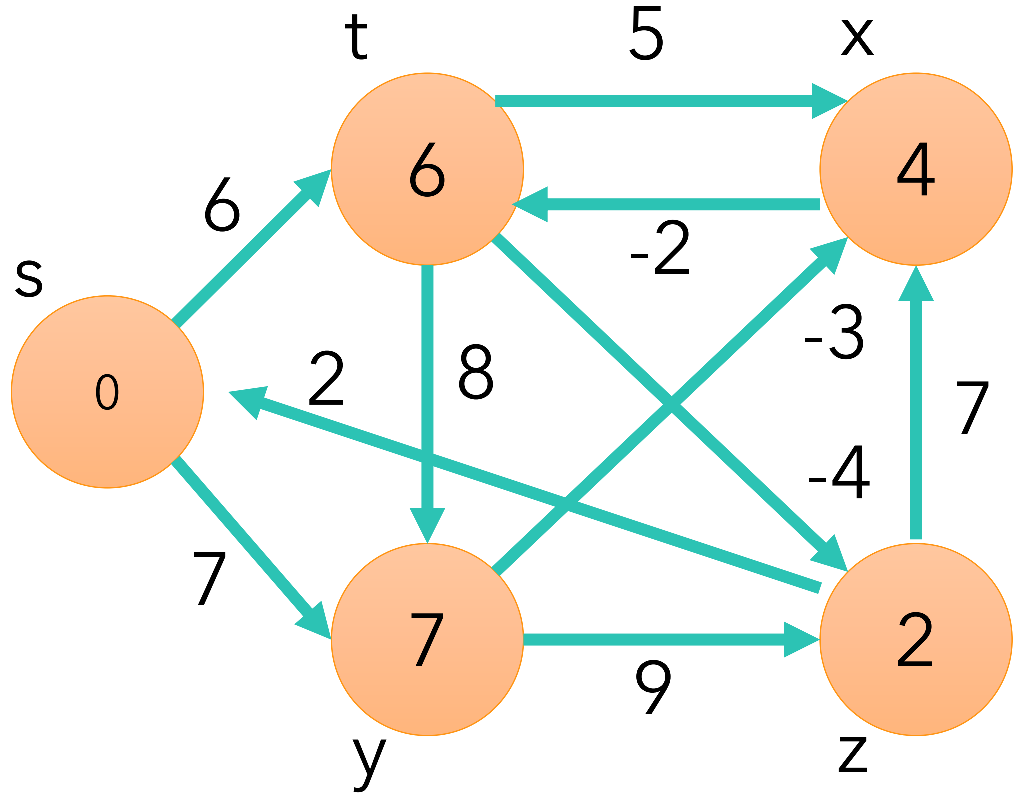
Grafos

Single source shortest paths – Bellman-Ford

i = 1

Relajar todos

los vértices



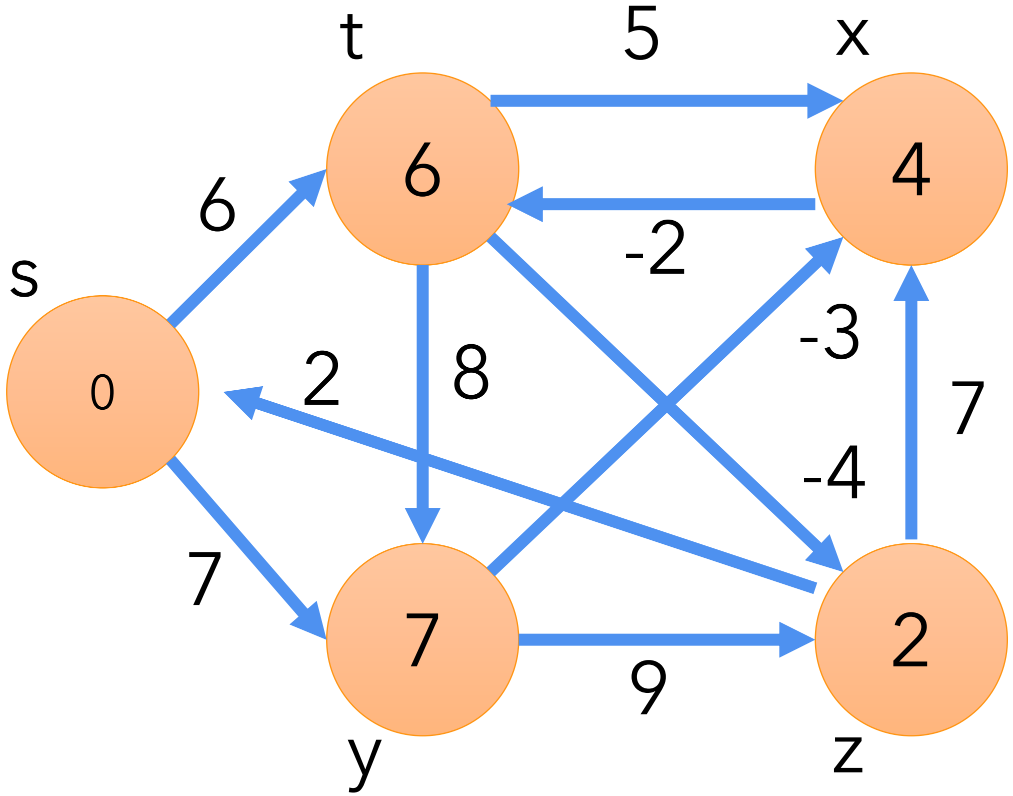
	s	t	y	x	z
d[ ]	0	6	7	4	2
π[ ]	NULL	s	s	y	t

Grafos

Single source shortest paths – Bellman-Ford

i = 2

Relajar todos los vértices



	s	t	y	x	z
d[ ]	0	6	7	4	2
π[ ]	NULL	s	s	y	t

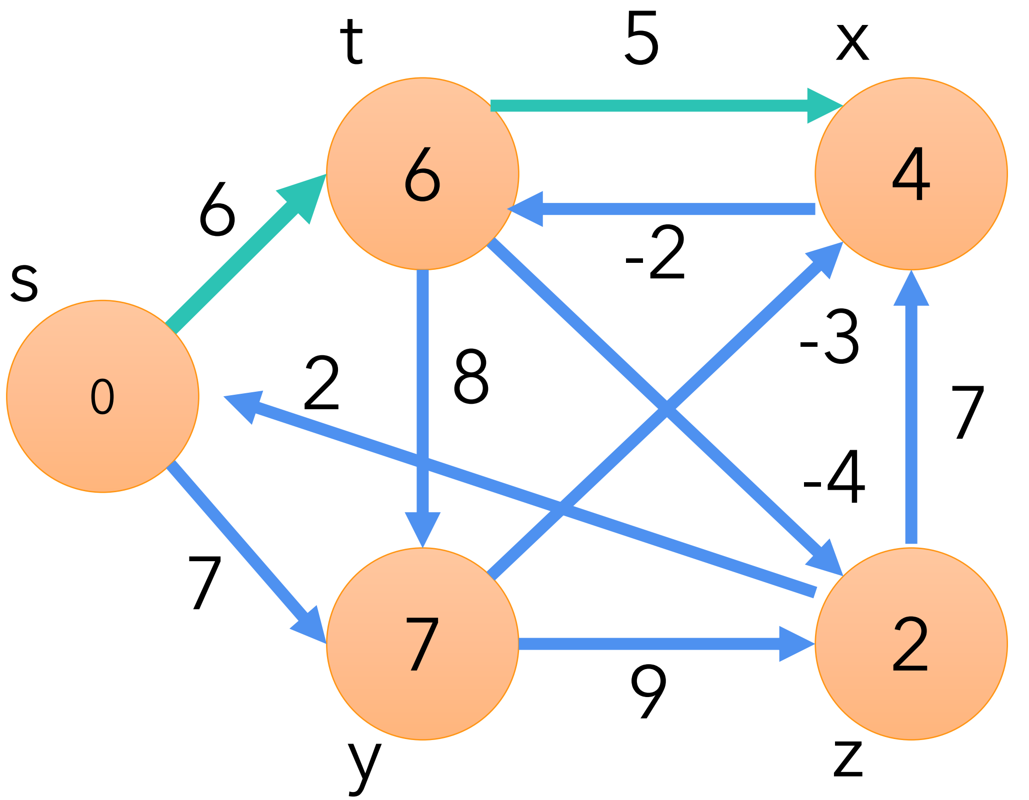
Grafos

Single source shortest paths – Bellman-Ford

i = 2

Relajar todos

los vértices



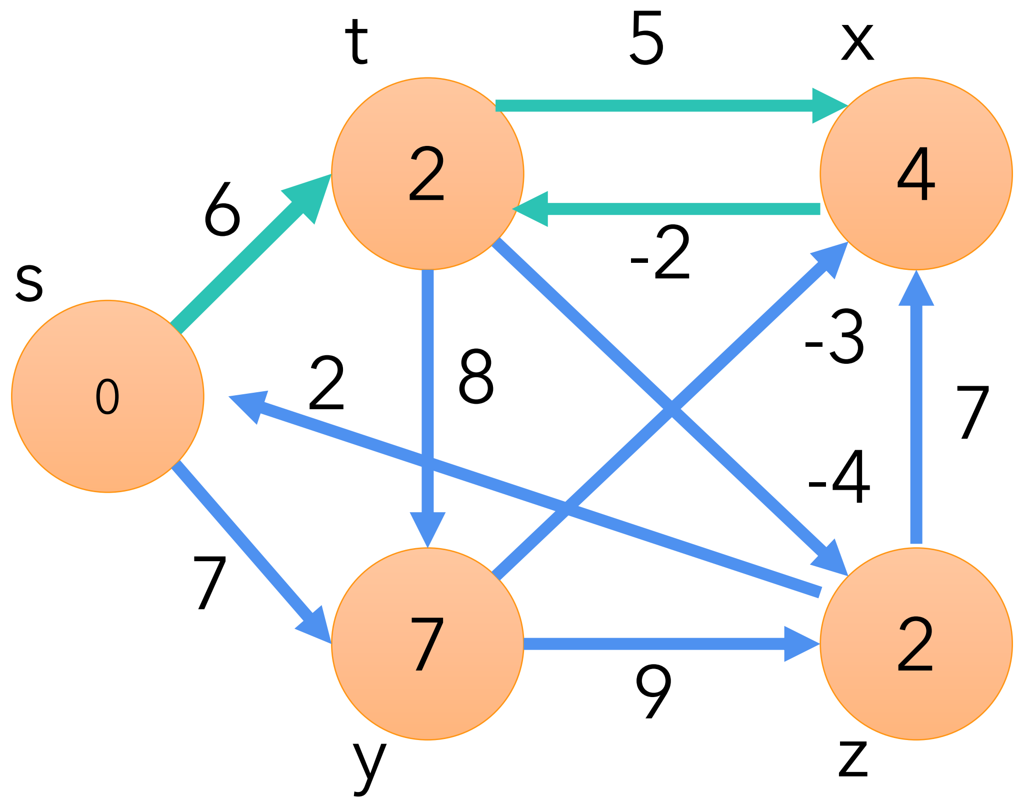
	s	t	y	x	z
d[ ]	0	6	7	4	2
π[ ]	NULL	s	s	y	t

Grafos

Single source shortest paths – Bellman-Ford

i = 2

Relajar todos los vértices



	s	t	y	x	z
d[ ]	0	2	7	4	2
π[ ]	NULL	x	s	y	t

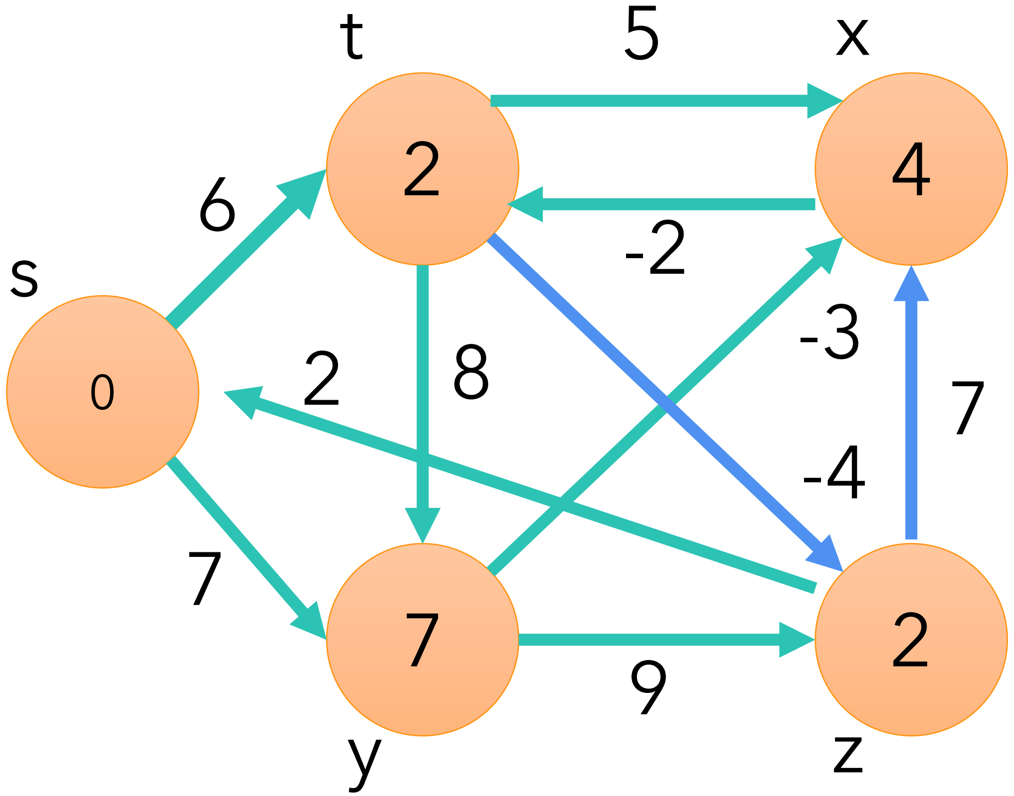
Grafos

Single source shortest paths – Bellman-Ford

i = 2

Relajar todos

los vértices



	s	t	y	x	z
d[ ]	0	2	7	4	2
π[ ]	NULL	x	s	y	t

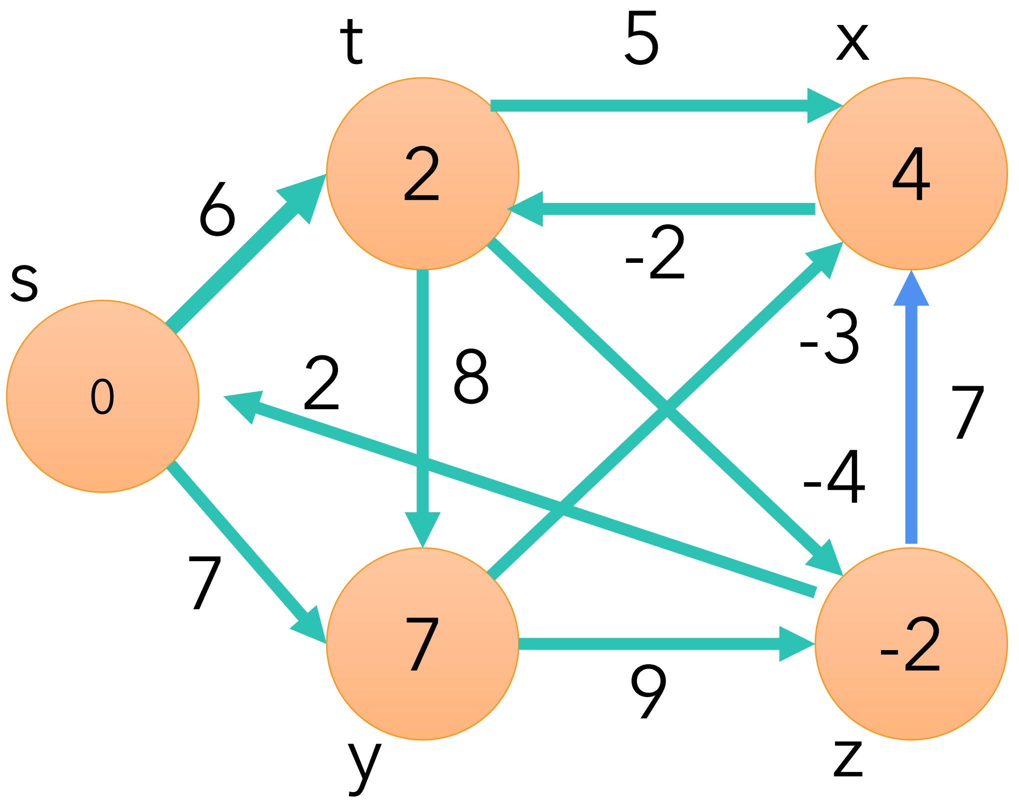
Grafos

Single source shortest paths – Bellman-Ford

i = 2

Relajar todos

los vértices



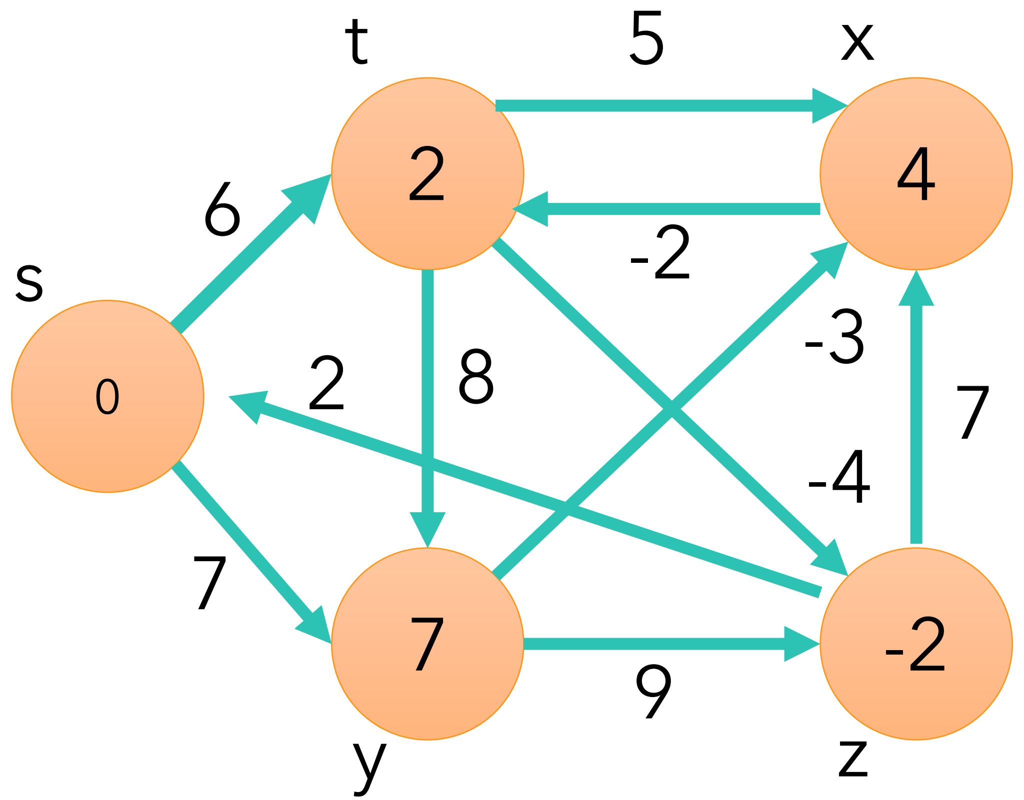
	s	t	y	x	z
d[ ]	0	2	7	4	-2
π[ ]	NULL	x	s	y	t

