

Programación Dinámica: Ejercicio 5

Se tiene un grafo dirigido $G = \langle N, A \rangle$, siendo $N = \{1, \dots, n\}$ el conjunto de nodos y $A \subseteq N \times N$ el conjunto de aristas. Sea M la matriz de adyacencia del grafo G , es decir, $M[i, j] = \text{TRUE}$ si existe la arista (i, j) , y $M[i, j] = \text{FALSE}$ en caso contrario.

Se está interesado en saber desde qué vértices se puede acceder a cualquier otro vértice (mediante un camino de cualquier longitud), mediante el algoritmo de Warshall, por lo que

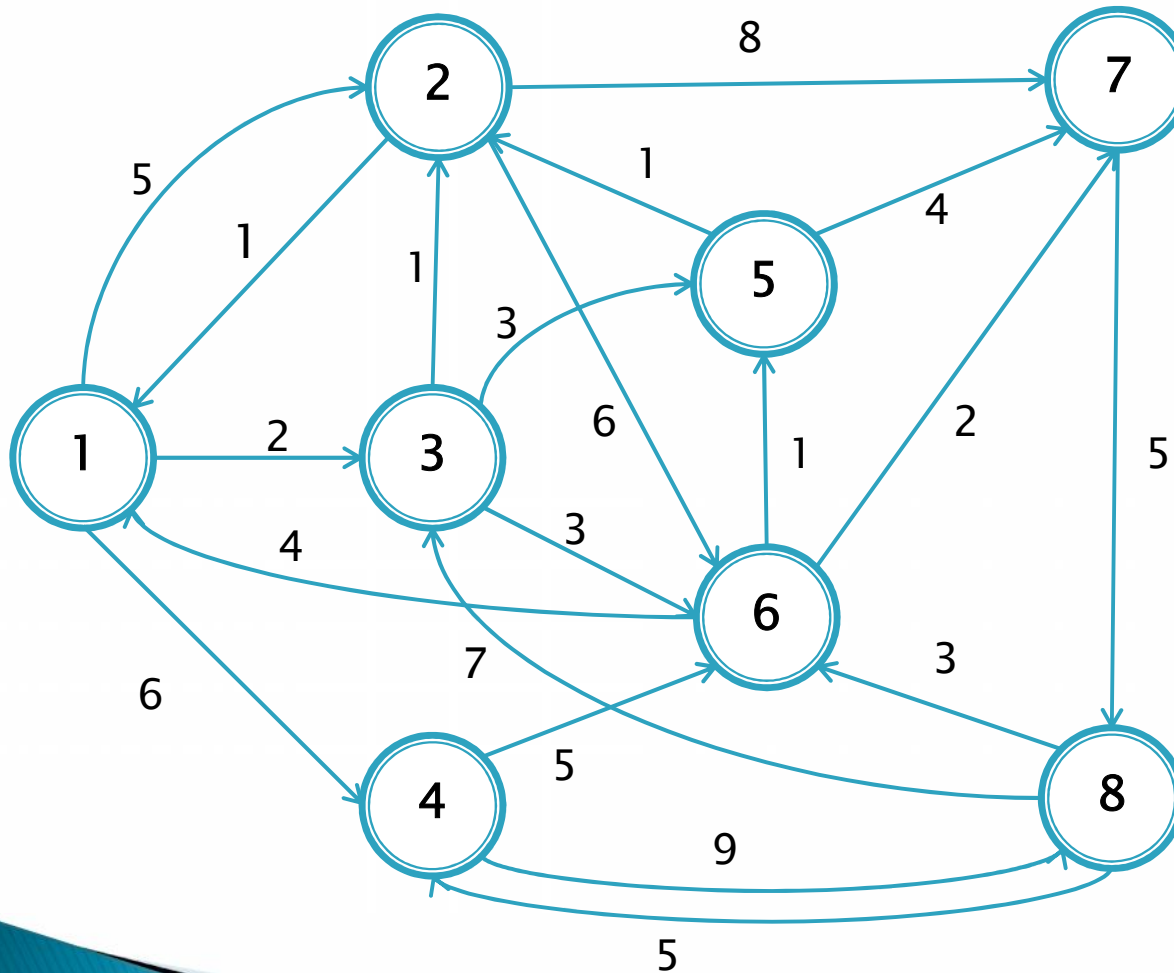
- ▶ Hay que obtener una matriz de caminos C de manera que $C[i, j] = \text{TRUE}$ si existe un camino (de cualquier longitud) entre los nodos i y j , y $C[i, j] = \text{FALSE}$ si no existe forma de llegar de i a j .
- ▶ Para ello, hay que ir considerando con un enfoque Bottom-Up una cantidad mayor de nodos. Se empieza intentando ir directamente de un nodo a otro, después se intenta encontrar los caminos que pueden utilizar el vértice 1, a continuación los que pueden usar los vértices 1 y 2, después lo que se consiguen con los vértices 1, 2 y 3, etc, hasta llegar a obtener los caminos que pueden utilizar todos los vértices de 1 hasta n .

Implementar un método de Programación Dinámica en los términos anteriores, que obtenga la matriz de existencia de caminos C de un grafo teniendo como datos la cantidad de nodos n y la matriz de adyacencia M .

Programación Dinámica: Ejercicio 5

Ejemplo 1:

Matriz M de entrada →



Matriz C que obtengo →
inicialmente a partir de M

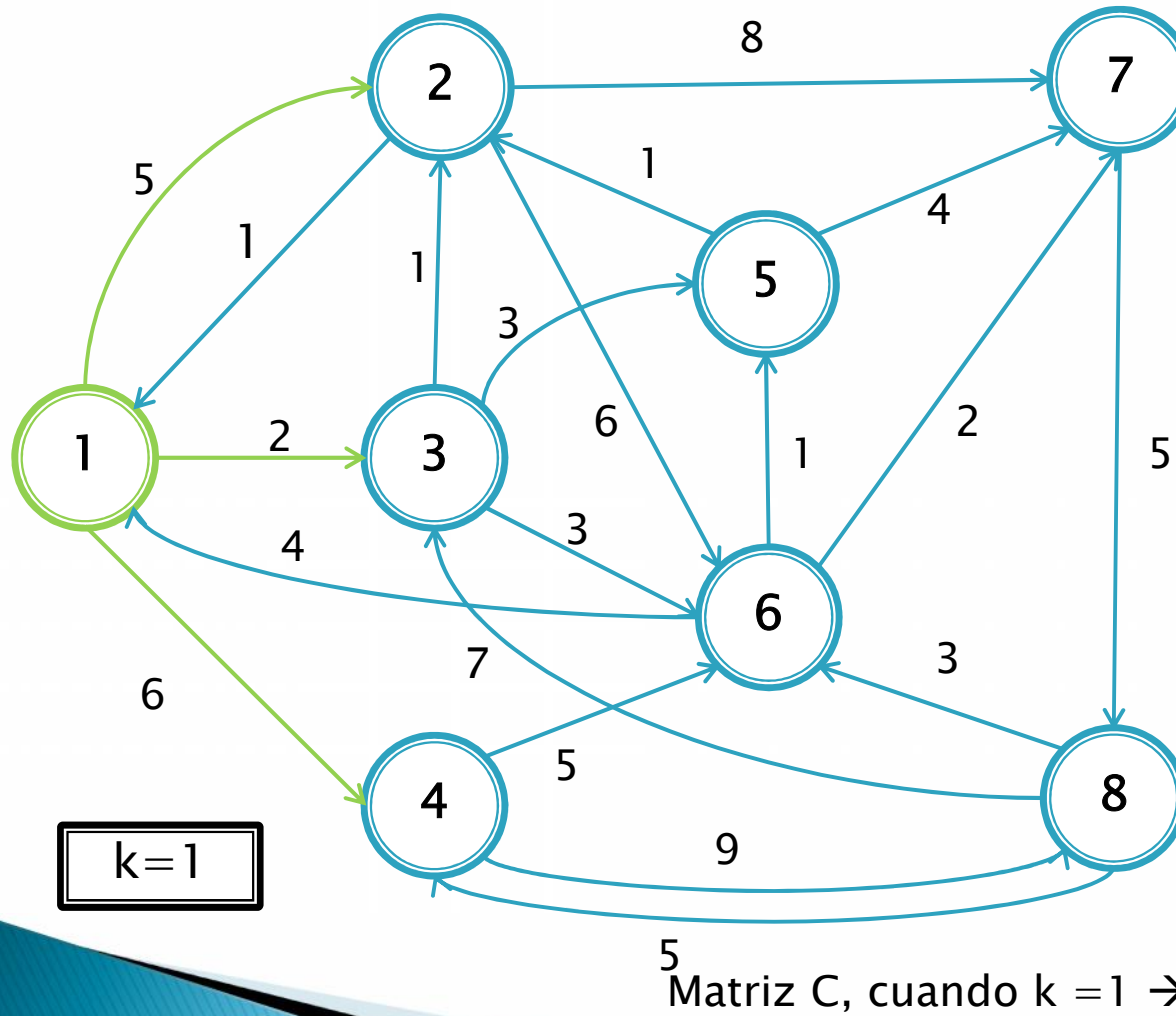
	1	2	3	4	5	6	7	8
1	0	5	2	6	∞	∞	∞	∞
2	1	0	∞	∞	∞	6	8	∞
3	∞	1	0	∞	3	3	∞	∞
4	∞	∞	∞	0	∞	5	∞	9
5	∞	1	∞	∞	0	∞	4	∞
6	4	∞	∞	∞	1	0	2	∞
7	∞	∞	∞	∞	∞	∞	0	5
8	∞	∞	7	5	∞	3	∞	0

