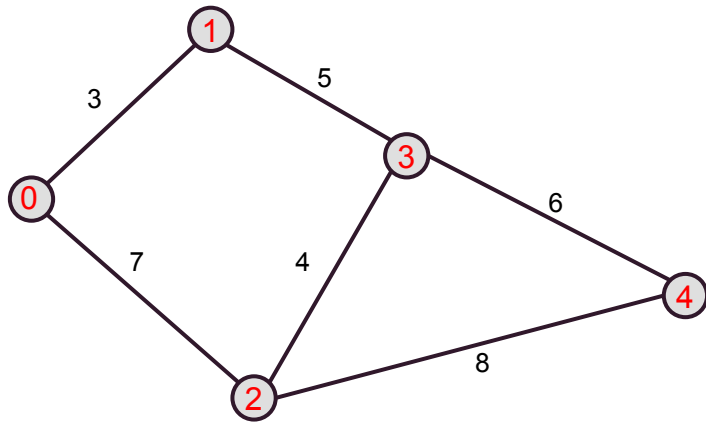


Algoritmo de Floyd- Warshall

Informática - Hoja de Ejercicios 7

Objetivo

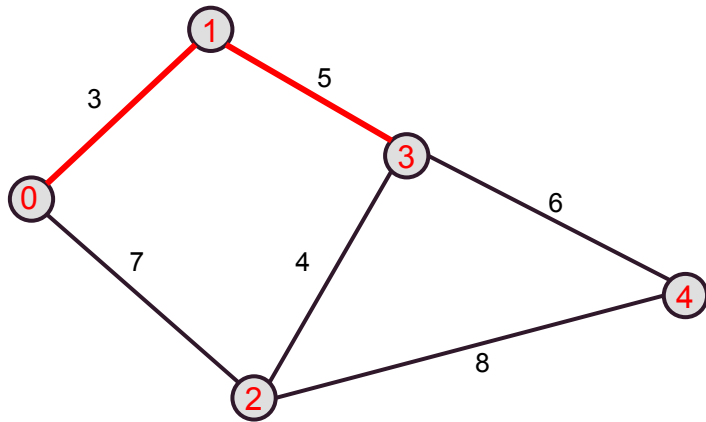
Dado un grafo ponderado, queremos obtener el camino de distancia mínima entre dos vértices cualesquiera.



	0	1	2	3	4
0					
1					
2					
3					
4					

Objetivo

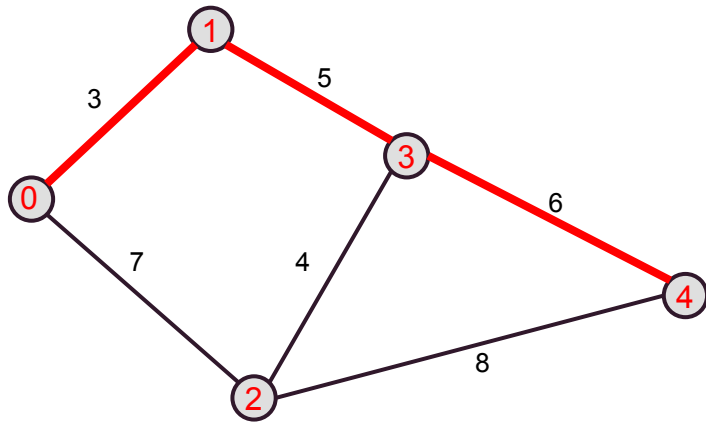
Dado un grafo ponderado, queremos obtener el camino de distancia mínima entre dos vértices cualesquiera.



	0	1	2	3	4
0				8	
1					
2					
3					
4					

Objetivo

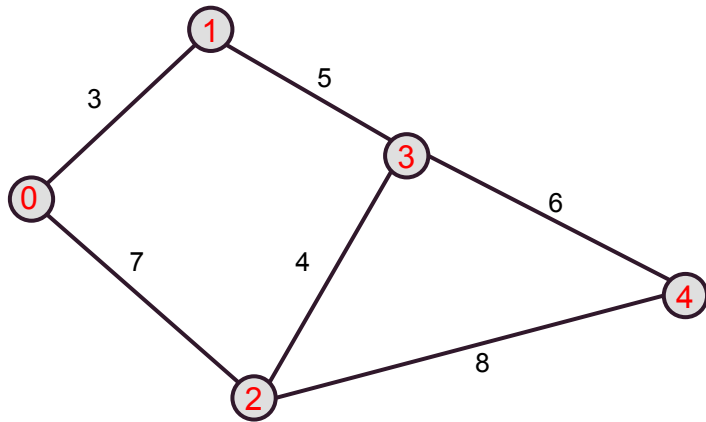
Dado un grafo ponderado, queremos obtener el camino de distancia mínima entre dos vértices cualesquiera.