Grafos

Single source shortest paths – Bellman-Ford

i = 2

Relajar todos los vértices



	s	t	y	x	z
d[ ]	0	2	7	4	-2
π[ ]	NULL	x	s	y	t

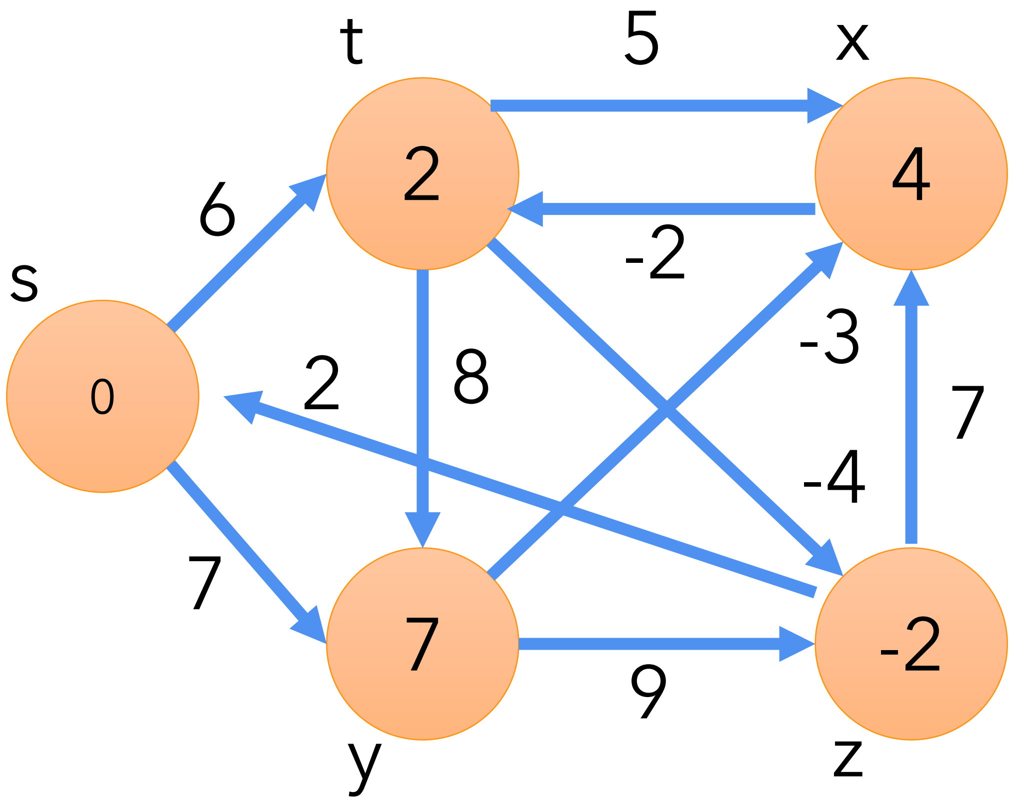
Grafos

Single source shortest paths – Bellman-Ford

i = 3

Relajar todos

los vértices



	s	t	y	x	z
d[ ]	0	2	7	4	-2
π[ ]	NULL	x	s	y	t

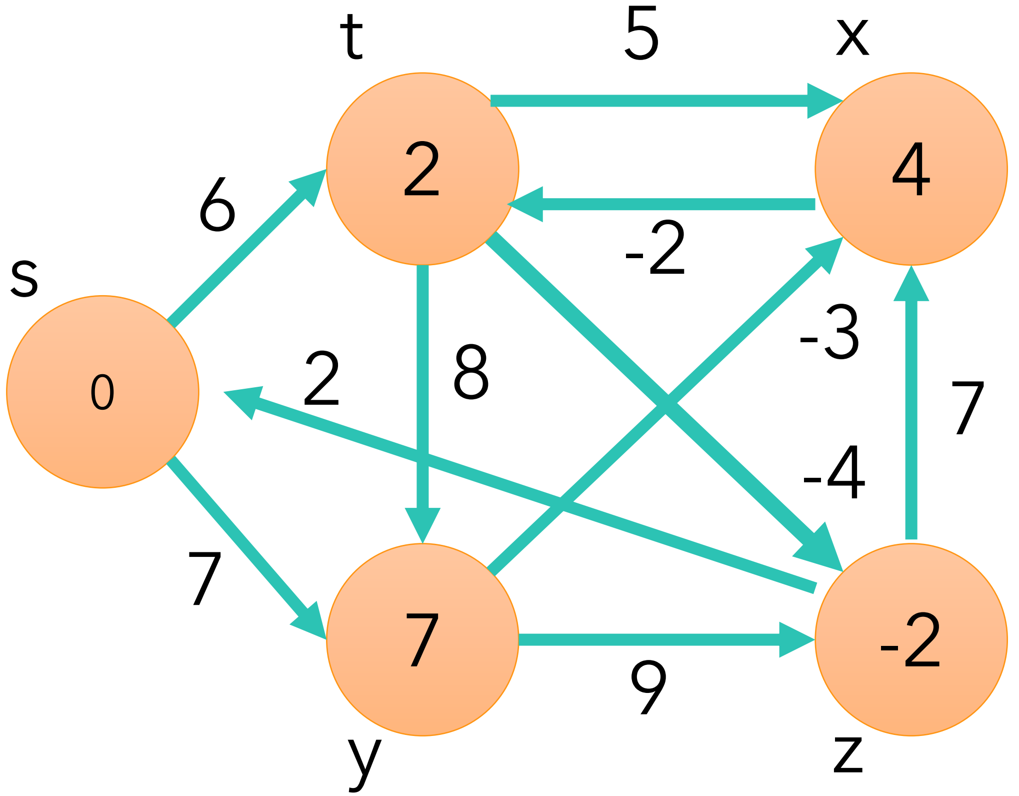
Grafos

Single source shortest paths – Bellman-Ford

i = 3

Relajar todos

los vértices



	s	t	y	x	z
d[ ]	0	2	7	4	-2
π[ ]	NULL	x	s	y	t

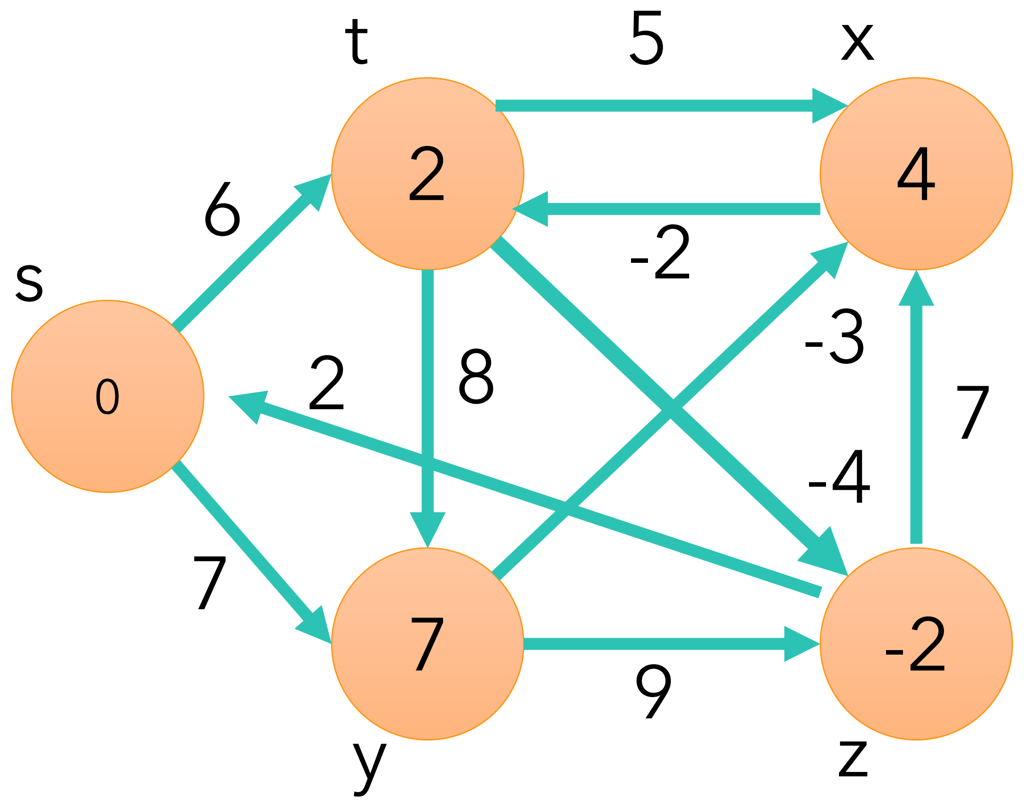
Grafos

Single source shortest paths – Bellman-Ford

i = 4

Relajar todos

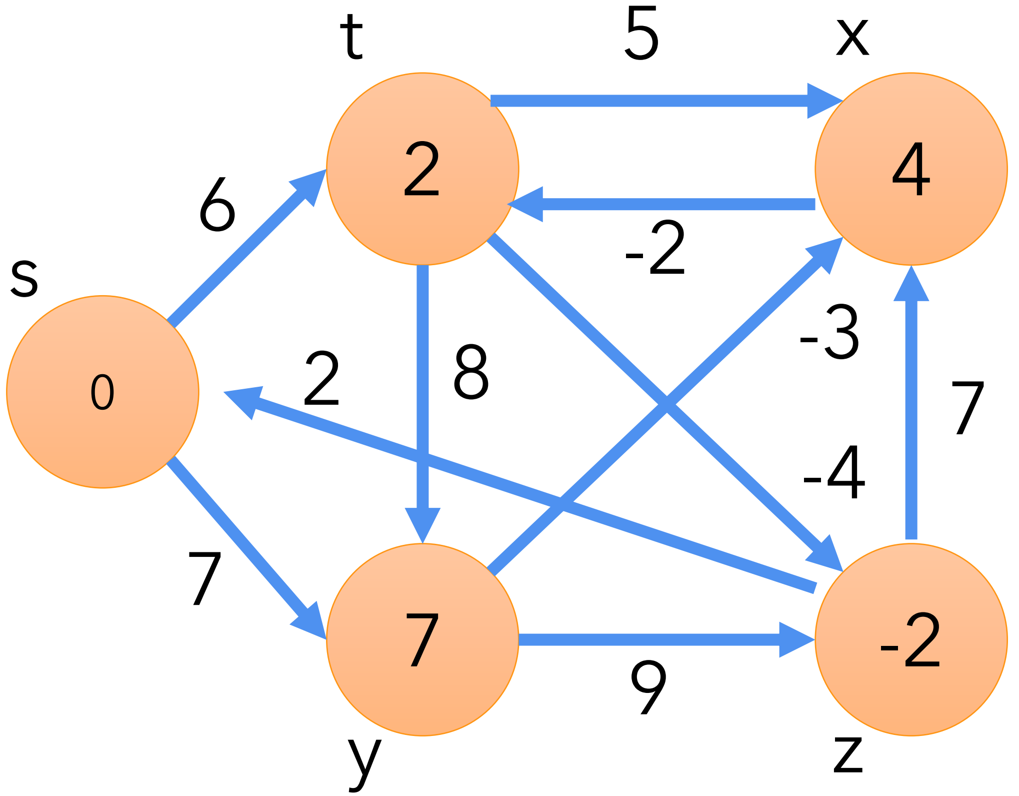
los vértices



	s	t	y	x	z
d[ ]	0	2	7	4	-2
π[ ]	NULL	x	s	y	t

Grafos

Single source shortest paths – Bellman-Ford



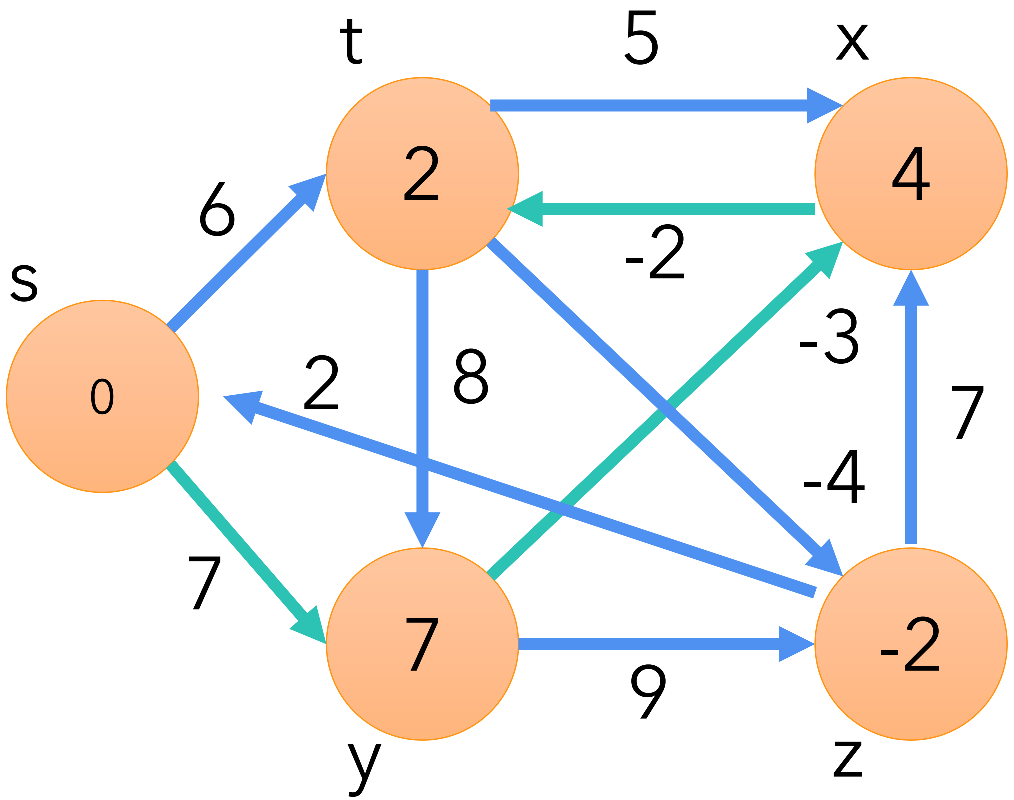
	s	t	y	x	z
d[ ]	0	2	7	4	-2
π[ ]	NULL	x	s	y	t

Grafos

Single source shortest paths – Bellman-Ford

Reconstruir el

camino mas corto (s, t)



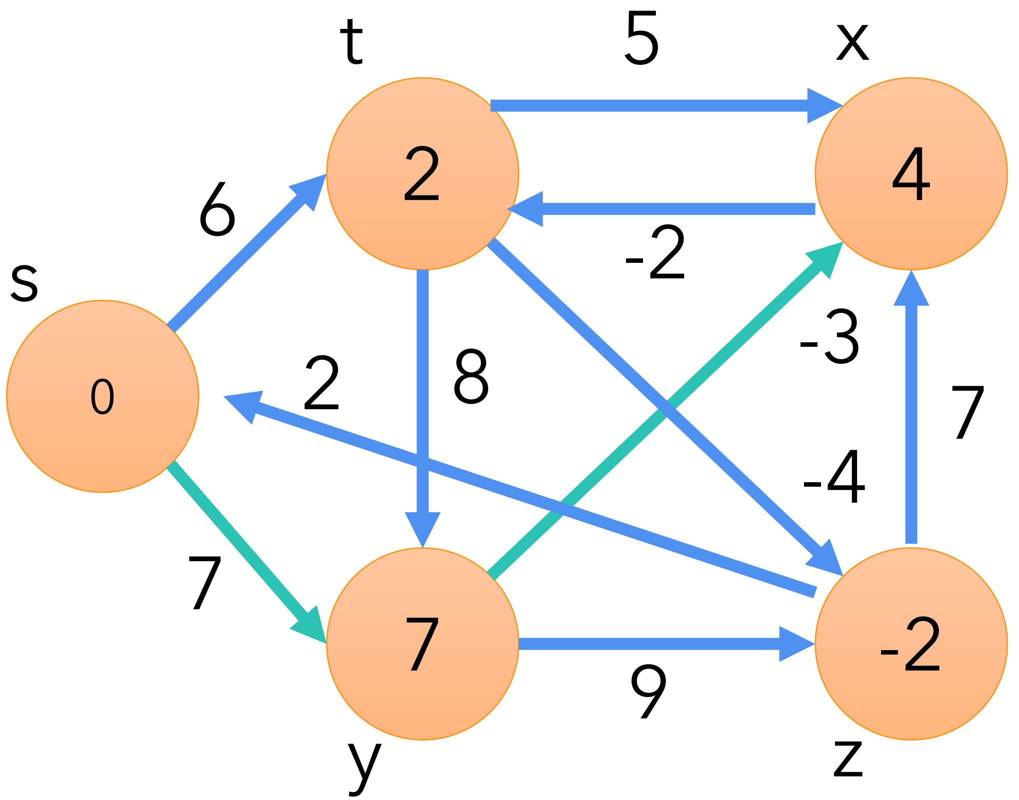
	s	t	y	x	z
d[ ]	0	2	7	4	-2
π[ ]	NULL	x	s	y	t

Grafos

Single source shortest paths – Bellman-Ford

Reconstruir el

camino mas corto (s, x)



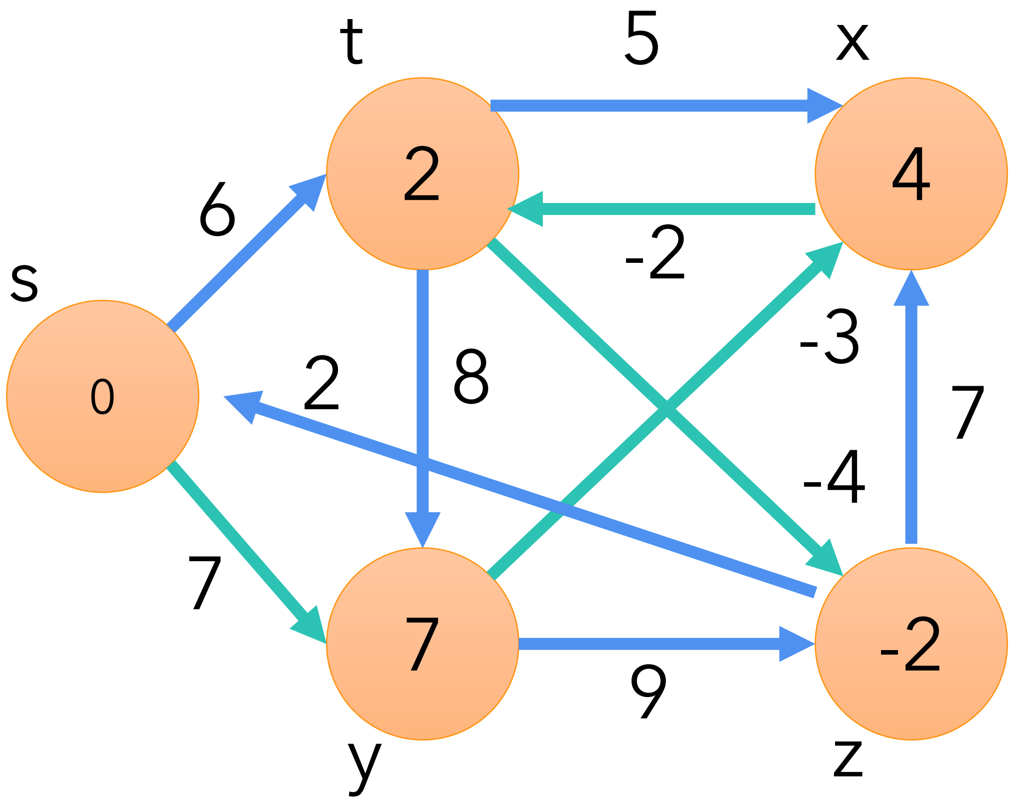
	s	t	y	x	z
d[ ]	0	2	7	4	-2
π[ ]	NULL	x	s	y	t