	1	2	3	4	5	6	7	8
1	T	T	T	T	F	F	F	F
2	T	T	F	F	F	T	T	F
3	F	T	T	F	T	T	F	F
4	F	F	F	T	F	T	F	T
5	F	T	F	F	T	F	T	F
6	T	F	F	F	T	T	T	F
7	F	F	F	F	F	F	T	T
8	F	F	T	T	F	T	F	T

Programación Dinámica: Ejercicio 5

Ejemplo 1:

Matriz M de entrada →



Matriz C, cuando $k = 1 \rightarrow$

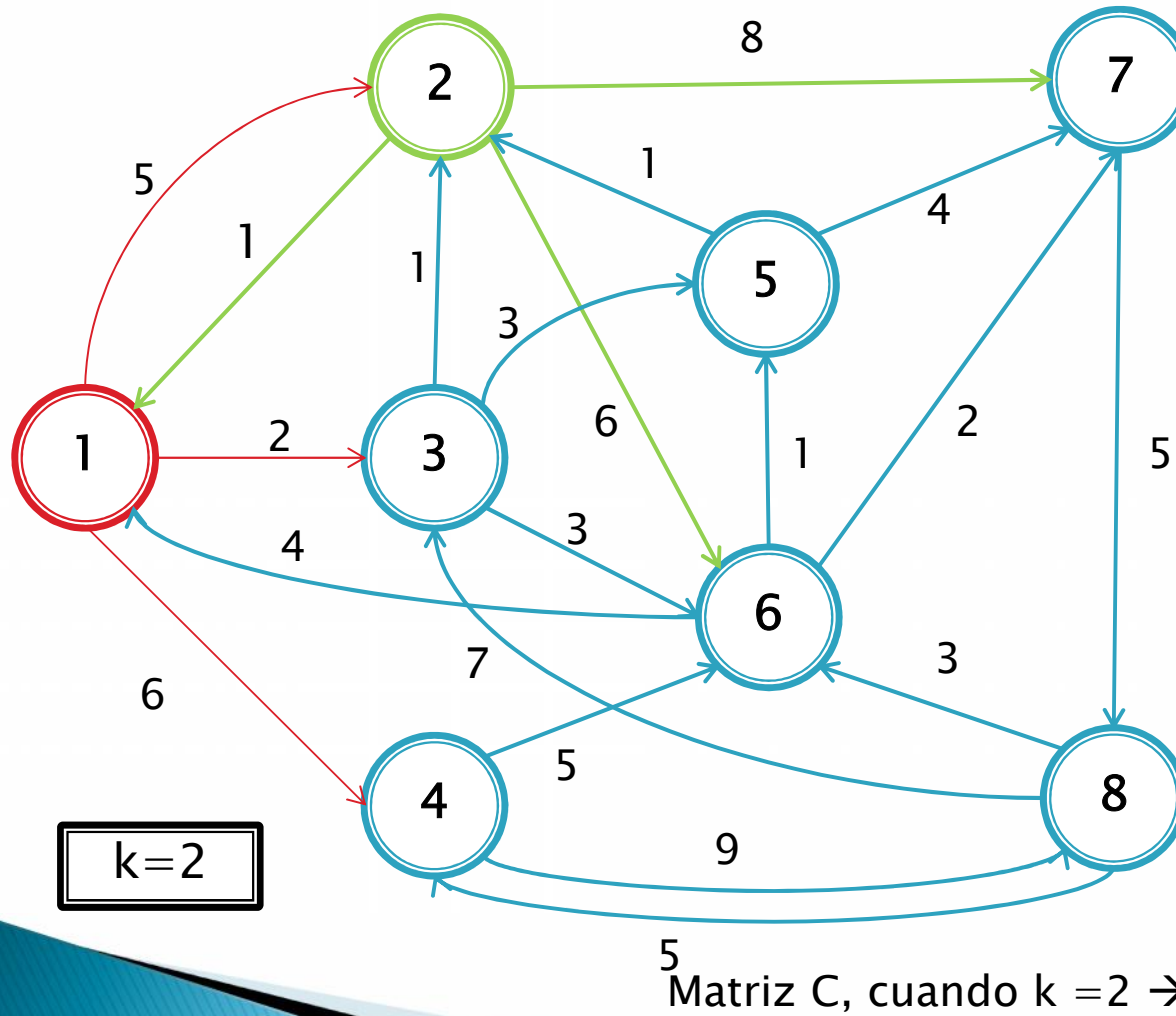
	1	2	3	4	5	6	7	8
1	0	5	2	6	∞	∞	∞	∞
2	1	0	∞	∞	∞	6	8	∞
3	∞	1	0	∞	3	3	∞	∞
4	∞	∞	∞	0	∞	5	∞	9
5	∞	1	∞	∞	0	∞	4	∞
6	4	∞	∞	∞	1	0	2	∞
7	∞	∞	∞	∞	∞	∞	0	5
8	∞	∞	7	5	∞	3	∞	0

	1	2	3	4	5	6	7	8
1	T	T	T	T	F	F	F	F
2	T	T	T	T	F	T	T	F
3	F	T	T	F	T	T	F	F
4	F	F	F	T	F	T	F	T
5	F	T	F	F	T	F	T	F
6	T	T	T	T	T	T	T	F
7	F	F	F	F	F	F	T	T
8	F	F	T	T	F	T	F	T

Programación Dinámica: Ejercicio 5

Ejemplo 1:

Matriz M de entrada →



Matriz C, cuando $k = 2 \rightarrow$

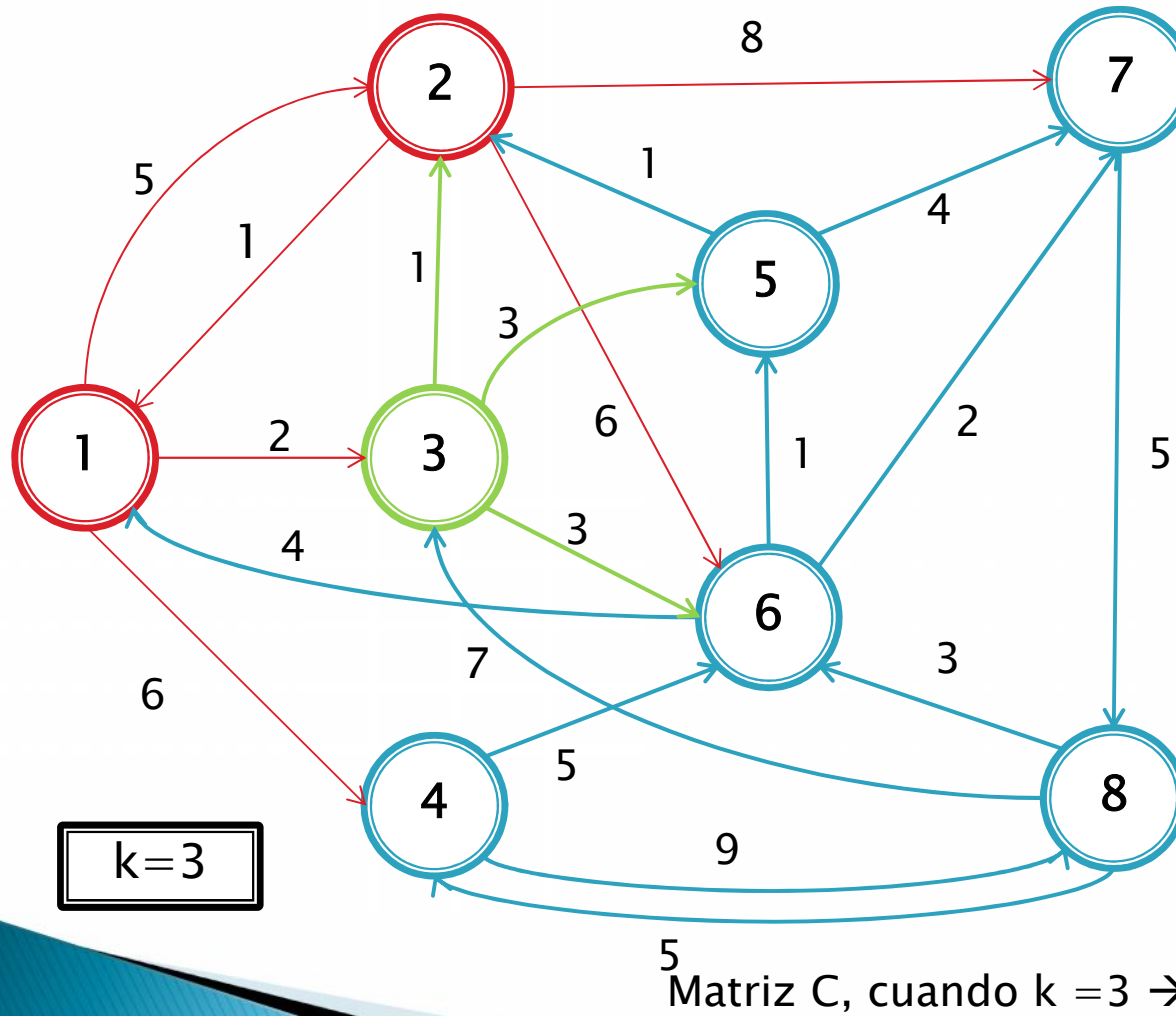
	1	2	3	4	5	6	7	8
1	0	5	2	6	∞	∞	∞	∞
2	1	0	∞	∞	∞	6	8	∞
3	∞	1	0	∞	3	3	∞	∞
4	∞	∞	∞	0	∞	5	∞	9
5	∞	1	∞	∞	0	∞	4	∞
6	4	∞	∞	∞	1	0	2	∞
7	∞	∞	∞	∞	∞	∞	0	5
8	∞	∞	7	5	∞	3	∞	0

	1	2	3	4	5	6	7	8
1	T	T	T	T	F	T	T	F
2	T	T	T	T	F	T	T	F
3	T	T	T	T	T	T	T	F
4	F	F	F	T	F	T	F	T
5	T	T	T	T	T	T	T	F
6	T	T	T	T	T	T	T	F
7	F	F	F	F	F	F	T	T
8	F	F	T	T	F	T	F	T

Programación Dinámica: Ejercicio 5

Ejemplo 1:

Matriz M de entrada →



Matriz C, cuando $k = 3 \rightarrow$

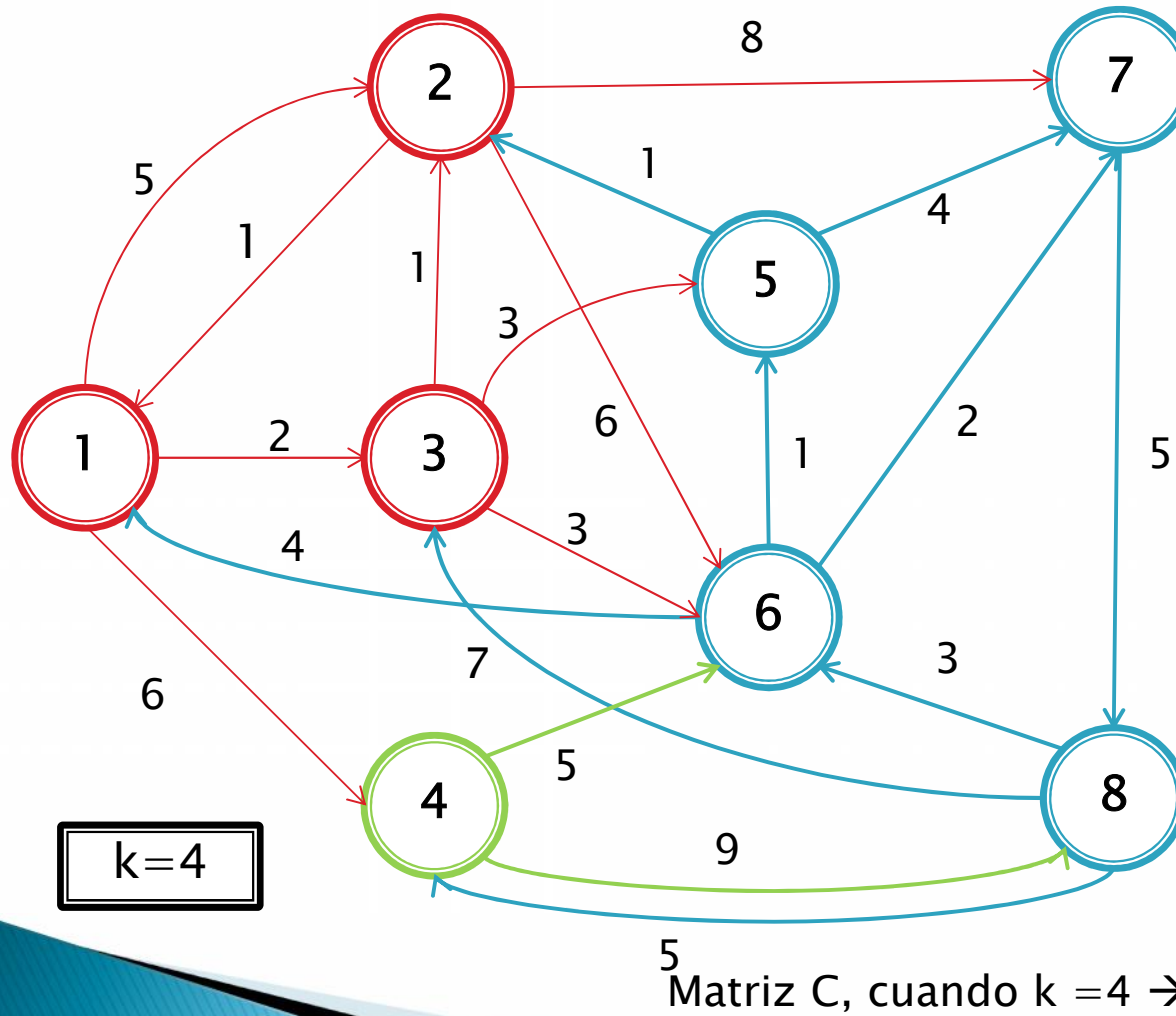
	1	2	3	4	5	6	7	8
1	0	5	2	6	∞	∞	∞	∞
2	1	0	∞	∞	∞	6	8	∞
3	∞	1	0	∞	3	3	∞	∞
4	∞	∞	∞	0	∞	5	∞	9
5	∞	1	∞	∞	0	∞	4	∞
6	4	∞	∞	∞	1	0	2	∞
7	∞	∞	∞	∞	∞	∞	0	5
8	∞	∞	7	5	∞	3	∞	0