	0	1	2	3	4
0				8	14
1					
2					
3					
4					

Objetivo

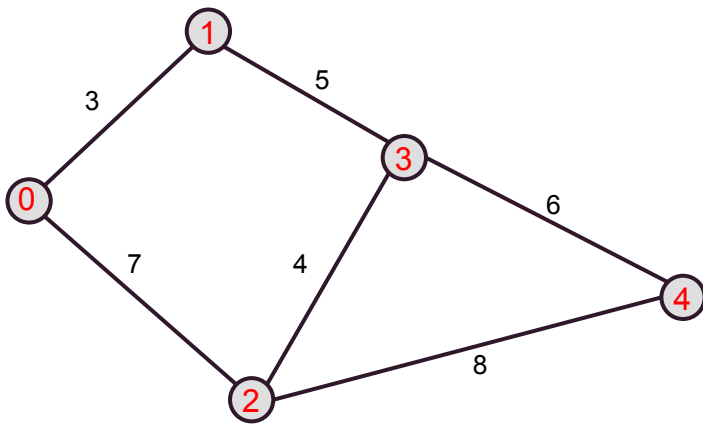
Dado un grafo ponderado, queremos obtener el camino de distancia mínima entre dos vértices cualesquiera.



	0	1	2	3	4
0	0	3	7	8	14
1	3	0	9	5	11
2	7	9	0	4	8
3	8	5	4	0	6
4	14	11	8	6	0

Representación de un grafo

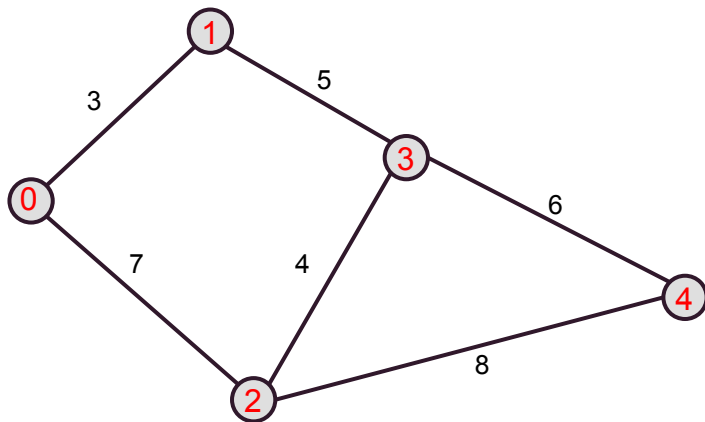
Matriz de adyacencia, con los pesos de cada arista. Si no hay arista entre dos vértices determinados, se considera $+\infty$



	0	1	2	3	4
0	0	3	7	$+\infty$	$+\infty$
1	3	0	$+\infty$	5	$+\infty$
2	7	$+\infty$	0	4	8
3	$+\infty$	5	4	0	6
4	$+\infty$	$+\infty$	8	6	0

Representación de un grafo

Matriz de adyacencia, con los pesos de cada arista. Si no hay arista entre dos vértices determinados, se considera $+\infty$



	0	1	2	3	4
0	0	3	7	$+\infty$	$+\infty$
1	3	0	$+\infty$	5	$+\infty$
2	7	$+\infty$	0	4	8
3	$+\infty$	5	4	0	6
4	$+\infty$	$+\infty$	8	6	0

Simétrica, con ceros en la diagonal

Representación de un grafo

Sólo almacenamos los elementos por debajo de la diagonal principal.