Grafos

Single source shortest paths – Bellman-Ford

Reconstruir el

camino mas corto (s, z)



	s	t	y	x	z
d[ ]	0	2	7	4	-2
π[ ]	NULL	x	s	y	t

Grafos

Single source shortest paths – Dijkstra

- Soluciona el problema SSSP para grafos sin pesos negativo
- Llega a ser más eficiente computacionalmente que el algoritmo de Bellman-Ford
- Emplea una cola de prioridad mínima, fácilmente implementada con un MIN-HEAP
- En la cola de prioridad mínima se mantienen los vértices cuyo camino más corto aún no ha sido determinado
- El algoritmo, iterativamente selecciona un vértice en la cola, y relaja todas las artistas que salen de este vértice
- La clave para organizar la cola de prioridad mínima es d[ ]

Grafos

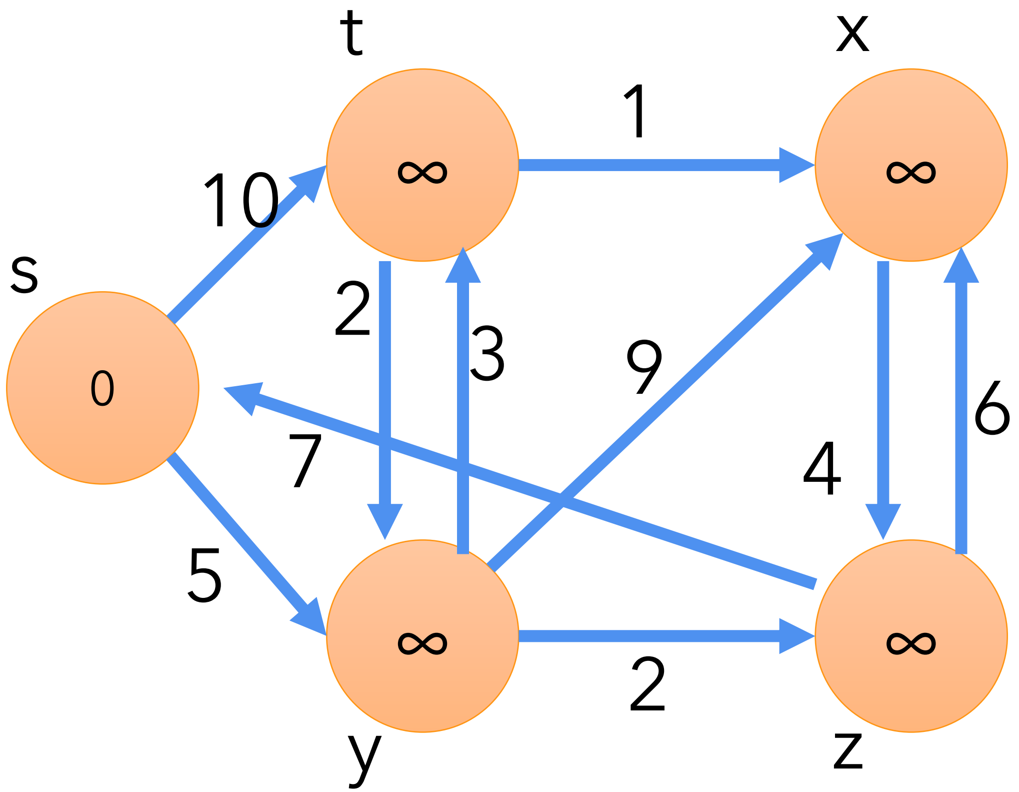
Single source shortest paths – Dijkstra

```
DIJKSTRA(G,W,s)
Sv = []
Q = new MinPriorityQueue(V)
WHILE !Q.isEmpty()
        u = Q.EXTRACT_MIN()
        Sv = Sv U u
        FOR each v adjunct to u
            RELAX(u,v,W)
```

O(V+E log V)

Grafos

Single source shortest paths – Dijkstra



s	t	y	x	z
0	∞	∞	∞	∞

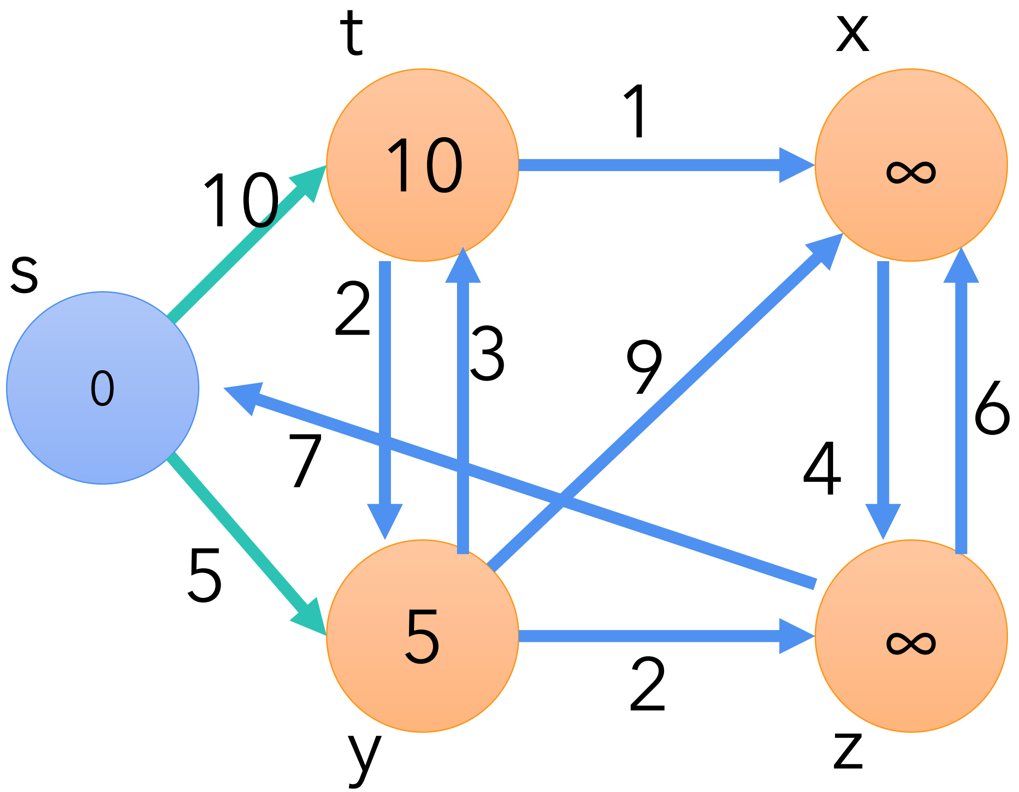
	s	t	y	x	z
d[ ]	0	∞	∞	∞	∞
π[ ]	NULL	NULL	NULL	NULL	NULL

Grafos

Single source shortest paths – Dijkstra

Mientras la cola

no este vacía



y	t	x	z
5	10	∞	∞

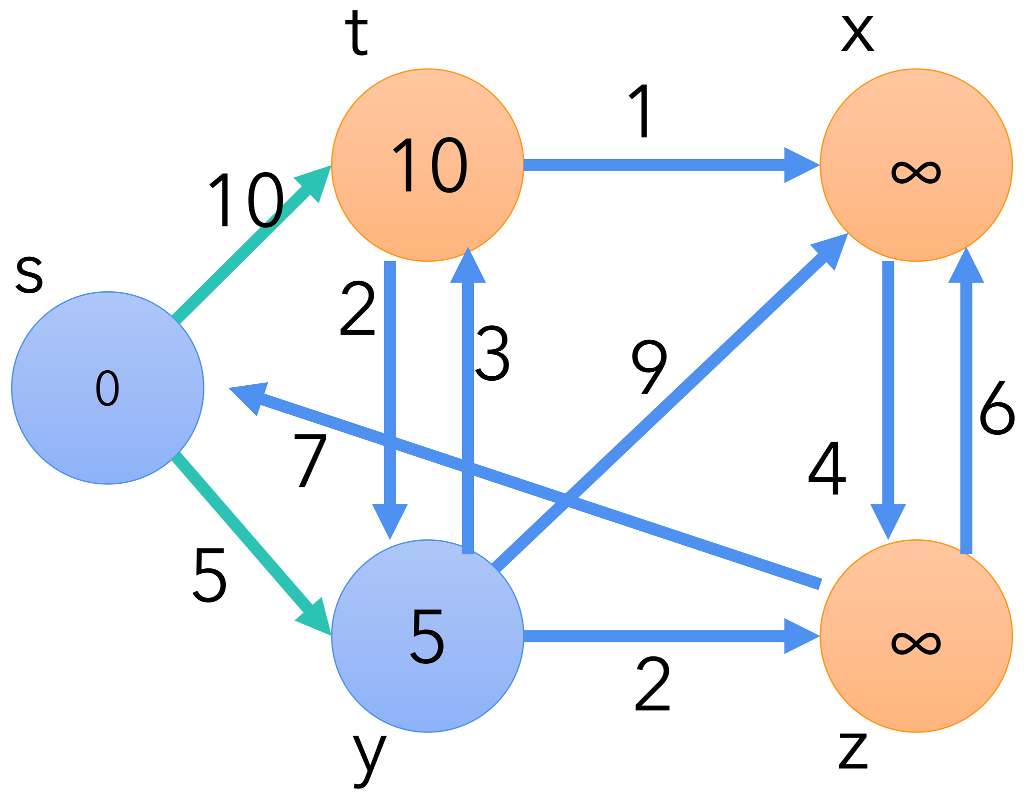
	s	t	y	x	z
d[ ]	0	10	5	∞	∞
π[ ]	NULL	s	s	NULL	NULL

Grafos

Single source shortest paths – Dijkstra

Mientras la cola

no este vacía



y	t	x	z
5	10	∞	∞

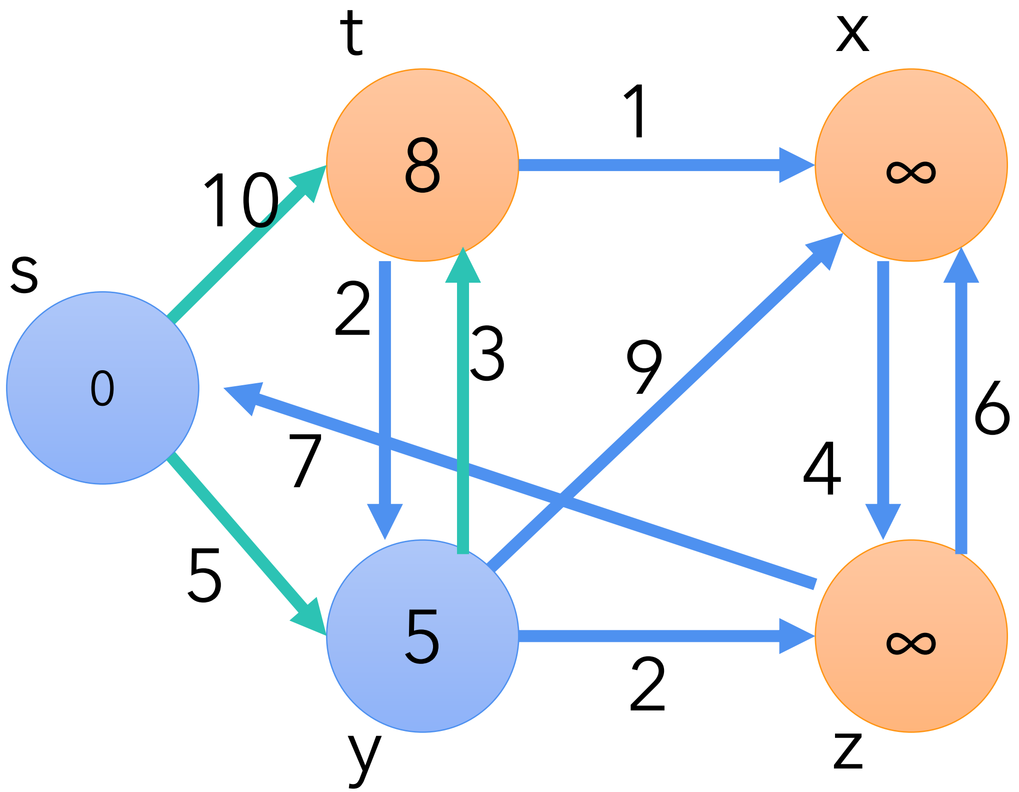
	s	t	y	x	z
d[ ]	0	10	5	∞	∞
π[ ]	NULL	s	s	NULL	NULL

Grafos

Single source shortest paths – Dijkstra

Mientras la cola

no este vacía



t	x	z
8	∞	∞

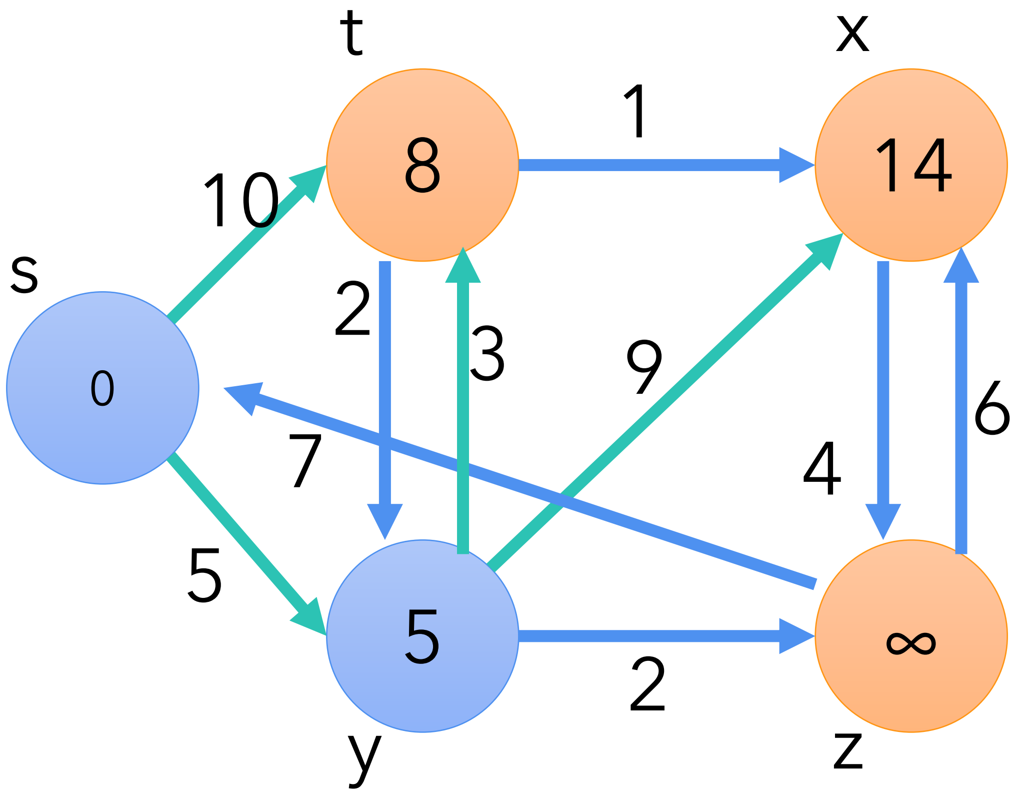
	s	t	y	x	z
d[ ]	0	8	5	∞	∞
π[ ]	NULL	y	s	NULL	NULL

Grafos

Single source shortest paths – Dijkstra

Mientras la cola

no este vacía



t	x	z
8	14	∞

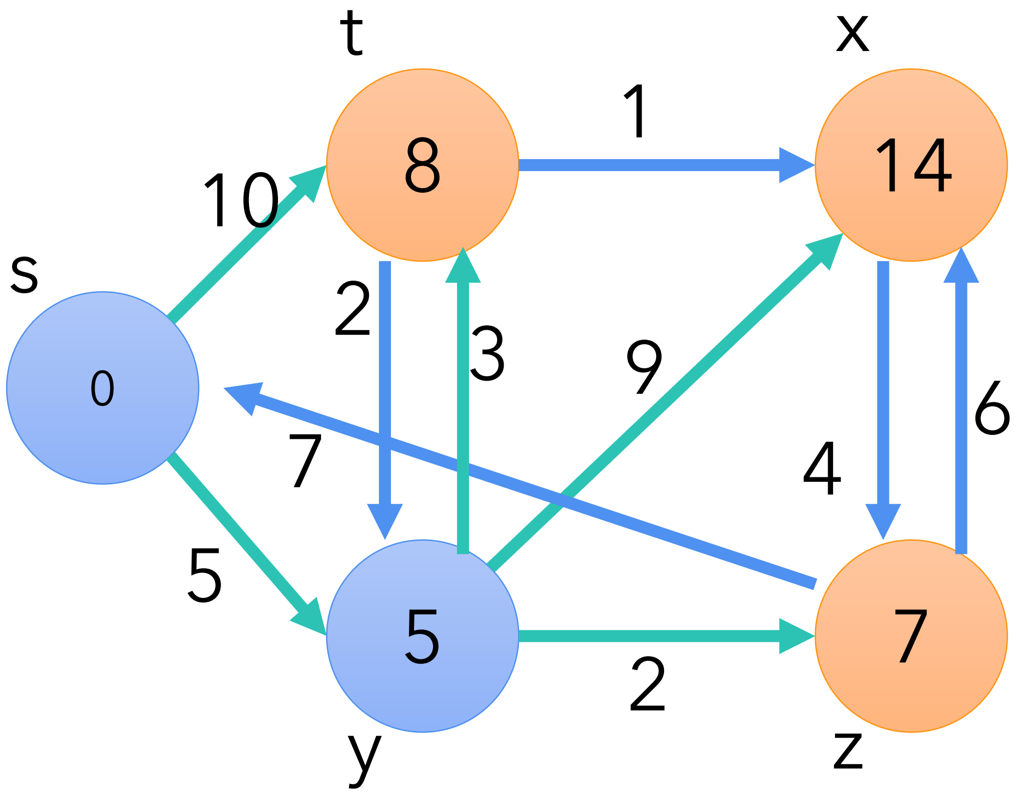
	s	t	y	x	z
d[ ]	0	8	5	14	∞
π[ ]	NULL	y	s	y	NULL