	1	2	3	4	5	6	7	8
1	T	T	T	T	T	T	T	F
2	T	T	T	T	T	T	T	F
3	T	T	T	T	T	T	T	F
4	F	F	F	T	F	T	F	T
5	T	T	T	T	T	T	T	F
6	T	T	T	T	T	T	T	F
7	F	F	F	F	F	F	T	T
8	T	T	T	T	T	T	T	T

Programación Dinámica: Ejercicio 5

Ejemplo 1:

Matriz M de entrada →



Matriz C, cuando $k = 4 \rightarrow$

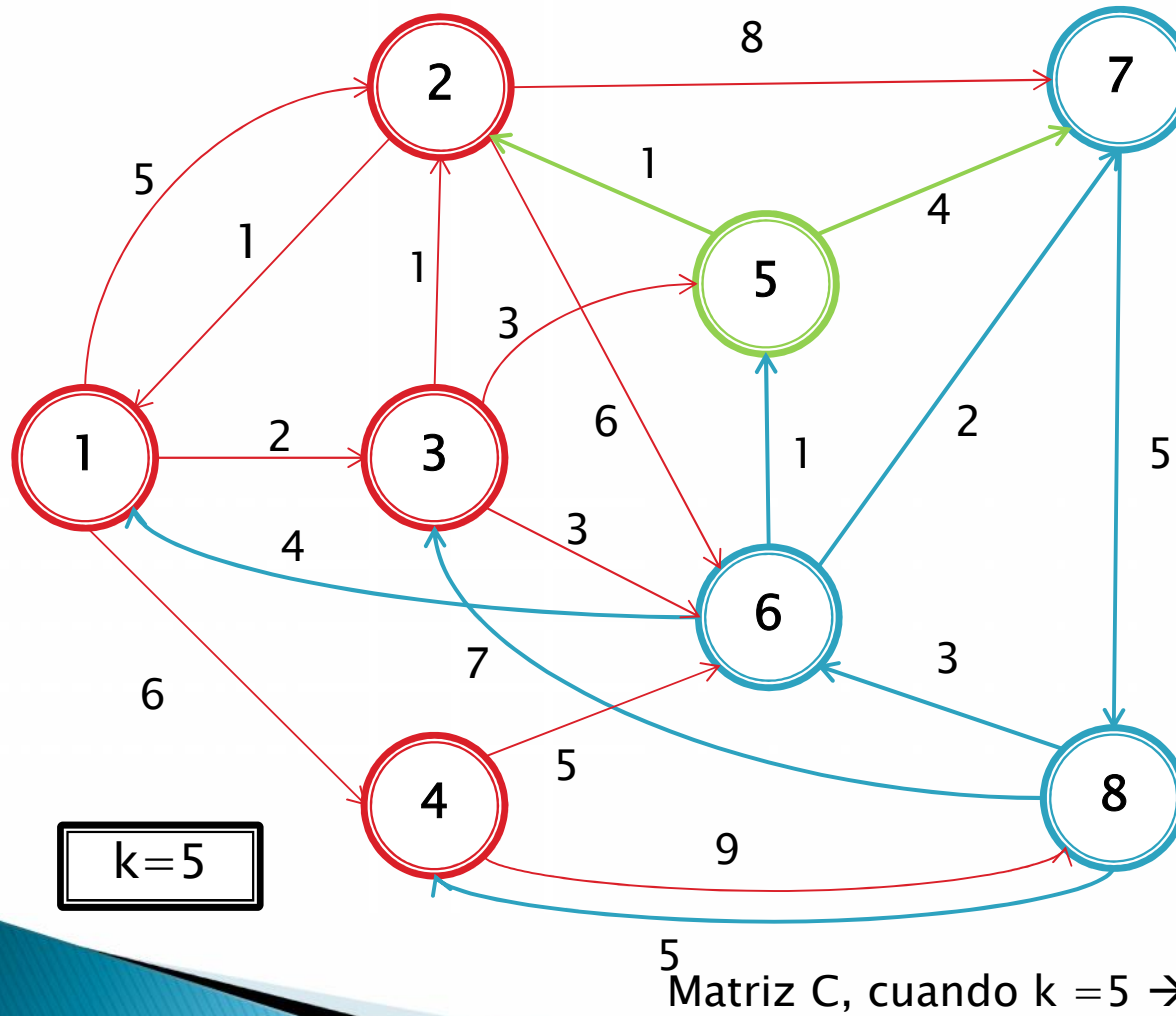
	1	2	3	4	5	6	7	8
1	0	5	2	6	∞	∞	∞	∞
2	1	0	∞	∞	∞	6	8	∞
3	∞	1	0	∞	3	3	∞	∞
4	∞	∞	∞	0	∞	5	∞	9
5	∞	1	∞	∞	0	∞	4	∞
6	4	∞	∞	∞	1	0	2	∞
7	∞	∞	∞	∞	∞	∞	0	5
8	∞	∞	7	5	∞	3	∞	0

	1	2	3	4	5	6	7	8
1	T	T	T	T	T	T	T	T
2	T	T	T	T	T	T	T	T
3	T	T	T	T	T	T	T	T
4	F	F	F	T	F	T	F	T
5	T	T	T	T	T	T	T	T
6	T	T	T	T	T	T	T	T
7	F	F	F	F	F	F	T	T
8	T	T	T	T	T	T	T	T

Programación Dinámica: Ejercicio 5

Ejemplo 1:

Matriz M de entrada →



Matriz C, cuando $k = 5 \rightarrow$

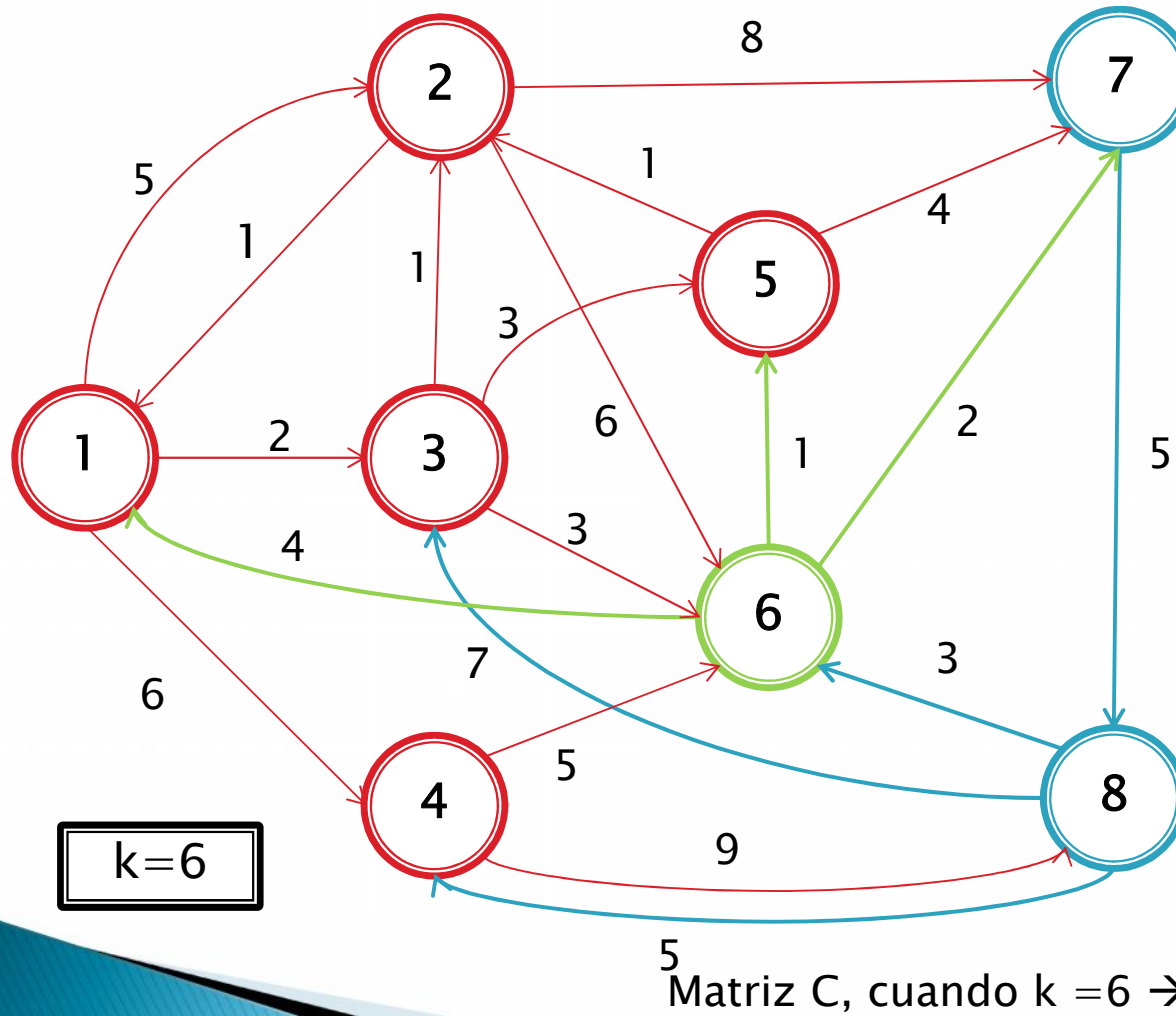
	1	2	3	4	5	6	7	8
1	0	5	2	6	∞	∞	∞	∞
2	1	0	∞	∞	∞	6	8	∞
3	∞	1	0	∞	3	3	∞	∞
4	∞	∞	∞	0	∞	5	∞	9
5	∞	1	∞	∞	0	∞	4	∞
6	4	∞	∞	∞	1	0	2	∞
7	∞	∞	∞	∞	∞	∞	0	5
8	∞	∞	7	5	∞	3	∞	0

	1	2	3	4	5	6	7	8
1	T	T	T	T	T	T	T	T
2	T	T	T	T	T	T	T	T
3	T	T	T	T	T	T	T	T
4	F	F	F	T	F	T	F	T
5	T	T	T	T	T	T	T	T
6	T	T	T	T	T	T	T	T
7	F	F	F	F	F	F	T	T
8	T	T	T	T	T	T	T	T

Programación Dinámica: Ejercicio 5

Ejemplo 1:

Matriz M de entrada →



Matriz C, cuando $k = 6 \rightarrow$

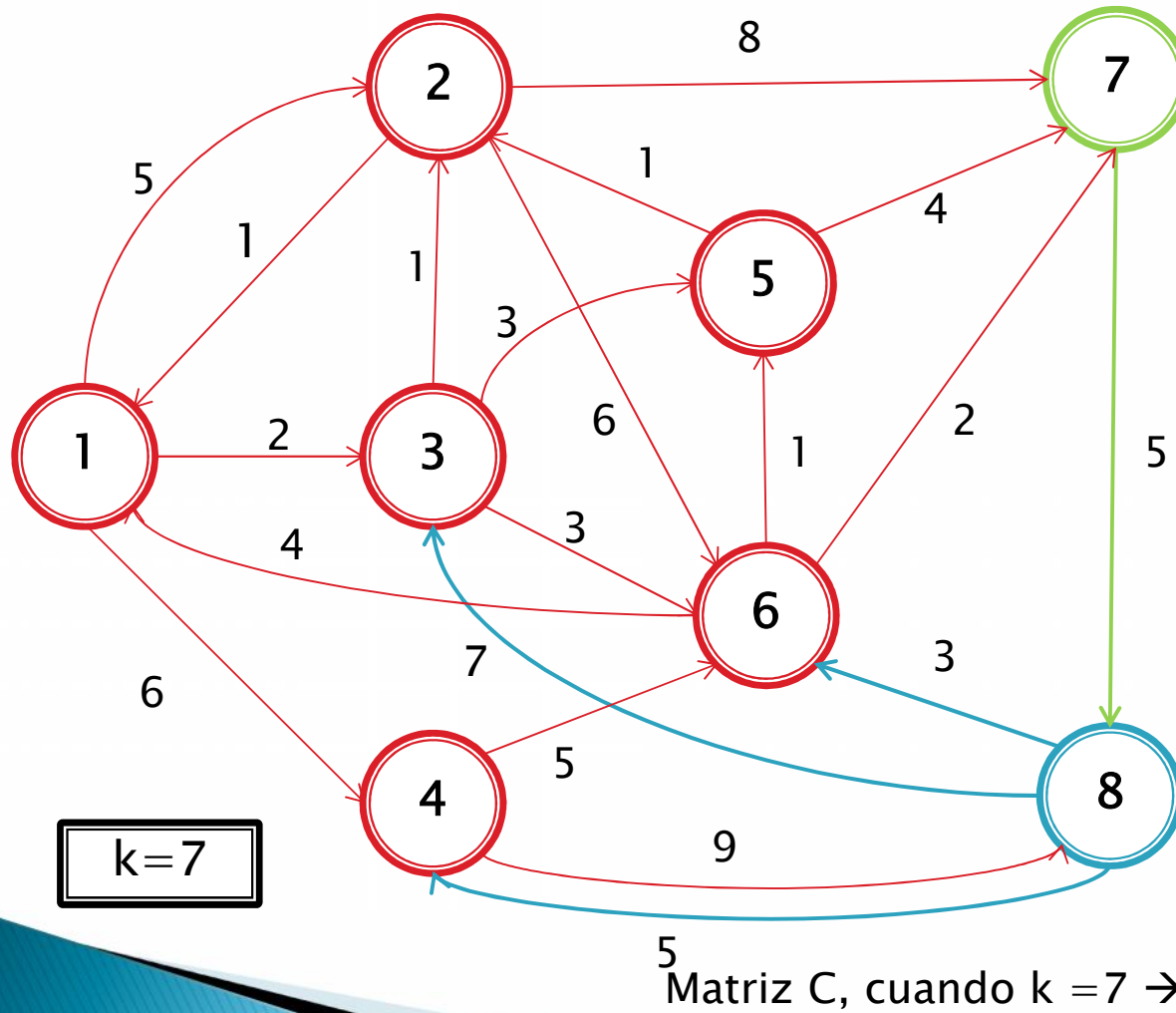
	1	2	3	4	5	6	7	8
1	0	5	2	6	∞	∞	∞	∞
2	1	0	∞	∞	∞	6	8	∞
3	∞	1	0	∞	3	3	∞	∞
4	∞	∞	∞	0	∞	5	∞	9
5	∞	1	∞	∞	0	∞	4	∞
6	4	∞	∞	∞	1	0	2	∞
7	∞	∞	∞	∞	∞	∞	0	5
8	∞	∞	7	5	∞	3	∞	0