	①	②	③	④	
①	0	3	7	$+\infty$	$+\infty$
②	3	0	$+\infty$	5	$+\infty$
③	7	$+\infty$	0	4	8
④	$+\infty$	5	4	0	6
⑤	$+\infty$	$+\infty$	8	6	0

`[[3], [7, inf], [inf, 5, 4], [inf, inf, 8, 6]]`

①

②

③

④

Representación de un grafo

```
class MatrizCiudades:
    def __init__(self, numeroCiudades):
        ....
    def elemento(self, i, j):
        ....
    def cambiaElemento(self, i, j, n):
        ....
    def numeroCiudades(self):
        return len(self.matriz)
```

Cálculo del camino mínimo

- Supongamos que tenemos n ciudades, numeradas desde 0 hasta $n-1$.
- Construimos una sucesión de matrices:

$$A_0 \rightarrow A_1 \rightarrow A_2 \rightarrow \dots \rightarrow A_n$$

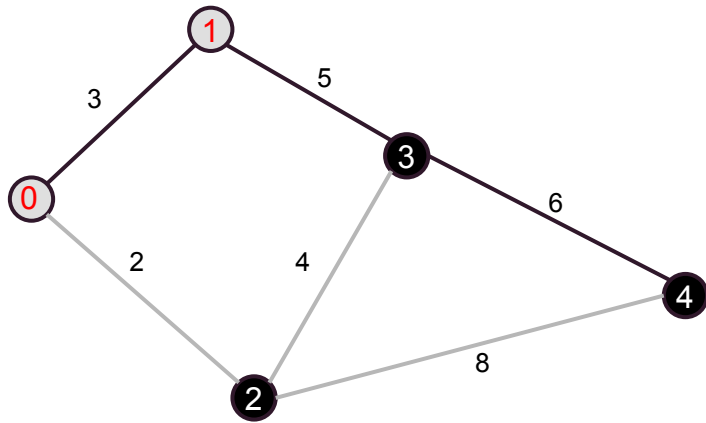
El significado de $A_k(i,j)$ es el siguiente:

Longitud del camino mínimo que hay desde la ciudad i hasta la ciudad j , suponiendo que sólo podemos pasar por las ciudades comprendidas entre 0 y $k-1$.

Cálculo del camino mínimo

Ejemplo: A_2

$A_2(i, j)$ = Longitud del camino mínimo que hay desde la ciudad i hasta la ciudad j , suponiendo que sólo podemos pasar por las ciudades 0 y 1 .

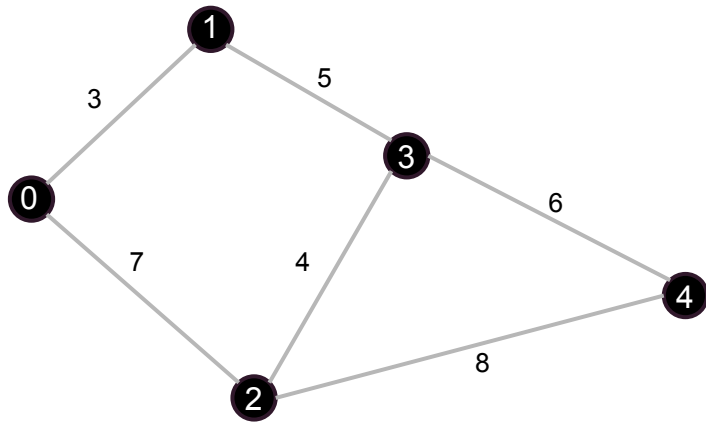


$$A_2(0, 3) = 8$$

$$A_2(0, 4) = +\infty$$

Cálculo del camino mínimo

$A_0(i, j)$ = Longitud del camino mínimo que hay desde la ciudad i hasta la ciudad j , suponiendo no podemos pasar por ninguna ciudad intermedia.

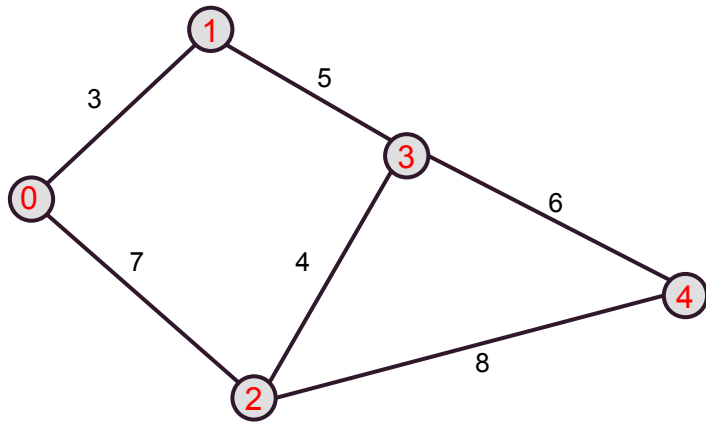


	0	1	2	3	4
0	0	3	7	$+\infty$	$+\infty$
1	3	0	$+\infty$	5	$+\infty$
2	7	$+\infty$	0	4	8
3	$+\infty$	5	4	0	6
4	$+\infty$	$+\infty$	8	6	0

A_0 es la matriz de adyacencia del grafo

Cálculo del camino mínimo

$A_n(i, j)$ = Longitud del camino mínimo que hay desde la ciudad i hasta la ciudad j , suponiendo que podemos pasar por cualquier ciudad intermedia.