Grafos

Single source shortest paths – Dijkstra

Mientras la cola

no este vacía



z	t	x
7	8	14

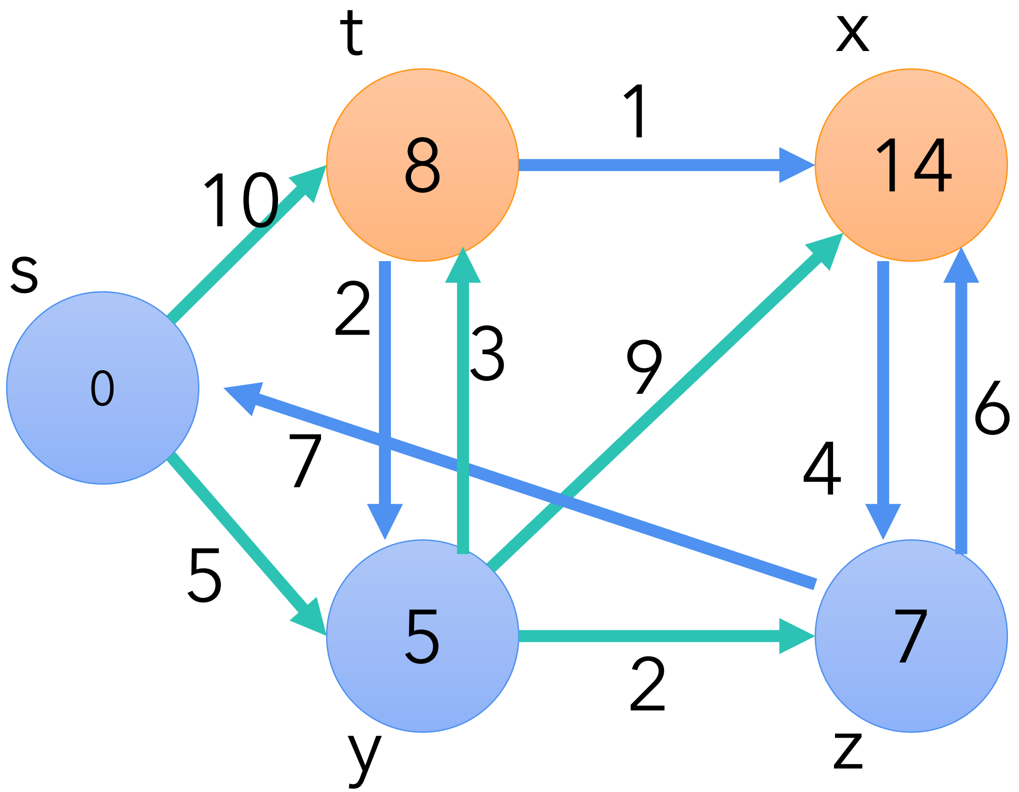
	s	t	y	x	z
d[ ]	0	8	5	14	7
π[ ]	NULL	y	s	y	y

Grafos

Single source shortest paths – Dijkstra

Mientras la cola

no este vacía



z	t	x
7	8	14

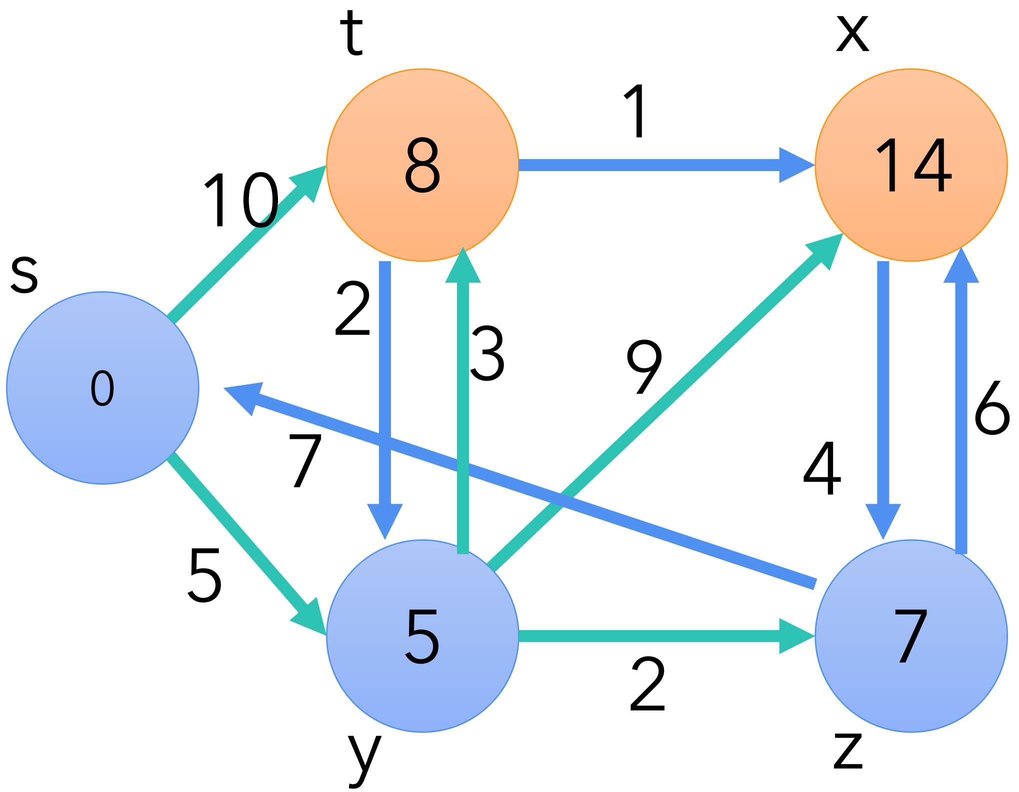
	s	t	y	x	z
d[ ]	0	8	5	14	7
π[ ]	NULL	y	s	y	y

Grafos

Single source shortest paths – Dijkstra

Mientras la cola

no este vacía



t	x
8	14

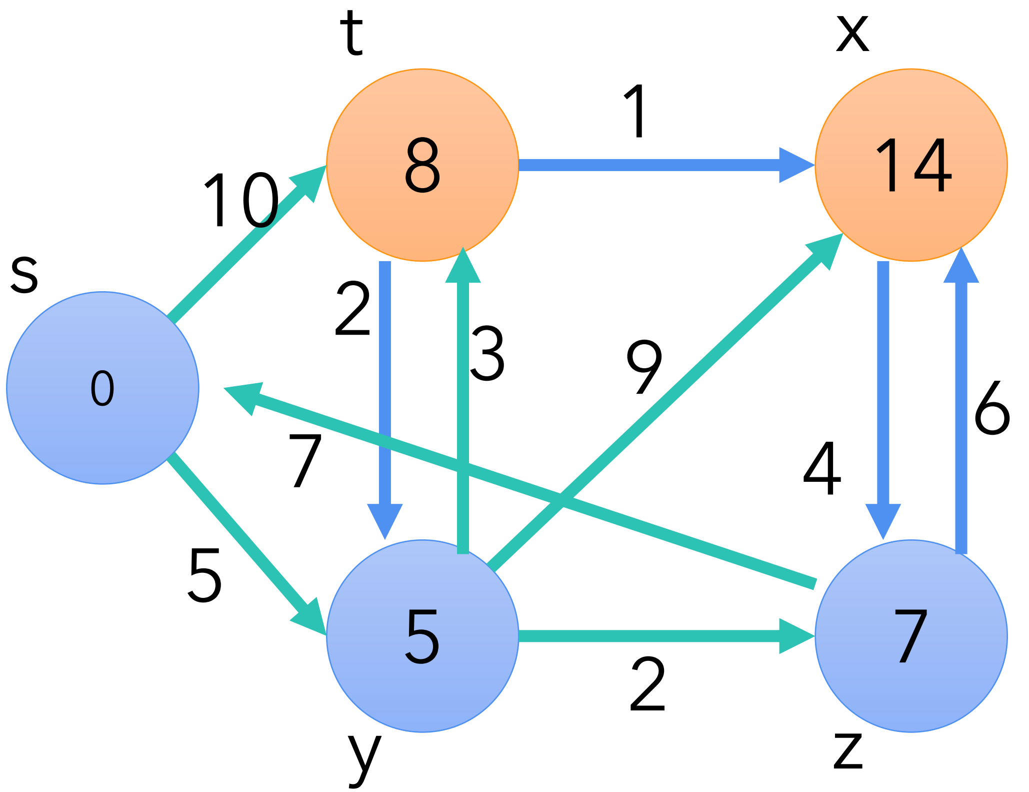
	s	t	y	x	z
d[ ]	0	8	5	14	7
π[ ]	NULL	y	s	y	y

Grafos

Single source shortest paths – Dijkstra

Mientras la cola

no este vacía



t	x
8	14

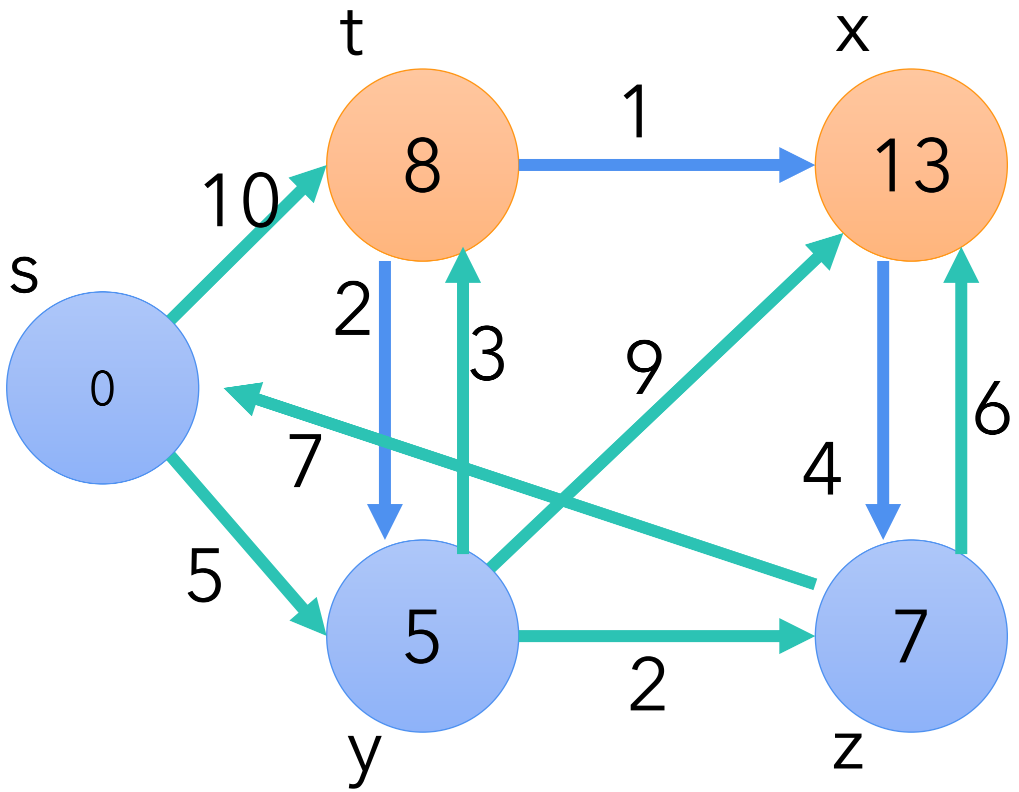
	s	t	y	x	z
d[ ]	0	8	5	14	7
π[ ]	NULL	y	s	y	y

Grafos

Single source shortest paths – Dijkstra

Mientras la cola

no este vacía



t	x
8	13

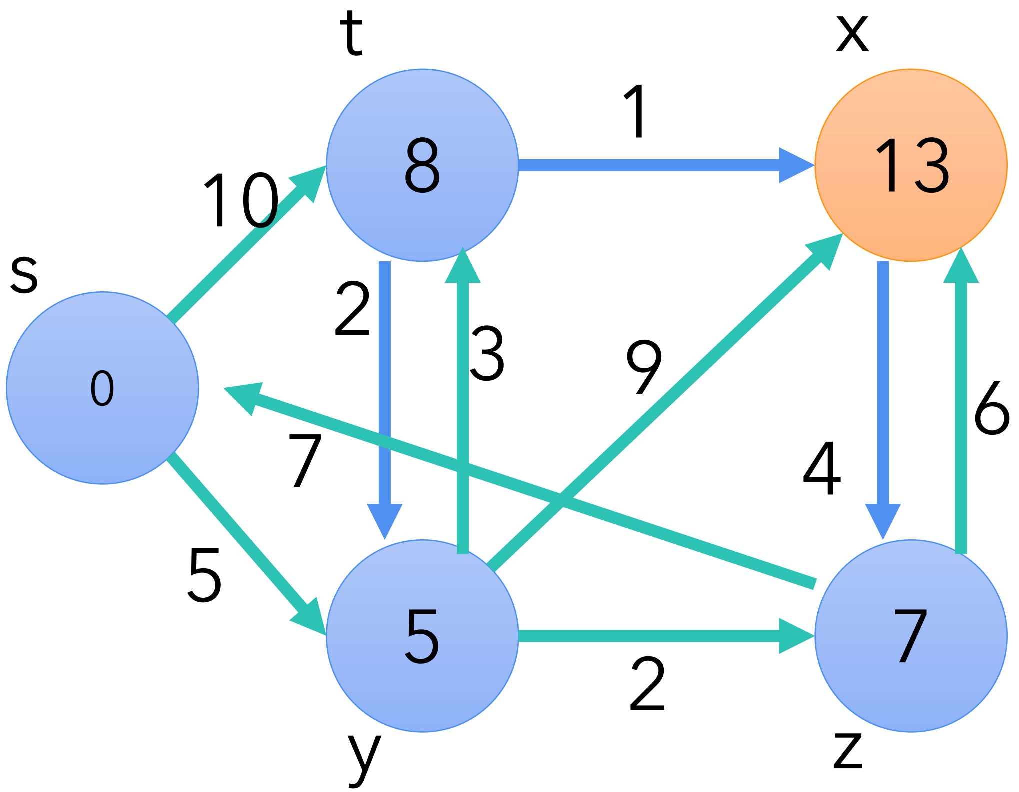
	s	t	y	x	z
d[ ]	0	8	5	13	7
π[ ]	NULL	y	s	z	y

Grafos

Single source shortest paths – Dijkstra

Mientras la cola

no este vacía



t	x
8	13

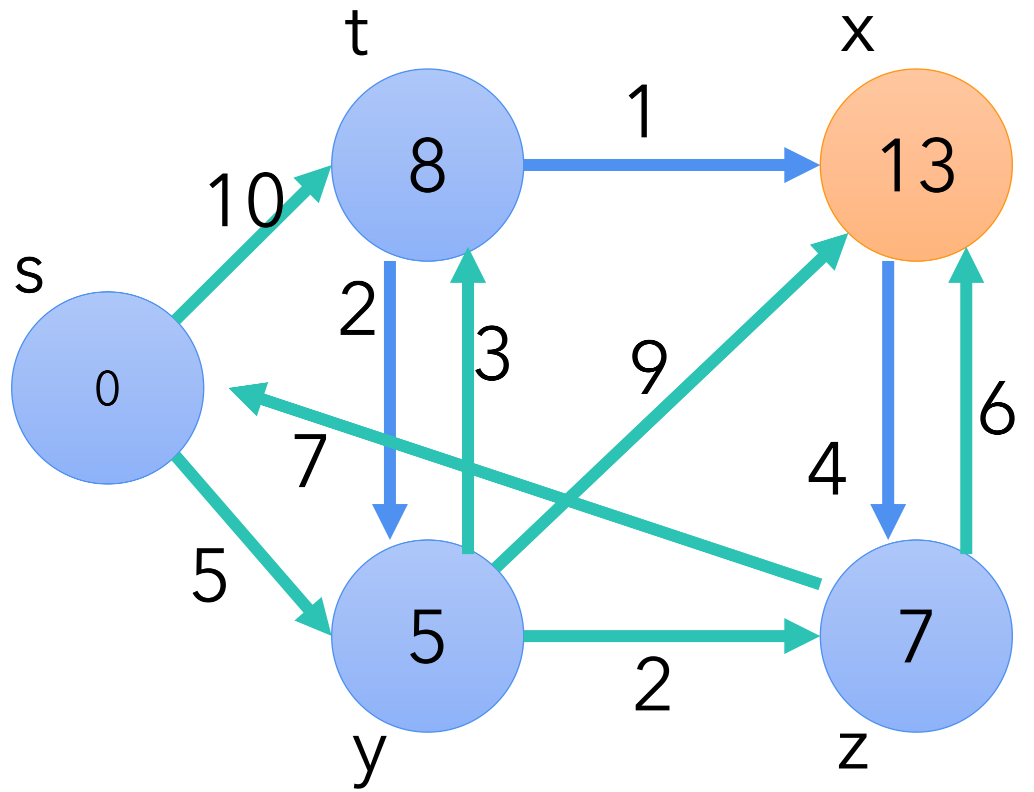
	s	t	y	x	z
d[ ]	0	8	5	13	7
π[ ]	NULL	y	s	z	y

Grafos

Single source shortest paths – Dijkstra

Mientras la cola

no este vacía



x
13

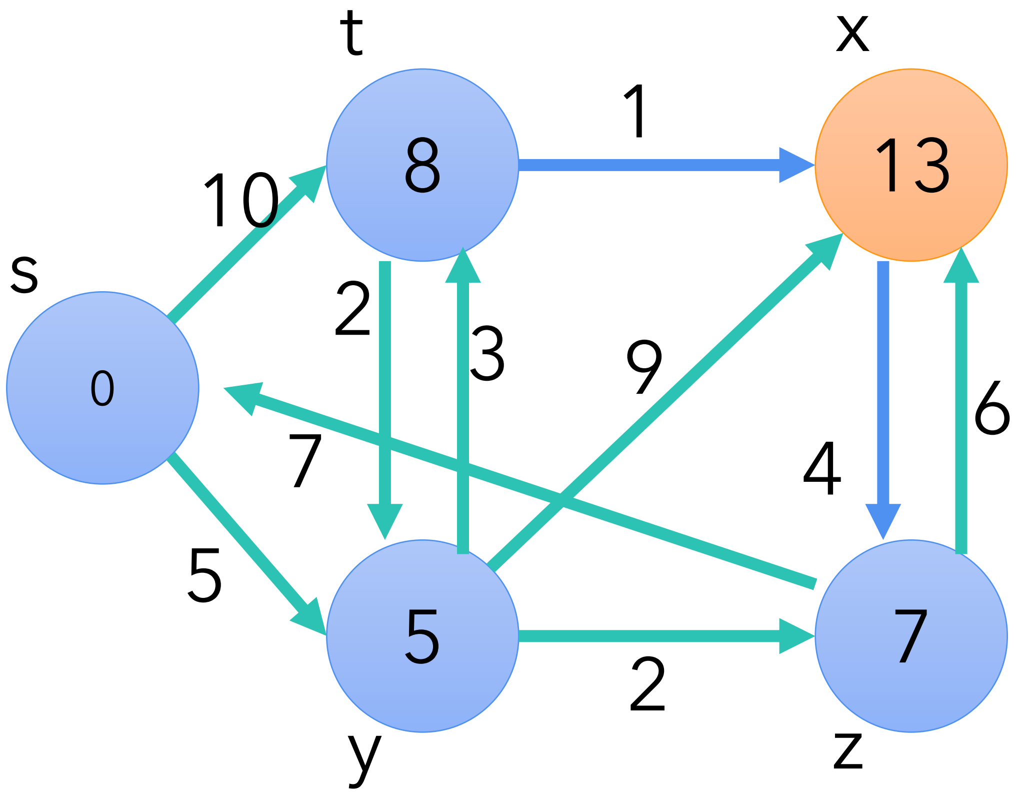
	s	t	y	x	z
d[ ]	0	8	5	13	7
π[ ]	NULL	y	s	z	y