	1	2	3	4	5	6	7	8
1	T	T	T	T	T	T	T	T
2	T	T	T	T	T	T	T	T
3	T	T	T	T	T	T	T	T
4	T	T	T	T	T	T	T	T
5	T	T	T	T	T	T	T	T
6	T	T	T	T	T	T	T	T
7	F	F	F	F	F	F	T	T
8	T	T	T	T	T	T	T	T

Programación Dinámica: Ejercicio 5

Ejemplo 1:

Matriz M de entrada →



Matriz C, cuando $k = 7 \rightarrow$

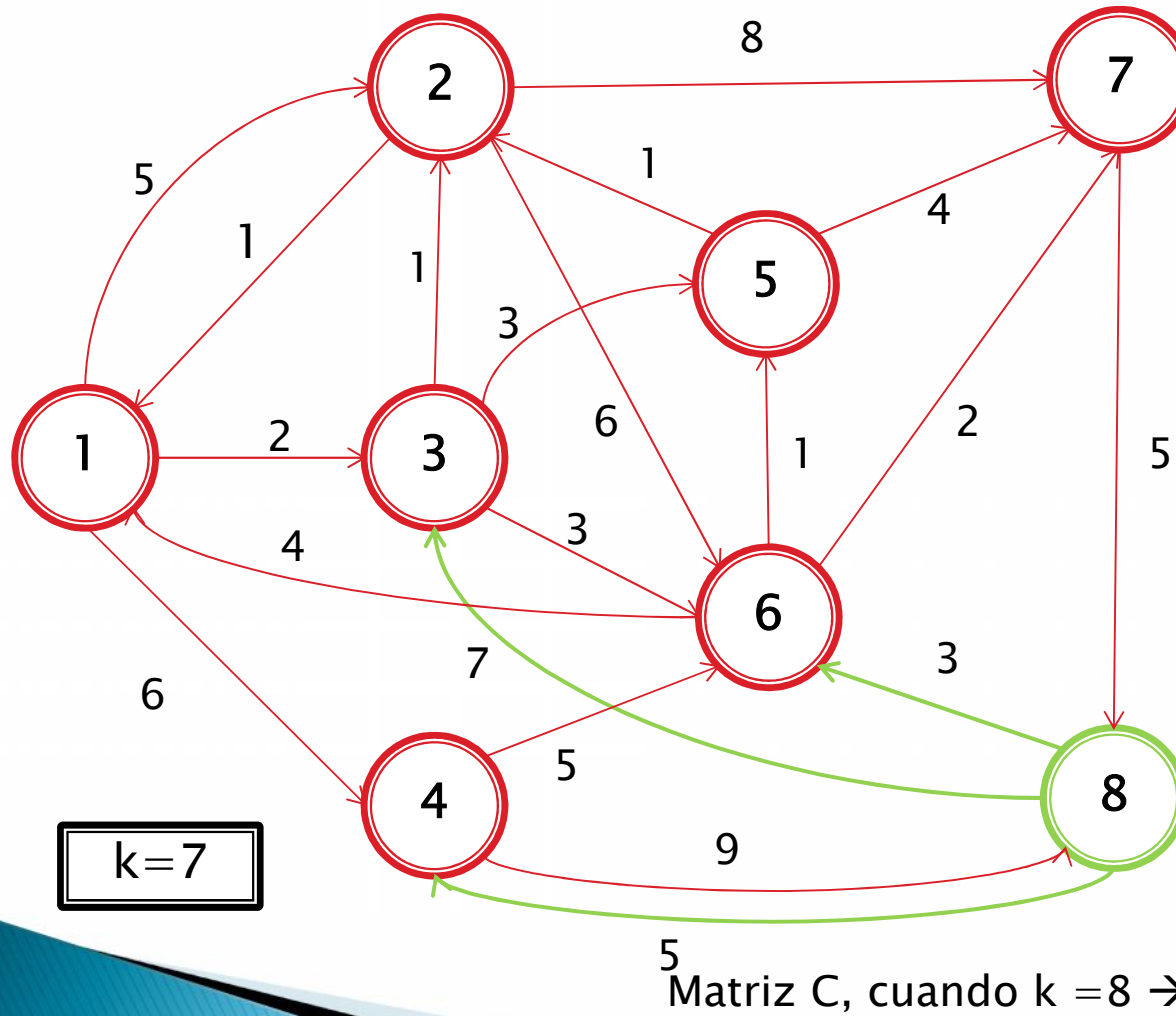
	1	2	3	4	5	6	7	8
1	0	5	2	6	∞	∞	∞	∞
2	1	0	∞	∞	∞	6	8	∞
3	∞	1	0	∞	3	3	∞	∞
4	∞	∞	∞	0	∞	5	∞	9
5	∞	1	∞	∞	0	∞	4	∞
6	4	∞	∞	∞	1	0	2	∞
7	∞	∞	∞	∞	∞	∞	0	5
8	∞	∞	7	5	∞	3	∞	0

	1	2	3	4	5	6	7	8
1	T	T	T	T	T	T	T	T
2	T	T	T	T	T	T	T	T
3	T	T	T	T	T	T	T	T
4	T	T	T	T	T	T	T	T
5	T	T	T	T	T	T	T	T
6	T	T	T	T	T	T	T	T
7	F	F	F	F	F	F	T	T
8	T	T	T	T	T	T	T	T

Programación Dinámica: Ejercicio 5

Ejemplo 1:

Matriz M de entrada →



Matriz C, cuando $k = 8 \rightarrow$

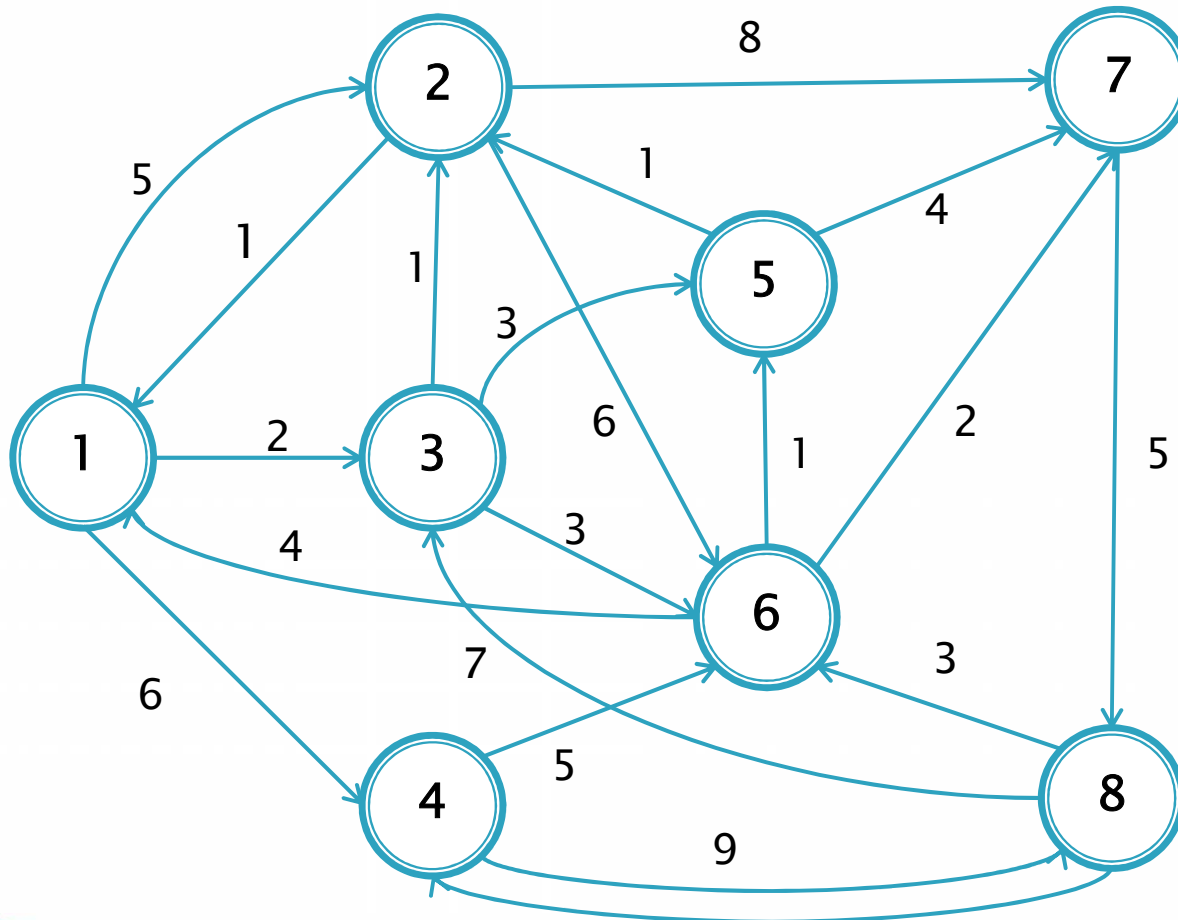
	1	2	3	4	5	6	7	8
1	0	5	2	6	∞	∞	∞	∞
2	1	0	∞	∞	∞	6	8	∞
3	∞	1	0	∞	3	3	∞	∞
4	∞	∞	∞	0	∞	5	∞	9
5	∞	1	∞	∞	0	∞	4	∞
6	4	∞	∞	∞	1	0	2	∞
7	∞	∞	∞	∞	∞	∞	0	5
8	∞	∞	7	5	∞	3	∞	0

	1	2	3	4	5	6	7	8
1	T	T	T	T	T	T	T	T
2	T	T	T	T	T	T	T	T
3	T	T	T	T	T	T	T	T
4	T	T	T	T	T	T	T	T
5	T	T	T	T	T	T	T	T
6	T	T	T	T	T	T	T	T
7	T	T	T	T	T	T	T	T
8	T	T	T	T	T	T	T	T

Programación Dinámica: Ejercicio 5

Ejemplo 1:

Matriz C de inicio →



Matriz C que obtengo al →
finalizar el algoritmo de
Warshall

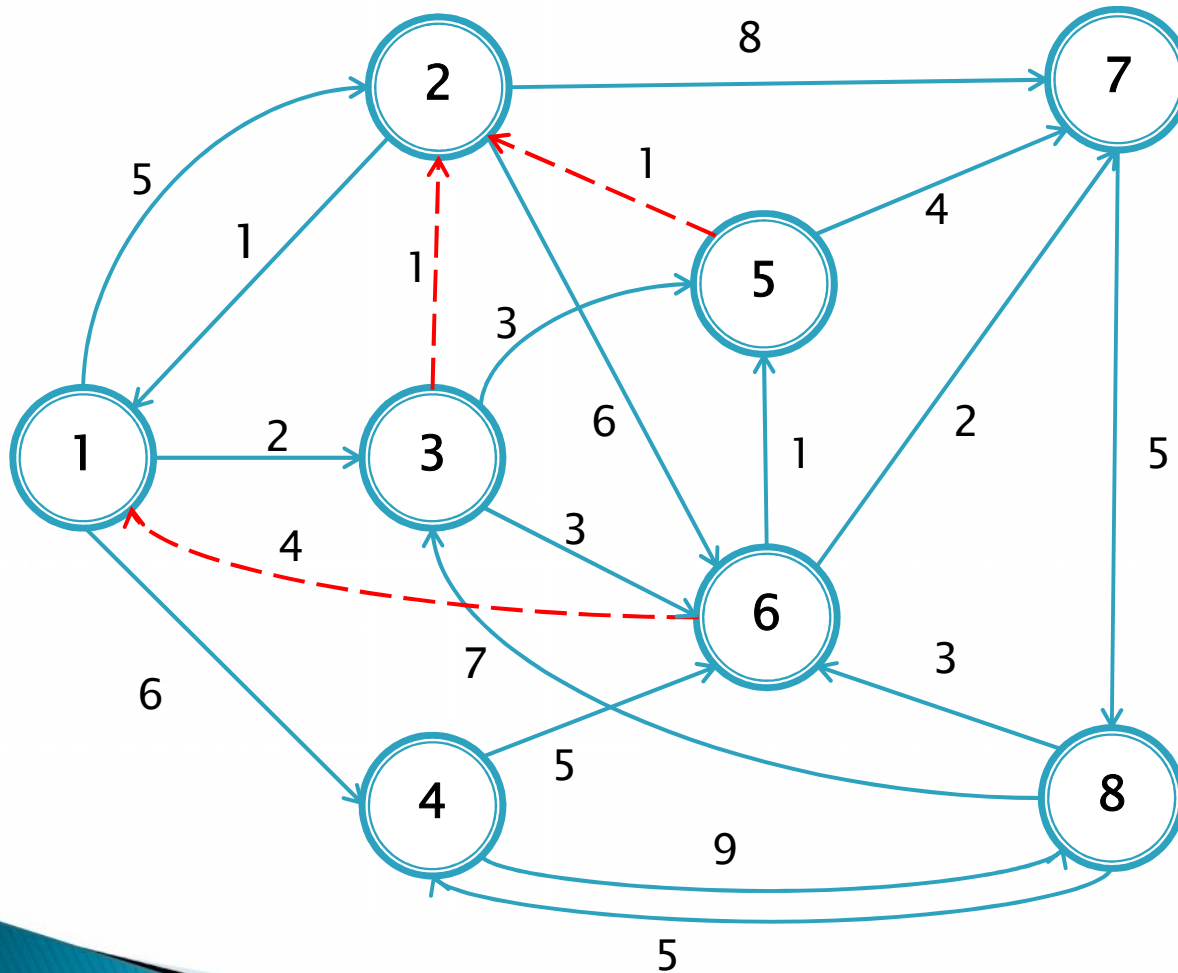
	1	2	3	4	5	6	7	8
1	T	T	T	T	F	F	F	F
2	T	T	F	F	F	T	T	F
3	F	T	T	F	T	T	F	F
4	F	F	F	T	F	T	F	T
5	F	T	F	F	T	F	T	F
6	T	F	F	F	T	T	T	F
7	F	F	F	F	F	F	T	T
8	F	F	T	T	F	T	F	T