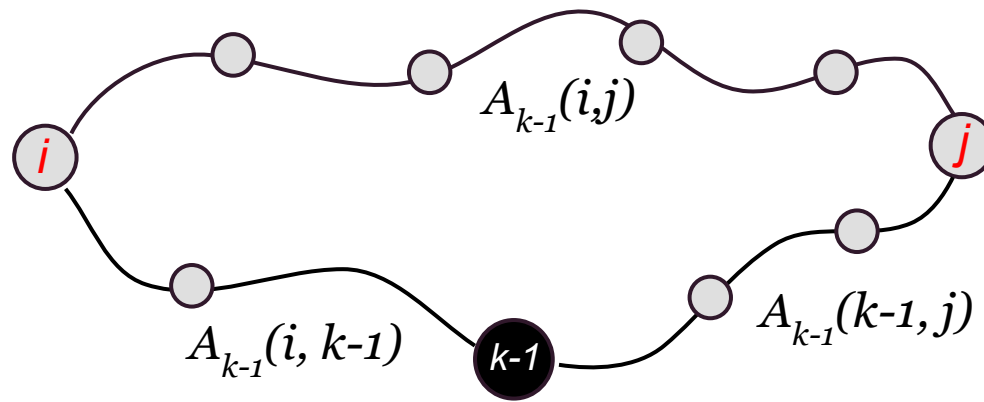


	0	1	2	3	4
0	0	3	7	8	14
1	3	0	9	5	11
2	7	9	0	4	8
3	8	5	4	0	6
4	14	11	8	6	0

A_n es lo que buscamos !

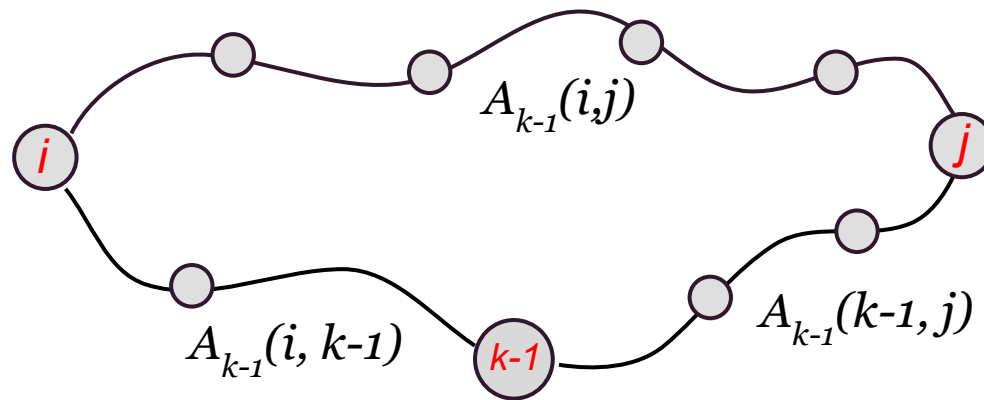
Cálculo del camino mínimo

¿Cómo calcular A_k a partir de A_{k-1} ?



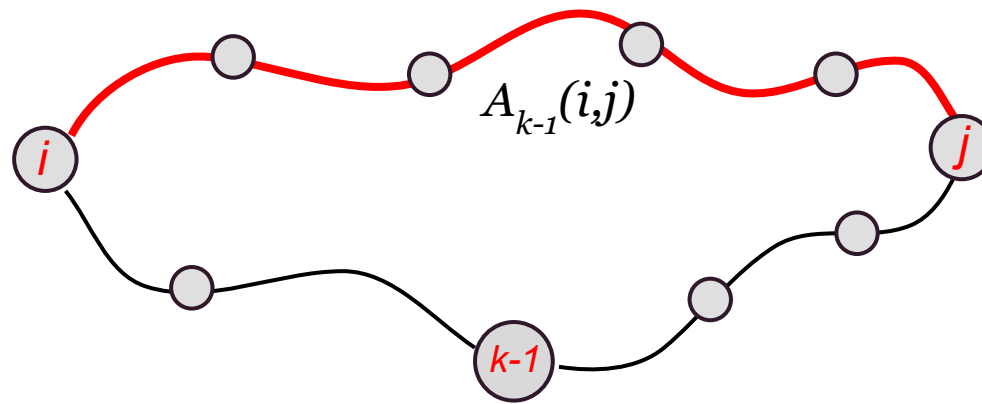
Cálculo del camino mínimo

¿Cómo calcular A_k a partir de A_{k-1} ?