Grafos

Single source shortest paths – Dijkstra

Mientras la cola

no este vacía



x
13

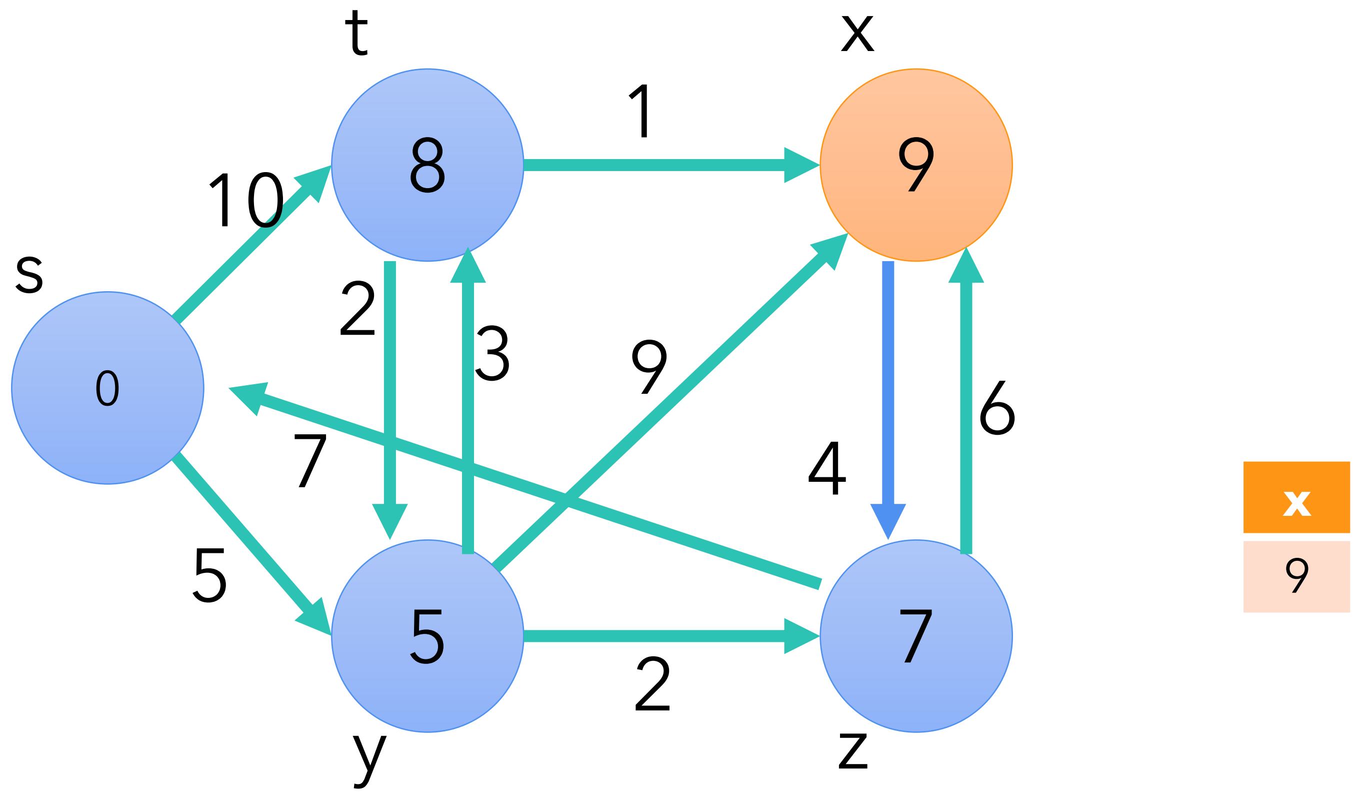
	s	t	y	x	z
d[ ]	0	8	5	13	7
π[ ]	NULL	y	s	z	y

Grafos

Single source shortest paths – Dijkstra

Mientras la cola

no este vacía



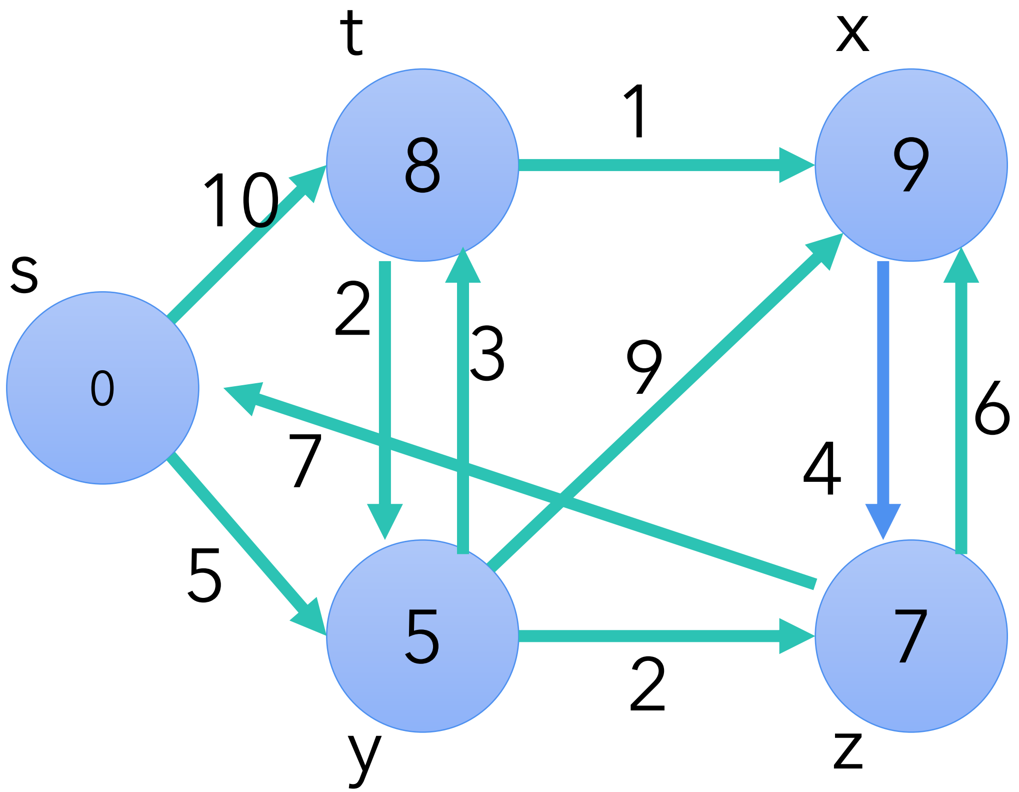
	s	t	y	x	z
d[ ]	0	8	5	9	7
π[ ]	NULL	y	s	t	y

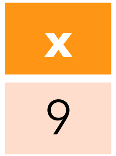
Grafos

Single source shortest paths – Dijkstra

Mientras la cola

no este vacía





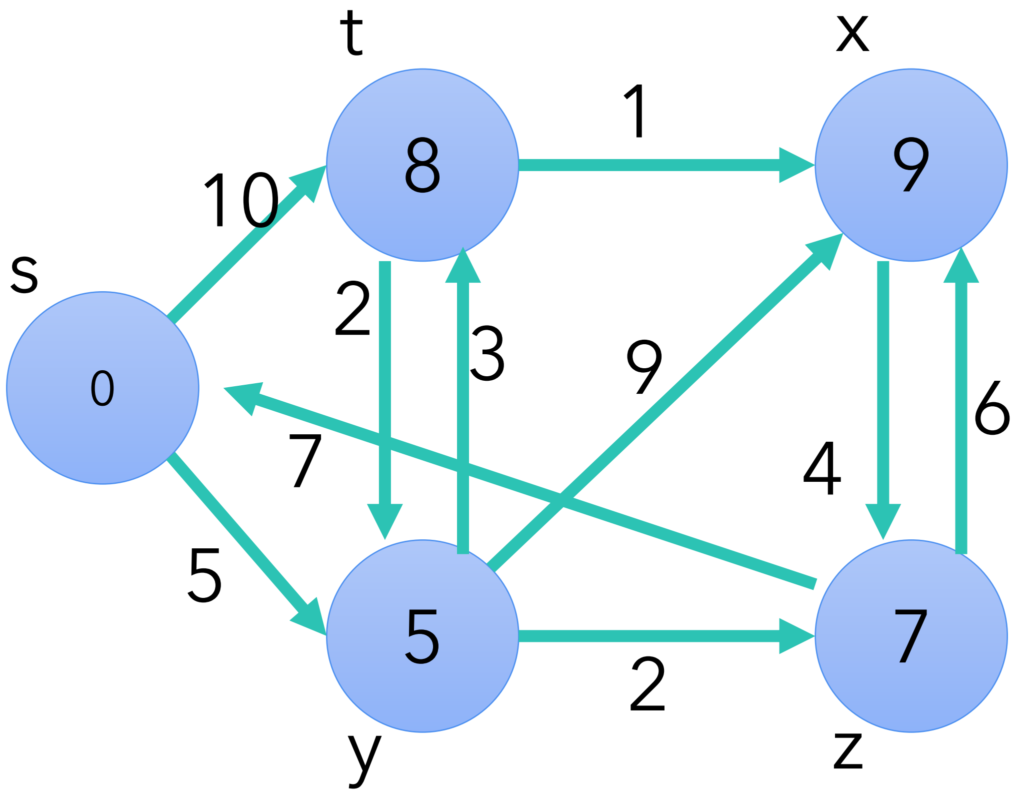
	s	t	y	x	z
d[ ]	0	8	5	9	7
π[ ]	NULL	y	s	t	y

Grafos

Single source shortest paths – Dijkstra

Mientras la cola

no este vacía



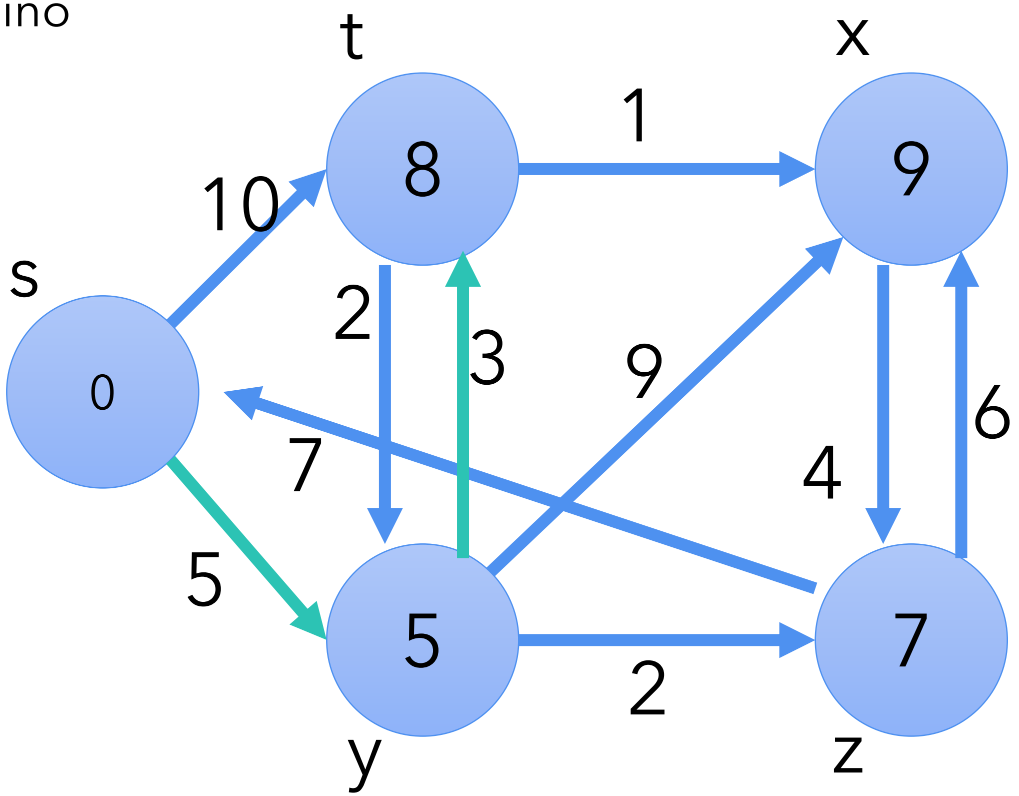
	s	t	y	x	z
d[ ]	0	8	5	9	7
π[ ]	NULL	y	s	t	y

Grafos

Single source shortest paths – Dijkstra

Reconstrucción del camino mas corto

(s, t)



	s	t	y	x	z
d[ ]	0	8	5	9	7
π[ ]	NULL	y	s	t	y

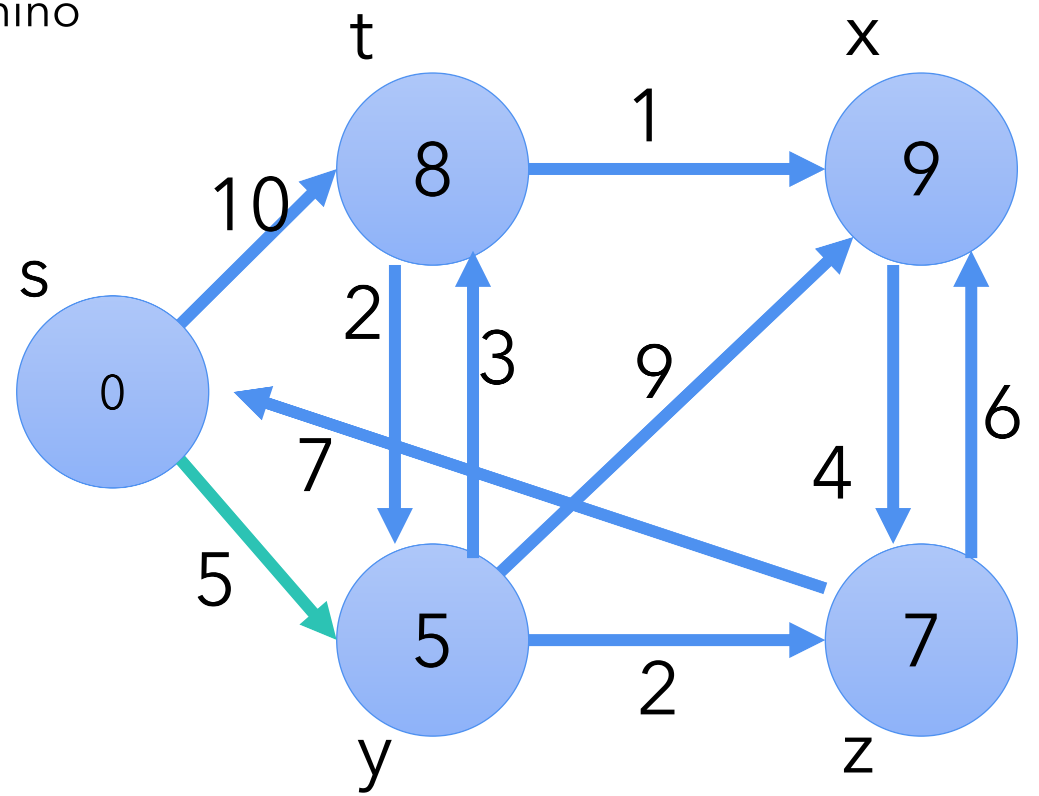
Grafos

Single source shortest paths – Dijkstra

Reconstrucción del camino

mas corto

(s, y)



	s	t	y	x	z
d[ ]	0	8	5	9	7
π[ ]	NULL	y	s	t	y

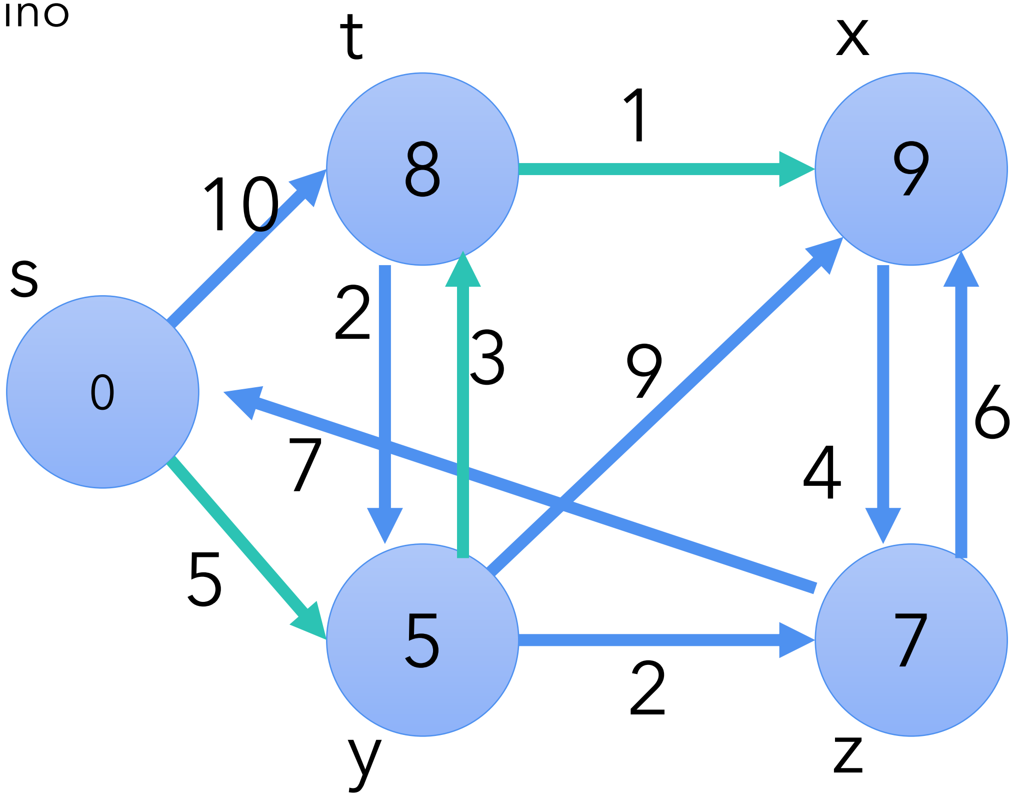
Grafos

Single source shortest paths – Dijkstra

Reconstrucción del camino

mas corto

(s, x)