	1	2	3	4	5	6	7	8
1	T	T	T	T	T	T	T	T
2	T	T	T	T	T	T	T	T
3	T	T	T	T	T	T	T	T
4	T	T	T	T	T	T	T	T
5	T	T	T	T	T	T	T	T
6	T	T	T	T	T	T	T	T
7	T	T	T	T	T	T	T	T
8	T	T	T	T	T	T	T	T

Programación Dinámica: Ejercicio 5

Ejemplo 2: elimino las conexiones indicadas

Matriz M de entrada →



Matriz C que obtengo → inicialmente a partir de M

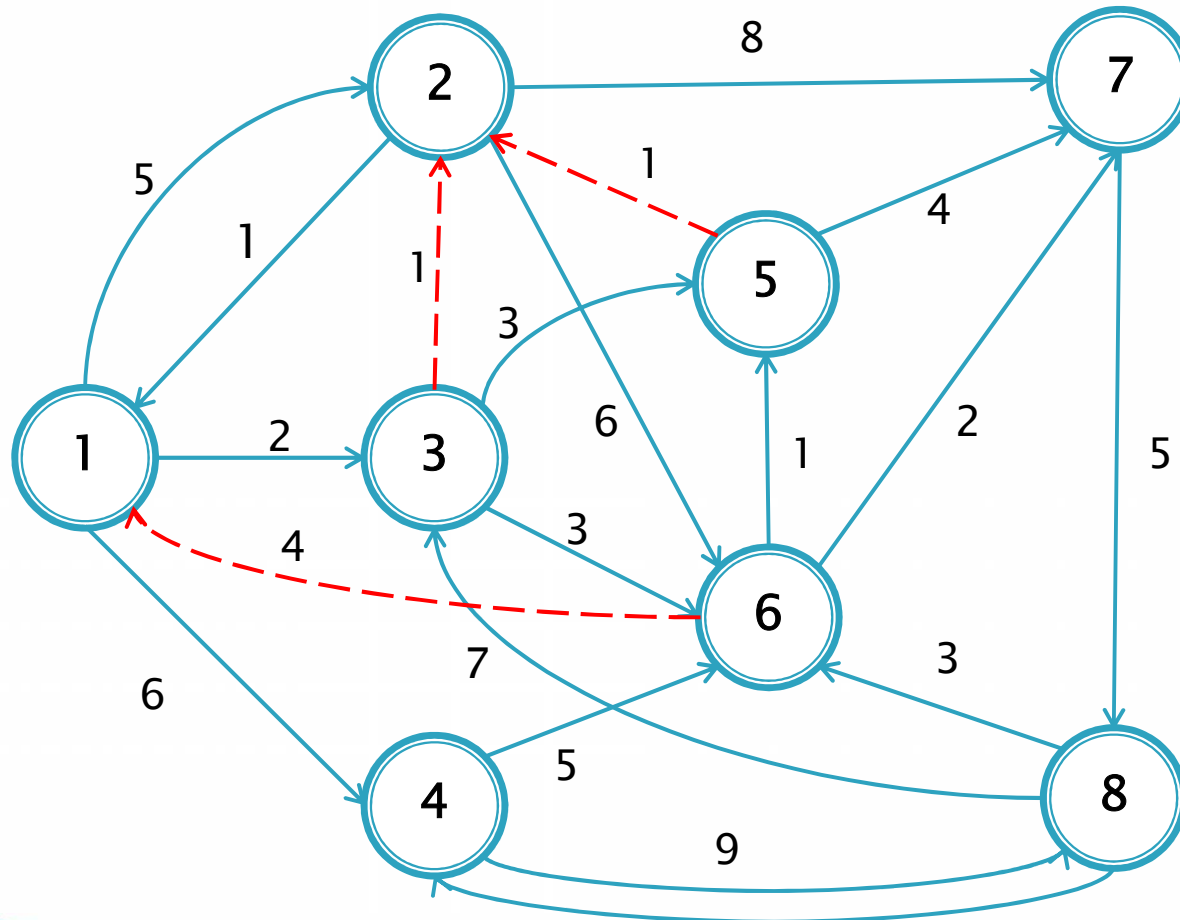
	1	2	3	4	5	6	7	8
1	0	5	2	6	∞	∞	∞	∞
2	1	0	∞	∞	∞	6	8	∞
3	∞	∞	0	∞	3	3	∞	∞
4	∞	∞	∞	0	∞	5	∞	9
5	∞	∞	∞	∞	0	∞	4	∞
6	∞	∞	∞	∞	1	0	2	∞
7	∞	∞	∞	∞	∞	∞	0	5
8	∞	∞	7	5	∞	3	∞	0

	1	2	3	4	5	6	7	8
1	T	T	T	T	F	F	F	F
2	T	T	F	F	F	T	T	F
3	F	F	T	F	T	T	F	F
4	F	F	F	T	F	T	F	T
5	F	F	F	F	T	F	T	F
6	F	F	F	F	T	T	T	F
7	F	F	F	F	F	F	T	T
8	F	F	T	T	F	T	F	T

Programación Dinámica: Ejercicio 5

Ejemplo 2: elimino las conexiones indicadas

Matriz C de inicio →



Matriz C que obtengo al →
finalizar el algoritmo de
Warshall

	1	2	3	4	5	6	7	8
1	T	T	T	T	F	F	F	F
2	T	T	F	F	F	T	T	F
3	F	F	T	F	T	T	F	F
4	F	F	F	T	F	T	F	T
5	F	F	F	F	T	F	T	F
6	F	F	F	F	T	T	T	F
7	F	F	F	F	F	F	T	T
8	F	F	T	T	F	T	F	T

	1	2	3	4	5	6	7	8
1	T	T	T	T	T	T	T	T
2	T	T	T	T	T	T	T	T
3	F	F	T	T	T	T	T	T
4	F	F	T	T	T	T	T	T
5	F	F	T	T	T	T	T	T
6	F	F	T	T	T	T	T	T
7	F	F	T	T	T	T	T	T
8	F	F	T	T	T	T	T	T

Programación Dinámica: Ejercicio 5

▶ Algoritmo de Warshall

1. Comprobamos las conexiones de 1 con todos los nodos
2. Comprobamos si obtenemos nuevas conexiones usando el nodo 1 como intermedio.
3. Comprobamos si obtenemos nuevas conexiones usando el nodo 2 como intermedio.
4. ...
- N Comprobamos si obtenemos nuevas conexiones usando el nodo N como intermedio.

Programación Dinámica: Ejercicio 5

- ▶ Algoritmo de Warshall

- ▶ Utilizamos una matriz auxiliar C , que me diga si he encontrado una conexión desde el nodo i hasta el nodo j .

- ▶ Inicialmente:

$$C[i][j] = \text{FALSE si } M[i][j] = \infty$$

$$C[i][j] = \text{TRUE si } M[i][j] < \infty$$

- ▶ En cada iteración k ($1 \leq k \leq N$), se tiene que:

$$C[i][j] = (C[i][j]) \vee \{ (C[i][k]) \wedge (C[k][j]) \}$$

Programación Dinámica: Ejercicio 5

```
const N = ...
tipos matriz= array[1... N] [1...N] de booleano
proc Warshall (E M : matriz; E/S C: matriz)
var i, j: entero
    desde i=1 hasta N hacer
        desde j=1 hasta N hacer
            C[i][j] = M[i][j]
        fdesde
    fdesde
    desde k=1 hasta N hacer
        desde i=1 hasta N hacer
            desde j=1 hasta N hacer
                si ( C[i][k] y C[k][j] ) entonces
                    C[i][j] = true;
            fsi
        fdesde
    fdesde
fproc
```