Cálculo del camino mínimo

¿Cómo calcular A_k a partir de A_{k-1} ?

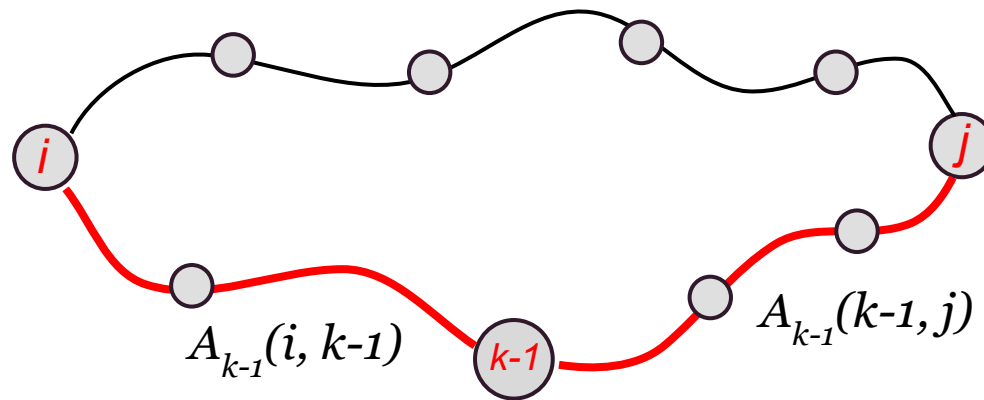


Si el nuevo camino mínimo **no** pasa por el nodo $k-1$

$$A_k(i,j) = A_{k-1}(i,j)$$

Cálculo del camino mínimo

¿Cómo calcular A_k a partir de A_{k-1} ?

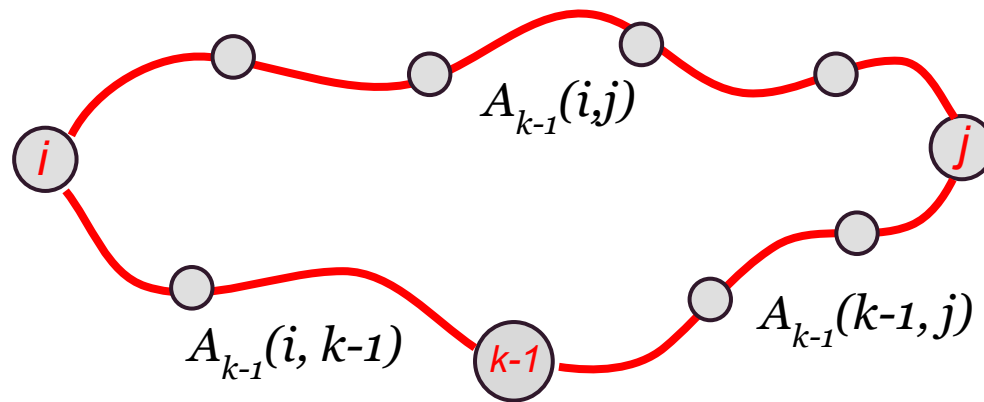


Si el nuevo camino mínimo pasa por el nodo $k-1$

$$A_k(i, j) = A_{k-1}(i, k-1) + A_{k-1}(k-1, j)$$

Cálculo del camino mínimo

¿Cómo calcular A_k a partir de A_{k-1} ?