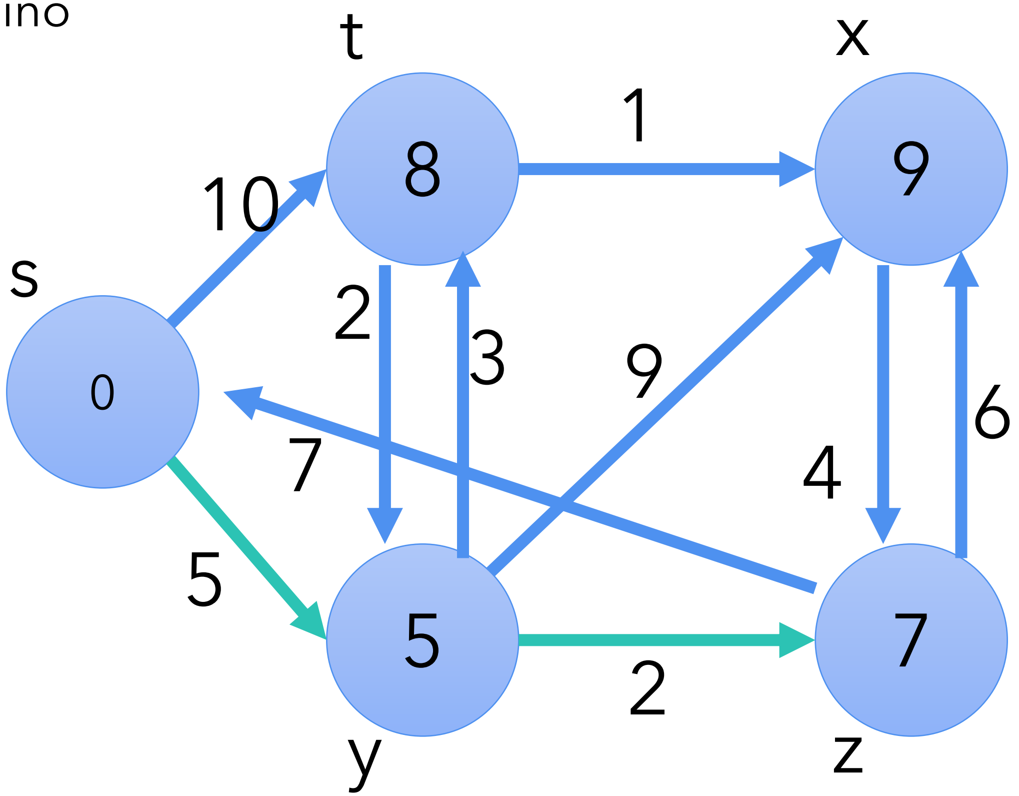
	s	t	y	x	z
d[ ]	0	8	5	9	7
π[ ]	NULL	y	s	t	y

Grafos
Single source shortest paths – Dijkstra

Reconstrucción del camino

mas corto

(s, z)

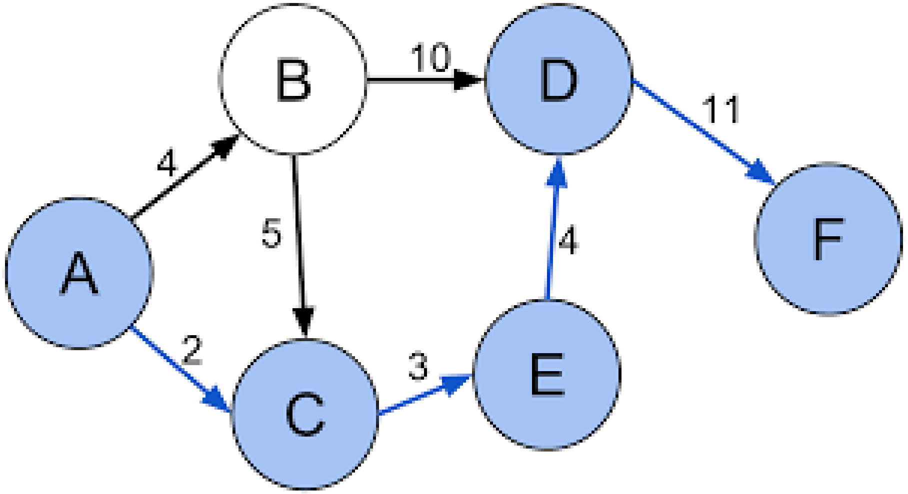


	s	t	y	x	z
d[ ]	0	8	5	9	7
π[ ]	NULL	y	s	t	y

Grafos

Descripción del problema

- El camino más corto desde un vértice s a todos los demás vértices del grado:
  - Bellman-Ford O(VE)
  - Dijkstra O(V + E log V)
- El camino más corto entre todos los pares del vértice del grafo
  - Floyd-Warshall
  - Johnson



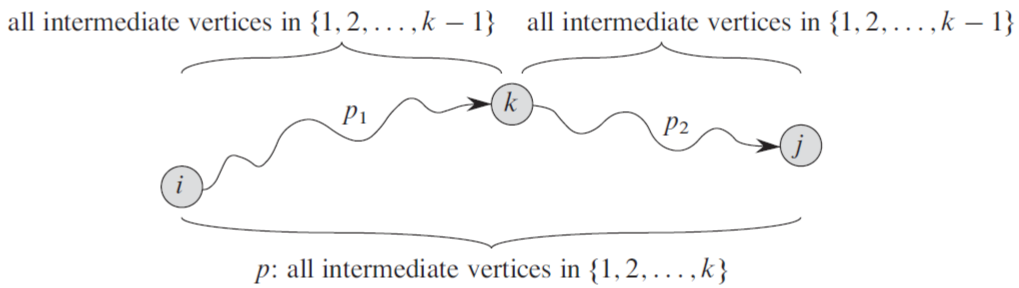
Grafos

All pairs shortest paths– Floyd-Warshall

- Considera todos los vértices intermedios para determinar el camino más corto, donde un vértice intermedio en un camino:

$$p = \langle v_1, v_2, v_3 \dots . v_l \rangle$$

  son todos los vértices diferentes a v_1 y v_l



- Si p_1 es el camino mas corto entre i y k, y p_2 es el camino mas corto entre k y j, entonces, el camino mas corto entre i y j es $p_1 \cup p_2$

Grafos

All pairs shortest paths– Floyd-Warshall

- Vamos a usar una representacion matricial para el camino mas corto
- La matriz D mantendrá las estimaciones de los caminos mas cortos en cada iteración y la matriz π los respectivos predecesores
- Inicialmente, la matriz D será igual a la matriz de pesos
- Para la matriz π:

$$\pi_{ij}^{(0)} = \begin{cases} \text{NIL} & \text{if } i = j \text{ or } w_{ij} = \infty , \\ i & \text{if } i \neq j \text{ and } w_{ij} < \infty . \end{cases}$$

- En cada iteración del algoritmo, se compara los caminos intermedios de i hasta j, y se actualiza D y π:

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0 , \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1 . \end{cases}$$

Grafos

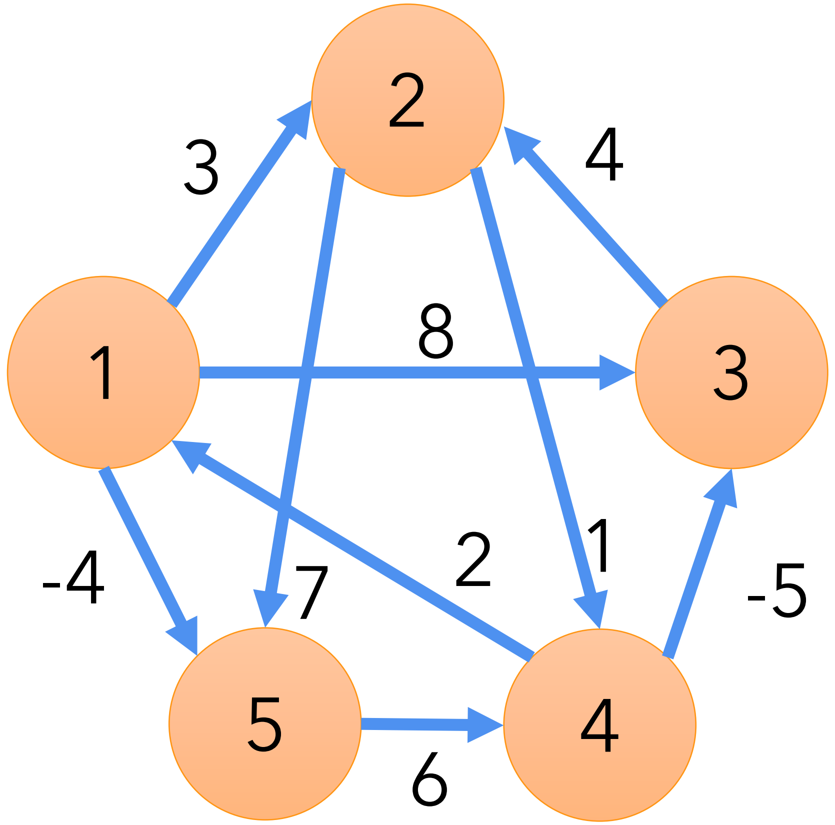
All pairs shortest paths- Floyd-Warshall

```
FLOYD-WARSHALL(W)
n = W.rows
D = W
FOR k=1 to n
    FOR i = 1 to n
        FOR j = 1 to n
            D(i,j) = min(D(i,j), D(i,k)+D(k,j))
```

$O(V^3)$

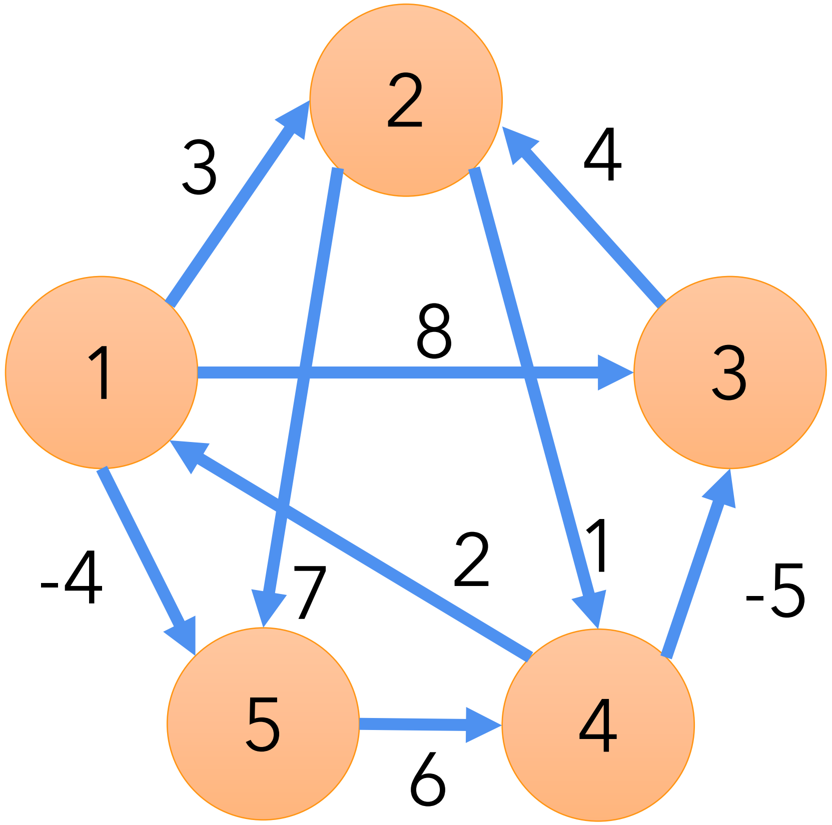
Grafos

All pairs shortest paths– Floyd-Warshall



Grafos

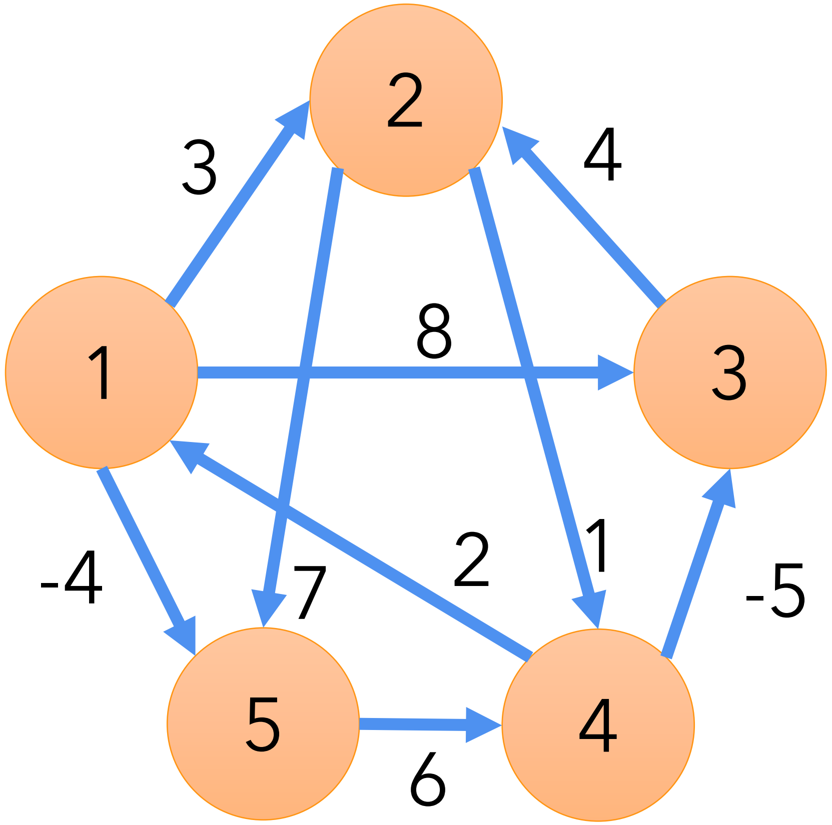
All pairs shortest paths– Floyd-Warshall



D	1	2	3	4	5
1	0	3	8		-4
2		0		1	7
3		4	0		
4	2		-5	0	
5				6	0

Grafos

All pairs shortest paths– Floyd-Warshall



D	1	2	3	4	5
1	0	3	8	∞	-4
2	∞	0	∞	1	7
3	∞	4	0	∞	∞
4	2	∞	-5	0	∞
5	∞	∞	∞	6	0

π	1	2	3	4	5
1	N	1	1	N	1
2	N	N	N	2	2
3	N	3	N	N	N
4	4	N	4	N	N
5	N	N	N	5	N

Grafos

All pairs shortest paths- Floyd-Warshall

k = 1, i = 1, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d11 = min(d11, d11+d11)

d12 = min(d12, d11+d12)=min(3,0+3)

d13 = min(d13,d11+d13)=min(8,0+8)

d14 = min(d14,d11+d14)=min(∞, 0+∞)

d15 = min(d15,d11+d15)=min(-4,0-4)

D	1	2	3	4	5
1	0	3	8	∞	-4
2	∞	0	∞	1	7
3	∞	4	0	∞	∞
4	2	∞	-5	0	∞
5	∞	∞	∞	6	0

π	1	2	3	4	5
1	N	1	1	N	1
2	N	N	N	2	2
3	N	3	N	N	N
4	4	N	4	N	N
5	N	N	N	5	N

Grafos

All pairs shortest paths– Floyd-Warshall

k = 1, i = 2, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d21 = min(d21, d21+d11)=min(∞, ∞+0)

d22 = min(d22, d21+d12)=min(0, ∞+3)

d23 = min(d23,d21+d13)=min(∞, ∞+8)

d24 = min(d24,d21+d14)=min(1, ∞+7)

d25 = min(d25,d21+d15)=min(7, ∞-4)

D	1	2	3	4	5
1	0	3	8	∞	-4
2	∞	0	∞	1	7
3	∞	4	0	∞	∞
4	2	∞	-5	0	∞
5	∞	∞	∞	6	0

π	1	2	3	4	5
1	N	1	1	N	1
2	N	N	N	2	2
3	N	3	N	N	N
4	4	N	4	N	N
5	N	N	N	5	N

Grafos

All pairs shortest paths– Floyd-Warshall

k = 1, i = 3, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d31 = min(d31, d31+d11)=min(∞, ∞+0)

d32 = min(d32, d31+d12)=min(4, ∞+3)

d33 = min(d33,d31+d13)=min(0, ∞+8)

d34 = min(d34,d31+d14)=min(∞, ∞+7)

d35 = min(d35,d31+d15)=min(∞, ∞-4)

D	1	2	3	4	5
1	0	3	8	∞	-4
2	∞	0	∞	1	7
3	∞	4	0	∞	∞
4	2	∞	-5	0	∞
5	∞	∞	∞	6	0

π	1	2	3	4	5
1	N	1	1	N	1
2	N	N	N	2	2
3	N	3	N	N	N
4	4	N	4	N	N
5	N	N	N	5	N

Grafos

All pairs shortest paths– Floyd-Warshall

k = 1, i = 4, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d41 = min(d41, d41+d11)=min(2, 2+0)

d42 = min(d42, d41+d12)=min(∞, 2+3)=5

d43 = min(d43,d41+d13)=min(-5, 2+8)

d44 = min(d44,d41+d14)=min(0, 2+7)

d45 = min(d45,d41+d15)=min(∞, 2-4)=-2

D	1	2	3	4	5
1	0	3	8	∞	-4
2	∞	0	∞	1	7
3	∞	4	0	∞	∞
4	2	∞	-5	0	∞
5	∞	∞	∞	6	0

π	1	2	3	4	5
1	N	1	1	N	1
2	N	N	N	2	2
3	N	3	N	N	N
4	4	N	4	N	N
5	N	N	N	5	N

Grafos

All pairs shortest paths– Floyd-Warshall

k = 1, i = 4, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

D	1	2	3	4	5
1	0	3	8	∞	-4
2	∞	0	∞	1	7
3	∞	4	0	∞	∞
4	2	**5**	-5	0	**-2**
5	∞	∞	∞	6	0

d42 = min(d42, d41+d12)=min(∞, 2+3)=5

d45 = min(d45,d41+d15)=min(∞, 2-4)=-2

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,, \\ \pi_{kj}^{(k-1)} & \text{if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,. \end{cases}$$

π	1	2	3	4	5
1	N	1	1	N	1
2	N	N	N	2	2
3	N	3	N	N	N
4	4	**1**	4	N	**1**
5	N	N	N	5	N

Grafos

All pairs shortest paths- Floyd-Warshall

k = 1, i = 5, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d51 = min(d51, d51+d11)=min(∞, ∞ +0)

d52 = min(d52, d51+d12)=min(∞, ∞ +3)

d53 = min(d53,d51+d13)=min(∞, ∞ +8)

d54 = min(d54,d51+d14)=min(6, ∞ +7)

d55 = min(d55,d51+d15)=min(0, ∞ -4)

D	1	2	3	4	5
1	0	3	8	∞	-4
2	∞	0	∞	1	7
3	∞	4	0	∞	∞
4	2	5	-5	0	-2
5	∞	∞	∞	6	0

π	1	2	3	4	5
1	N	1	1	N	1
2	N	N	N	2	2
3	N	3	N	N	N
4	4	1	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths- Floyd-Warshall

k = 2, i = 1, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d11 = min(d11, d12+d21)=min(0, 3 + ∞)

d12 = min(d12, d12+d22)=min(3, 3 +0)

d13 = min(d13,d12+d23)=min(8, 3 + ∞)

d14 = min(d14,d12+d24)=min(∞, 3 +1)=4

d15 = min(d15,d12+d25)=min(-4, 3 +7)

D	1	2	3	4	5
1	0	3	8	∞	-4
2	∞	0	∞	1	7
3	∞	4	0	∞	∞
4	2	5	-5	0	-2
5	∞	∞	∞	6	0

π	1	2	3	4	5
1	N	1	1	N	1
2	N	N	N	2	2
3	N	3	N	N	N
4	4	1	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths- Floyd-Warshall

k = 2, i = 1, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0 , \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1 . \end{cases}$$

D	1	2	3	4	5
1	0	3	8	**4**	-4
2	∞	0	∞	1	7
3	∞	4	0	∞	∞
4	2	5	-5	0	-2
5	∞	∞	∞	6	0

d14 = min(d14,d12+d24)=min(∞, 3 +1)=4

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)} , \\ \pi_{kj}^{(k-1)} & \text{if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)} . \end{cases}$$

π	1	2	3	4	5
1	N	1	1	**2**	1
2	N	N	N	2	2
3	N	3	N	N	N
4	4	1	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths- Floyd-Warshall

k = 2, i = 2, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d21 = min(d21, d22+d21)=min(∞, 0+ ∞)

d22 = min(d22, d22+d22)=min(0, 0 +0)

d23 = min(d23,d22+d23)=min(∞, 0 + ∞)

d24 = min(d24,d22+d24)=min(1, 0 +1)

d25 = min(d25,d22+d25)=min(7, 0 +7)

D	**1**	**2**	**3**	**4**	**5**
1	0	3	8	4	-4
2	∞	0	∞	1	7
3	∞	4	0	∞	∞
4	2	5	-5	0	-2
5	∞	∞	∞	6	0

π	**1**	**2**	**3**	**4**	**5**
1	N	1	1	2	1
2	N	N	N	2	2
3	N	3	N	N	N
4	4	1	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths- Floyd-Warshall

k = 2, i = 3, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0 , \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1 . \end{cases}$$

d31 = min(d31, d32+d21)=min(∞, 4+ ∞)

d32 = min(d32, d32+d22)=min(4, 4 +0)

d33 = min(d33,d32+d23)=min(0, 4 + ∞)

d34 = min(d34,d32+d24)=min(∞, 4 +1)=5

d35 = min(d35,d32+d25)=min(∞, 4 +7)=11

D	**1**	**2**	**3**	**4**	**5**
1	0	3	8	4	-4
2	∞	0	∞	1	7
3	∞	4	0	∞	∞
4	2	5	-5	0	-2
5	∞	∞	∞	6	0

π	**1**	**2**	**3**	**4**	**5**
1	N	1	1	2	1
2	N	N	N	2	2
3	N	3	N	N	N
4	4	1	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths- Floyd-Warshall

k = 2, i = 3, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d34 = min(d34,d32+d24)=min(∞, 4 +1)=5

d35 = min(d35,d32+d25)=min(∞, 4 +7)=11

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,, \\ \pi_{kj}^{(k-1)} & \text{if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,. \end{cases}$$

D	**1**	**2**	**3**	**4**	**5**
1	0	3	8	4	-4
2	∞	0	∞	1	7
3	∞	4	0	**5**	**11**
4	2	5	-5	0	-2
5	∞	∞	∞	6	0

π	**1**	**2**	**3**	**4**	**5**
1	N	1	1	2	1
2	N	N	N	2	2
3	N	3	N	**2**	**2**
4	4	1	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths– Floyd-Warshall

k = 2, i = 4, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d41 = min(d41, d42+d21)=min(2, 5+ ∞)

d42 = min(d42, d42+d22)=min(5, 5 +0)

d43 = min(d43,d42+d23)=min(-5, 5 + ∞)

d44 = min(d44,d42+d24)=min(0, 5 +1)

d45 = min(d45,d42+d25)=min(-2, 5 +7)

D	**1**	**2**	**3**	**4**	**5**
1	0	3	8	4	-4
2	∞	0	∞	1	7
3	∞	4	0	5	11
4	2	5	-5	0	-2
5	∞	∞	∞	6	0

π	**1**	**2**	**3**	**4**	**5**
1	N	1	1	2	1
2	N	N	N	2	2
3	N	3	N	2	2
4	4	1	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths- Floyd-Warshall

k = 2, i = 5, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d51 = min(d51, d52+d21)=min(∞, ∞ + ∞)

d52 = min(d52, d52+d22)=min(∞, ∞ +0)

d53 = min(d53,d52+d23)=min(∞, ∞ + ∞)

d54 = min(d54,d52+d24)=min(6, ∞ +1)

d55 = min(d55,d52+d25)=min(0, ∞ +7)

D	1	2	3	4	5
1	0	3	8	4	-4
2	∞	0	∞	1	7
3	∞	4	0	5	11
4	2	5	-5	0	-2
5	∞	∞	∞	6	0

π	1	2	3	4	5
1	N	1	1	2	1
2	N	N	N	2	2
3	N	3	N	2	2
4	4	1	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths- Floyd-Warshall

k = 3, i = 1, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d11 = min(d11, d13+d31)=min(0, 8 + ∞)

d12 = min(d12, d13+d32)=min(3, 8 +4)

d13 = min(d13,d13+d33)=min(8, 8 + 0)

d14 = min(d14,d13+d34)=min(4, 8 +5)

d15 = min(d15,d13+d35)=min(-4, 8 +11)

D	1	2	3	4	5
1	0	3	8	4	-4
2	∞	0	∞	1	7
3	∞	4	0	5	11
4	2	5	-5	0	-2
5	∞	∞	∞	6	0

π	1	2	3	4	5
1	N	1	1	2	1
2	N	N	N	2	2
3	N	3	N	2	2
4	4	1	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths- Floyd-Warshall

k = 3, i = 2, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d21 = min(d21, d23+d31)=min(∞, ∞ + ∞)

d22 = min(d22, d23+d32)=min(0, ∞ +4)

d23 = min(d23,d23+d33)=min(∞, ∞ + 0)

d24 = min(d24,d23+d34)=min(1, ∞ +5)

d25 = min(d25,d23+d35)=min(7, ∞ +11)

D	1	2	3	4	5
1	0	3	8	4	-4
2	∞	0	∞	1	7
3	∞	4	0	5	11
4	2	5	-5	0	-2
5	∞	∞	∞	6	0

π	1	2	3	4	5
1	N	1	1	2	1
2	N	N	N	2	2
3	N	3	N	2	2
4	4	1	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths– Floyd-Warshall

k = 3, i = 3, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0 \,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1 \,. \end{cases}$$

d31 = min(d31, d33+d31)=min(∞, 0 + ∞)

d32 = min(d32, d33+d32)=min(4, 0 +4)

d33 = min(d33,d33+d33)=min(0, 0 + 0)

d34 = min(d34,d33+d34)=min(5, 0 +5)

d35 = min(d35,d33+d35)=min(11, 0 +11)

D	1	2	3	4	5
1	0	3	8	4	-4
2	∞	0	∞	1	7
3	∞	4	0	5	11
4	2	5	-5	0	-2
5	∞	∞	∞	6	0

π	1	2	3	4	5
1	N	1	1	2	1
2	N	N	N	2	2
3	N	3	N	2	2
4	4	1	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths– Floyd-Warshall

k = 3, i = 4, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d41 = min(d41, d43+d31)=min(2, -5 + ∞)

d42 = min(d42, d43+d32)=min(5, -5 +4)=-1

d43 = min(d43,d43+d33)=min(-5, -5 + 0)

d44 = min(d44,d43+d34)=min(0, -5 +5)

d45 = min(d45,d43+d35)=min(-2, -5 +11)

D	1	2	3	4	5
1	0	3	8	4	-4
2	∞	0	∞	1	7
3	∞	4	0	5	11
4	2	5	-5	0	-2
5	∞	∞	∞	6	0

π	1	2	3	4	5
1	N	1	1	2	1
2	N	N	N	2	2
3	N	3	N	2	2
4	4	1	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths– Floyd-Warshall

k = 3, i = 4, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0 , \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1 . \end{cases}$$

d42 = min(d42, d43+d32)=min(5, -5 +4)=-1

D	**1**	**2**	**3**	**4**	**5**
1	0	3	8	4	-4
2	∞	0	∞	1	7
3	∞	4	0	5	11
4	2	**-1**	-5	0	-2
5	∞	∞	∞	6	0

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)} , \\ \pi_{kj}^{(k-1)} & \text{if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)} . \end{cases}$$

π	**1**	**2**	**3**	**4**	**5**
1	N	1	1	2	1
2	N	N	N	2	2
3	N	3	N	2	2
4	4	**3**	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths- Floyd-Warshall

k = 3, i = 5, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0 , \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1 . \end{cases}$$

d51 = min(d51, d53+d31)=min(∞, ∞ + ∞)

d52 = min(d52, d53+d32)=min(∞, ∞ +4)

d53 = min(d53,d53+d33)=min(∞, ∞ + 0)

d54 = min(d54,d53+d34)=min(6, ∞ +5)

d55 = min(d55,d53+d35)=min(0, ∞ +11)

D	**1**	**2**	**3**	**4**	**5**
1	0	3	8	4	-4
2	∞	0	∞	1	7
3	∞	4	0	5	11
4	2	-1	-5	0	-2
5	∞	∞	∞	6	0

π	**1**	**2**	**3**	**4**	**5**
1	N	1	1	2	1
2	N	N	N	2	2
3	N	3	N	2	2
4	4	3	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths– Floyd-Warshall

k = 4, i = 1, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0 \, , \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1 \, . \end{cases}$$

d11 = min(d11, d14+d41)=min(0, 4 + 2)

d12 = min(d12, d14+d42)=min(3, 4 -1)

d13 = min(d13,d14+d43)=min(8, 4 -5)=-1

d14 = min(d14,d14+d44)=min(4, 4 +0)

d15 = min(d15,d14+d45)=min(-4, 4 -2)

D	**1**	**2**	**3**	**4**	**5**
1	0	3	8	4	-4
2	∞	0	∞	1	7
3	∞	4	0	5	11
4	2	-1	-5	0	-2
5	∞	∞	∞	6	0

π	**1**	**2**	**3**	**4**	**5**
1	N	1	1	2	1
2	N	N	N	2	2
3	N	3	N	2	2
4	4	3	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths- Floyd-Warshall

k = 4, i = 1, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d13 = min(d13,d14+d43)=min(8, 4 -5)=-1

D	1	2	3	4	5
1	0	3	**-1**	4	-4
2	∞	0	∞	1	7
3	∞	4	0	5	11
4	2	-1	-5	0	-2
5	∞	∞	∞	6	0

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,, \\ \pi_{kj}^{(k-1)} & \text{if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,. \end{cases}$$

π	1	2	3	4	5
1	N	1	**4**	2	1
2	N	N	N	2	2
3	N	3	N	2	2
4	4	3	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths– Floyd-Warshall

k = 4, i = 2, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0 , \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1 . \end{cases}$$

d21 = min(d21, d24+d41)=min(∞, 1 + 2)=3

d22 = min(d22, d24+d42)=min(0, 1 -1)

d23 = min(d23,d24+d43)=min(∞, 1 -5)=-4

d24 = min(d24,d24+d44)=min(1, 1+0)

d25 = min(d25,d24+d45)=min(7, 1 -2)=-1

D	**1**	**2**	**3**	**4**	**5**
1	0	3	-1	4	-4
2	∞	0	∞	1	7
3	∞	4	0	5	11
4	2	-1	-5	0	-2
5	∞	∞	∞	6	0

π	**1**	**2**	**3**	**4**	**5**
1	N	1	4	2	1
2	N	N	N	2	2
3	N	3	N	2	2
4	4	3	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths- Floyd-Warshall

k = 4, i = 2, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d21 = min(d21, d24+d41)=min(∞, 1 + 2)=3

d23 = min(d23,d24+d43)=min(∞, 1 -5)=-4

d25 = min(d25,d24+d45)=min(7, 1 -2)=-1

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,, \\ \pi_{kj}^{(k-1)} & \text{if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,. \end{cases}$$

D	**1**	**2**	**3**	**4**	**5**
1	0	3	-1	4	-4
2	**3**	0	**-4**	1	**-1**
3	∞	4	0	5	11
4	2	-1	-5	0	-2
5	∞	∞	∞	6	0

π	**1**	**2**	**3**	**4**	**5**
1	N	1	4	2	1
2	**4**	N	**4**	2	**1**
3	N	3	N	2	2
4	4	3	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths– Floyd-Warshall

k = 4, i = 3, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0 , \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1 . \end{cases}$$

d31 = min(d31, d34+d41)=min(∞, 5 + 2)=7

d32 = min(d32, d34+d42)=min(4, 5 -1)

d33 = min(d33,d34+d43)=min(0, 5 -5)

d34 = min(d34,d34+d44)=min(5, 5+0)

d35 = min(d35,d34+d45)=min(11, 5 -2)=3

D	**1**	**2**	**3**	**4**	**5**
1	0	3	-1	4	-4
2	3	0	-4	1	-1
3	∞	4	0	5	11
4	2	-1	-5	0	-2
5	∞	∞	∞	6	0

π	**1**	**2**	**3**	**4**	**5**
1	N	1	4	2	1
2	4	N	4	2	1
3	N	3	N	2	2
4	4	3	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths- Floyd-Warshall

k = 4, i = 3, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d31 = min(d31, d34+d41)=min(∞, 5 + 2)=7

d35 = min(d35,d34+d45)=min(11, 5 -2)=3

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,, \\ \pi_{kj}^{(k-1)} & \text{if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,. \end{cases}$$

D	**1**	**2**	**3**	**4**	**5**
1	0	3	-1	4	-4
2	3	0	-4	1	-1
3	**7**	4	0	5	**3**
4	2	-1	-5	0	-2
5	∞	∞	∞	6	0

π	**1**	**2**	**3**	**4**	**5**
1	N	1	4	2	1
2	4	N	4	2	1
3	**4**	3	N	2	**1**
4	4	3	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths– Floyd-Warshall

k = 4, i = 4, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d41 = min(d41, d44+d41)=min(2, 0 + 2)

d42 = min(d42, d44+d42)=min(-1, 0 -1)

d43 = min(d43,d44+d43)=min(-5, 0 -5)

d44 = min(d44,d44+d44)=min(0, 0+0)

d45 = min(d45,d44+d45)=min(-2, 0 -2)

D	1	2	3	4	5
1	0	3	-1	4	-4
2	3	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	∞	∞	∞	6	0

π	1	2	3	4	5
1	N	1	4	2	1
2	4	N	4	2	1
3	4	3	N	2	1
4	4	3	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths– Floyd-Warshall

k = 4, i = 5, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d51 = min(d51, d54+d41)=min(∞, 6 + 2)=8

d52 = min(d52, d54+d42)=min(∞, 6 -1)=5

d53 = min(d53,d54+d43)=min(∞, 6 -5)=1

d54 = min(d54,d54+d44)=min(6, 6+0)

d55 = min(d55,d54+d45)=min(0, 6 -2)

D	1	2	3	4	5
1	0	3	-1	4	-4
2	3	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	∞	∞	∞	6	0

π	1	2	3	4	5
1	N	1	4	2	1
2	4	N	4	2	1
3	4	3	N	2	1
4	4	3	4	N	1
5	N	N	N	5	N

Grafos

All pairs shortest paths– Floyd-Warshall

k = 4, i = 5, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d51 = min(d51, d54+d41)=min(∞, 6 + 2)=8

d52 = min(d52, d54+d42)=min(∞, 6 -1)=5

d53 = min(d53,d54+d43)=min(∞, 6 -5)=1

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,, \\ \pi_{kj}^{(k-1)} & \text{if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,. \end{cases}$$

D	**1**	**2**	**3**	**4**	**5**
1	0	3	-1	4	-4
2	3	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	**8**	**5**	**1**	6	0

π	**1**	**2**	**3**	**4**	**5**
1	N	1	4	2	1
2	4	N	4	2	1
3	4	3	N	2	1
4	4	3	4	N	1
5	**4**	**3**	**4**	5	N

Grafos

All pairs shortest paths- Floyd-Warshall

k = 5, i = 1, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0 , \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1 . \end{cases}$$

d11 = min(d11, d15+d51)=min(0, -4 + 8)

d12 = min(d12, d15+d52)=min(3, -4 + 5) =1

d13 = min(d13,d15+d53)=min(-1, -4 + 1)=-3

d14 = min(d14,d15+d54)=min(4, -4 + 6)=2

d15 = min(d15,d15+d55)=min(-4, -4 + 0)

D	**1**	**2**	**3**	**4**	**5**
1	0	3	-1	4	-4
2	3	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	8	5	1	6	0

π	**1**	**2**	**3**	**4**	**5**
1	N	1	4	2	1
2	4	N	4	2	1
3	4	3	N	2	1
4	4	3	4	N	1
5	4	3	4	5	N

Grafos

All pairs shortest paths- Floyd-Warshall

k = 5, i = 1, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d12 = min(d12, d15+d52)=min(3, -4 + 5) =1

d13 = min(d13,d15+d53)=min(-1, -4 + 1)=-3

d14 = min(d14,d15+d54)=min(4, -4 + 6)=2

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,, \\ \pi_{kj}^{(k-1)} & \text{if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,. \end{cases}$$

D	**1**	**2**	**3**	**4**	**5**
1	0	**1**	**-3**	**2**	-4
2	3	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	8	5	1	6	0

π	**1**	**2**	**3**	**4**	**5**
1	N	**3**	**4**	**5**	1
2	4	N	4	2	1
3	4	3	N	2	1
4	4	3	4	N	1
5	4	3	4	5	N

Grafos

All pairs shortest paths- Floyd-Warshall

k = 5, i = 2, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d21 = min(d21, d25+d51)=min(3, -1 + 8)

d22 = min(d22, d25+d52)=min(0, -1 + 5)

d23 = min(d23,d25+d53)=min(-4, -1 + 1)

d24 = min(d24,d25+d54)=min(1, -1+ 6)

d25 = min(d25,d25+d55)=min(-1, -1 + 0)

D	1	2	3	4	5
1	0	1	-3	2	-4
2	3	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	8	5	1	6	0

π	1	2	3	4	5
1	N	3	4	5	1
2	4	N	4	2	1
3	4	3	N	2	1
4	4	3	4	N	1
5	4	3	4	5	N

Grafos

All pairs shortest paths– Floyd-Warshall

k = 5, i = 3, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d31 = min(d31, d35+d51)=min(7, 3 + 8)

d32 = min(d32, d35+d52)=min(4, 3 + 5)

d33 = min(d33,d35+d53)=min(0, 3 + 1)

d34 = min(d34,d35+d54)=min(5, 3+ 6)

d35 = min(d35,d35+d55)=min(3, 3 + 0)

D	**1**	**2**	**3**	**4**	**5**
1	0	1	-3	2	-4
2	3	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	8	5	1	6	0

π	**1**	**2**	**3**	**4**	**5**
1	N	3	4	5	1
2	4	N	4	2	1
3	4	3	N	2	1
4	4	3	4	N	1
5	4	3	4	5	N

Grafos

All pairs shortest paths- Floyd-Warshall

k = 5, i = 4, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d41 = min(d41, d45+d51)=min(2, -2 + 8)

d42 = min(d42, d45+d52)=min(-1, - 2+ 5)

d43 = min(d43,d45+d53)=min(-5, -2 + 1)

d44 = min(d44,d45+d54)=min(0, -2+ 6)

d45 = min(d45,d45+d55)=min(-2, -2 + 0)

D	**1**	**2**	**3**	**4**	**5**
1	0	1	-3	2	-4
2	3	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	8	5	1	6	0