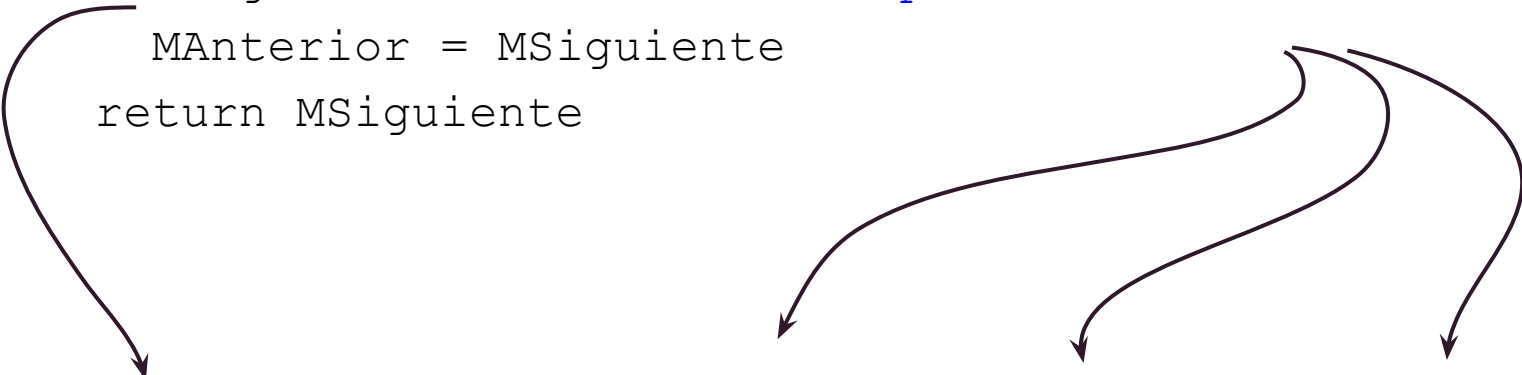


La opción que más nos conviene:

$$A_k(i, j) = \min \{ A_{k-1}(i, j), A_{k-1}(i, k-1) + A_{k-1}(k-1, j) \}$$

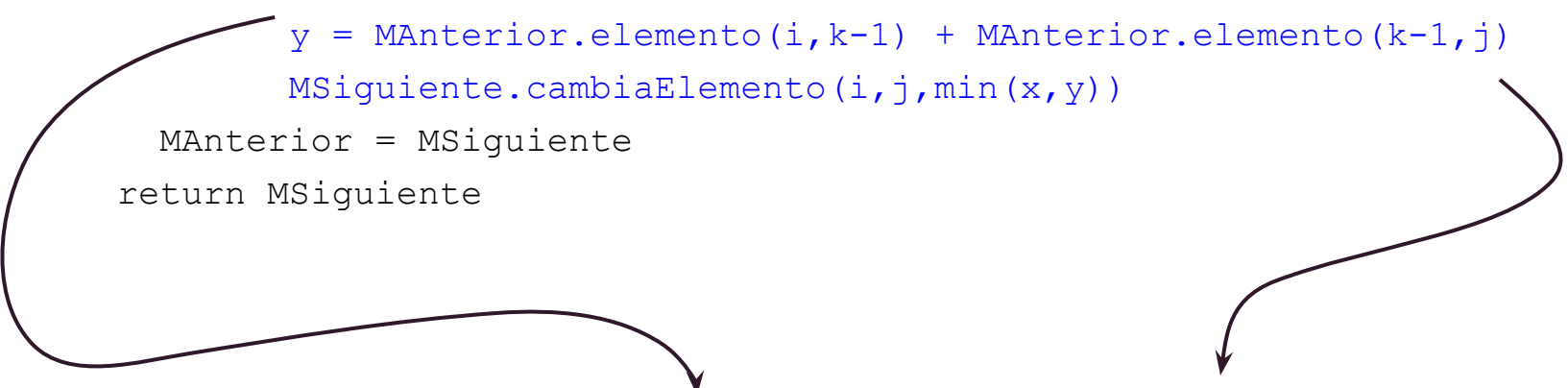
Resumen

```
def caminosMinimos (m):  
    MAnterior = m  
    n = m.numeroCiudades()  
    for k in range(n):  
        MSiguiente = ..calcular a partir de MAnterior..  
        MAnterior = MSiguiente  
    return MSiguiente
```


$$A_k(i,j) = \min \{ A_{k-1}(i,j), A_{k-1}(i,k-1)+A_{k-1}(k-1,j) \}$$

Solución

```
def caminosMinimos (m):  
    MAnterior = m  
    n = m.numeroCiudades()  
    for k in range(n):  
        MSiguiente = MatrizCiudades(n)  
        for i in range(1,n):  
            for j in range(i):  
                x = MAnterior.elemento(i,j)  
                y = MAnterior.elemento(i,k-1) + MAnterior.elemento(k-1,j)  
                MSiguiente.cambiaElemento(i,j,min(x,y))  
        MAnterior = MSiguiente  
    return MSiguiente
```


$$A_k(i,j) = \min \{ A_{k-1}(i,j), A_{k-1}(i,k-1) + A_{k-1}(k-1,j) \}$$