π	**1**	**2**	**3**	**4**	**5**
1	N	3	4	5	1
2	4	N	4	2	1
3	4	3	N	2	1
4	4	3	4	N	1
5	4	3	4	5	N

Grafos

All pairs shortest paths- Floyd-Warshall

k = 5, i = 5, j = 1…5

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

d51 = min(d51, d55+d51)=min(8, 0 + 8)

d52 = min(d52, d55+d52)=min(5, 0 + 5)

d53 = min(d53,d55+d53)=min(1, 0 + 1)

d54 = min(d54,d55+d54)=min(6, 0+ 6)

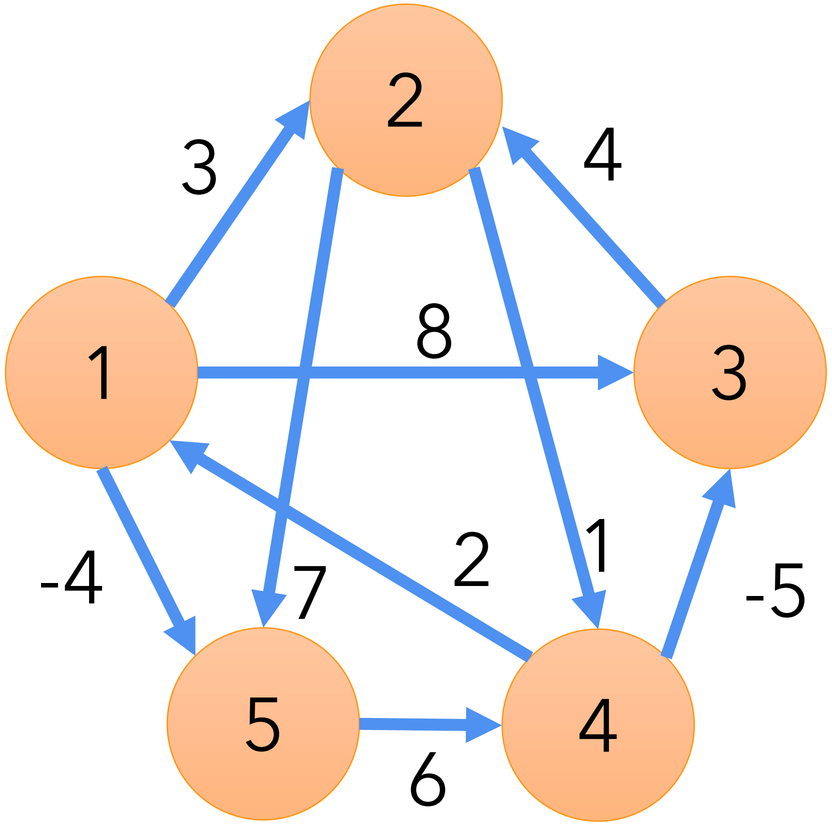
d55 = min(d55,d55+d55)=min(0, 0 + 0)

D	1	2	3	4	5
1	0	1	-3	2	-4
2	3	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	8	5	1	6	0

π	1	2	3	4	5
1	N	3	4	5	1
2	4	N	4	2	1
3	4	3	N	2	1
4	4	3	4	N	1
5	4	3	4	5	N

Grafos

All pairs shortest paths– Floyd-Warshall



D	1	2	3	4	5
1	0	1	-3	2	-4
2	3	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	8	5	1	6	0

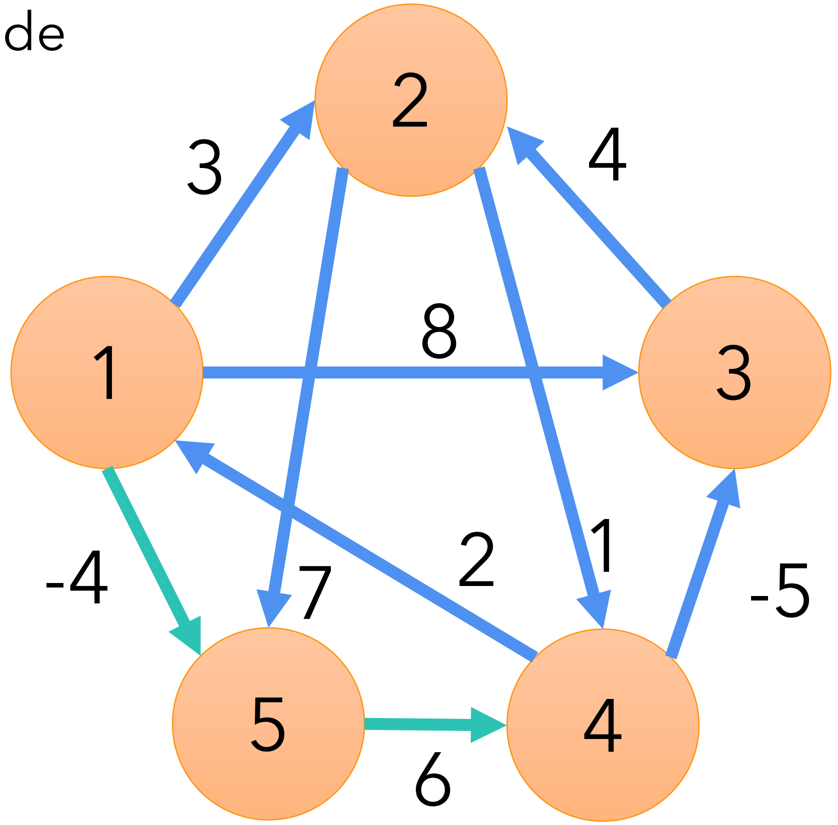
π	1	2	3	4	5
1	N	3	4	5	1
2	4	N	4	2	1
3	4	3	N	2	1
4	4	3	4	N	1
5	4	3	4	5	N

Grafos

All pairs shortest paths– Floyd-Warshall

Camino de

1 a 4



D	1	2	3	4	5
1	0	1	-3	2	-4
2	3	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	8	5	1	6	0

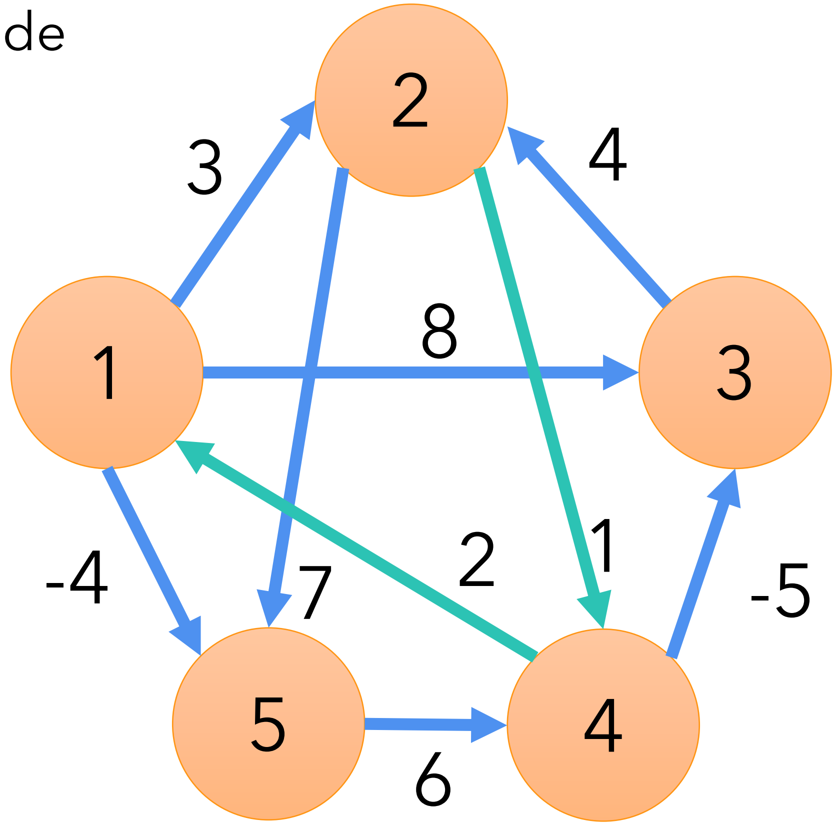
π	1	2	3	4	5
1	N	3	4	5	1
2	4	N	4	2	1
3	4	3	N	2	1
4	4	3	4	N	1
5	4	3	4	5	N

Grafos

All pairs shortest paths– Floyd-Warshall

Camino de

2 a 1



D	1	2	3	4	5
1	0	1	-3	2	-4
2	3	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	8	5	1	6	0

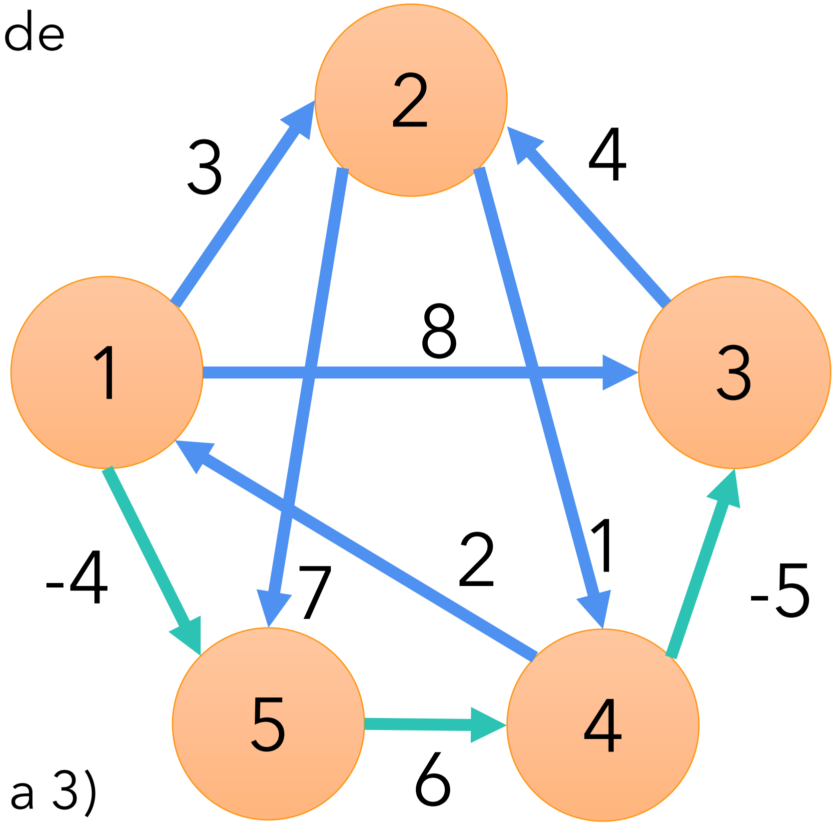
π	1	2	3	4	5
1	N	3	4	5	1
2	4	N	4	2	1
3	4	3	N	2	1
4	4	3	4	N	1
5	4	3	4	5	N

Grafos

All pairs shortest paths– Floyd-Warshall

Camino de

1 a 3



(1 a 4)(4 a 3)

(1 a 5) (5 a 4)(4 a 3)

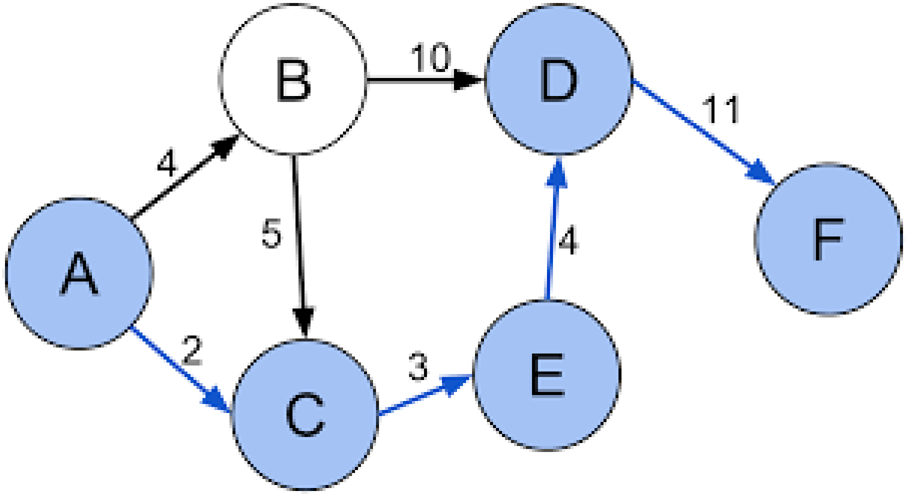
D	1	2	3	4	5
1	0	1	-3	2	-4
2	3	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	8	5	1	6	0

π	1	2	3	4	5
1	N	3	4	5	1
2	4	N	4	2	1
3	4	3	N	2	1
4	4	3	4	N	1
5	4	3	4	5	N

Grafos

Descripción del problema

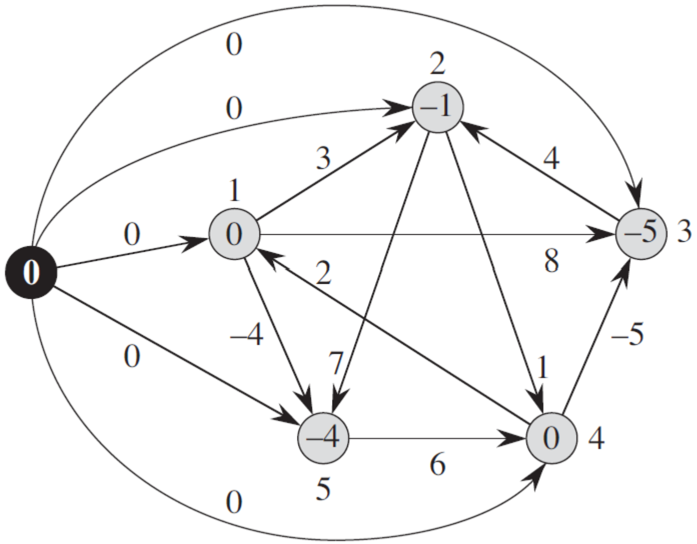
- El camino más corto desde un vértice s a todos los demás vértices del grado:
  - Bellman-Ford O(VE)
  - Dijkstra O(V + E log V)
- El camino más corto entre todos los pares del vértice del grafo
  - Floyd-Warshall
  - Johnson



Grafos

All pairs shortest paths– Jhonson

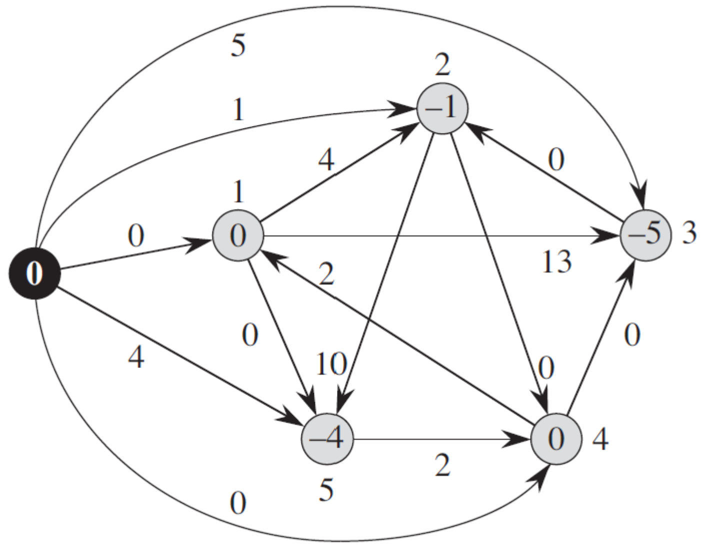
- Hace uso de los algoritmos Bellman-Ford y Dijkstra
- Recomendado solo para grafos espacidos
- El algoritmo inserta un nuevo vertice *s* conectado a todos los vértices del grafo con pesos igual a 0



Grafos

All pairs shortest paths– Jhonson

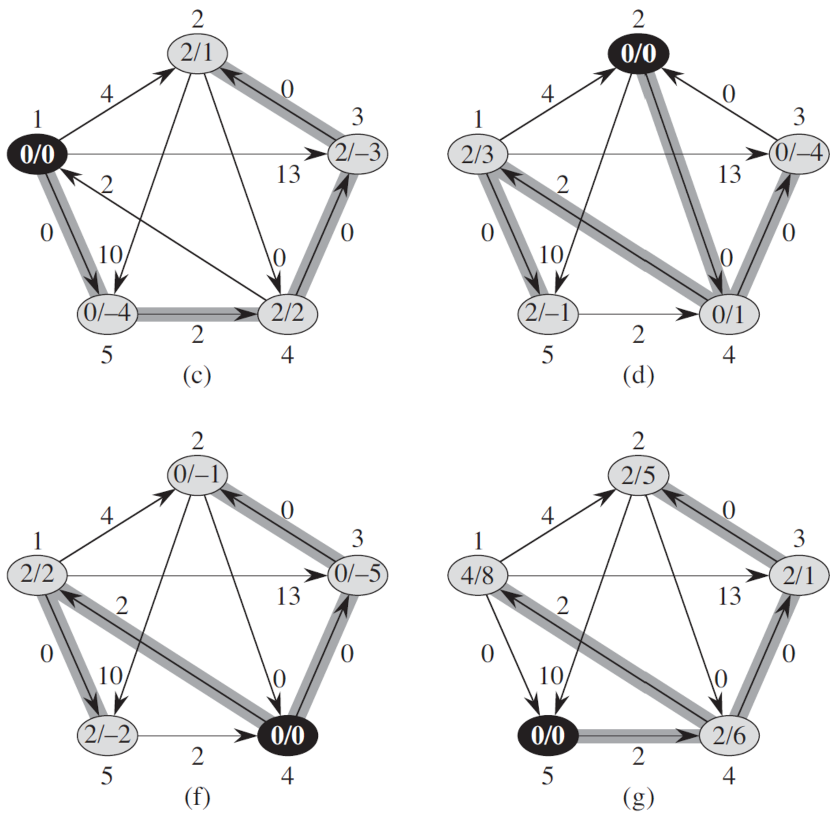
- Una vez inicializado, llama el algoritmo de Bellman-Ford para calcular el peso de cada vértice (equivalente a la solución del algoritmo de Bellman-Ford)



Grafos

All pairs shortest paths– Jhonson

- Aplica una función para actualizar pesos eliminando el nodo s insertado
- Se aplica el método de Dijkstra para cada nodo



Grafos

All pairs shortest paths– Jhonson

JOHNSON(G, w)

compute G', where $G'.V = G.V \cup \{s\}$,
    $G'.E = G.E \cup \{(s, v) : v \in G.V\}$, and
    $w(s, v) = 0$ for all $v \in G.V$
if BELLMAN-FORD(G', w, s) == FALSE
    print "the input graph contains a negative-weight cycle"
else for each vertex $v \in G'.V$
        set $h(v)$ to the value of $\delta(s, v)$
            computed by the Bellman-Ford algorithm
    for each edge $(u, v) \in G'.E$
        $\widehat{w}(u, v) = w(u, v) + h(u) - h(v)$
    let $D = (d_{uv})$ be a new $n \times n$ matrix
    for each vertex $u \in G.V$
        run DIJKSTRA$(G, \widehat{w}, u)$ to compute $\widehat{\delta}(u, v)$ for all $v \in G.V$
        for each vertex $v \in G.V$
            $d_{uv} = \widehat{\delta}(u, v) + h(v) - h(u)$
    return D