

Algoritmos - Actividad Guiada 1

Nombre: Luis Enrique Sanchez Zamora

URL: <https://colab.research.google.com/drive/1fbVQWsY1nJFSi31cXwpRqMiuUGNzfR9H?usp=sharing>

<https://github.com/sanieni6/03MIAR---Algoritmos-de-Optimizacion>

Torres de Hanoi con Divide y vencerás

In [1]:

```
def Torres_Hanoi(N, desde, hasta):
    if N == 1 :
        print("Lleva la ficha " ,desde , " hasta " , hasta )

    else:
        #Torres_Hanoi(N-1, desde, 6-desde-hasta )
        Torres_Hanoi(N-1, desde, 6-desde-hasta )
        print("Lleva la ficha " ,desde , " hasta " , hasta )
        #Torres_Hanoi(N-1,6-desde-hasta, hasta )
        Torres_Hanoi(N-1, 6-desde-hasta , hasta )
```

Torres_Hanoi(3, 1 , 3)

```
Lleva la ficha 1 hasta 3
Lleva la ficha 1 hasta 2
Lleva la ficha 3 hasta 2
Lleva la ficha 1 hasta 3
Lleva la ficha 2 hasta 1
Lleva la ficha 2 hasta 3
Lleva la ficha 1 hasta 3
```

In [2]:

```
#Sucesión_de_Fibonacci
#https://es.wikipedia.org/wiki/Sucesi%C3%B3n_de_Fibonacci
#Calculo del termino n-simo de la sucesión de Fibonacci
def Fibonacci(N:int):
    if N < 2:
        return 1
    else:
        return Fibonacci(N-1)+Fibonacci(N-2)
```

Fibonacci(5)

Out[2]:

8

Devolución de cambio por técnica voraz

In [3]:

```
def cambio_monedas(N, SM):
    SOLUCION = [0]*len(SM)    #SOLUCION = [0,0,0,0,..]
    ValorAcumulado = 0

    for i,valor in enumerate(SM):
        monedas = (N-ValorAcumulado)//valor
        SOLUCION[i] = monedas
        ValorAcumulado = ValorAcumulado + monedas*valor

    if ValorAcumulado == N:
```

```
return SOLUCION
```

```
cambio_monedas(15,[25,10,5,1])
```

```
Out[3]:
```

```
[0, 1, 1, 0]
```

N-Reinas por técnica de vueta atrás

```
In [4]:
```

```
def escribe(S):
    n = len(S)
    for x in range(n):
        print("")
        for i in range(n):
            if S[i] == x+1:
                print(" X ", end="")
            else:
                print(" - ", end="")

def es_prometedora(SOLUCION,etapa):
    #print(SOLUCION)
    #Si la solución tiene dos valores iguales no es valida => Dos reinas en la misma fila
    for i in range(etapa+1):
        #print("El valor " + str(SOLUCION[i]) + " está " + str(SOLUCION.count(SOLUCION[i]))
        + " veces")
        if SOLUCION.count(SOLUCION[i]) > 1:
            return False

    #Verifica las diagonales
    for j in range(i+1, etapa +1 ):
        #print("Comprobando diagonal de " + str(i) + " y " + str(j))
        if abs(i-j) == abs(SOLUCION[i]-SOLUCION[j]) : return False
    return True

def reinas(N, solucion=[], etapa=0):
    if len(solucion) == 0:
        solucion=[0 for i in range(N)]

    for i in range(1, N+1):
        solucion[etapa] = i

        if es_prometedora(solucion, etapa):
            if etapa == N-1:
                print(solucion)
                escribe(solucion)
                print()
            else:
                reinas(N, solucion, etapa+1)
        else:
            None

    solucion[etapa] = 0

reinas(8)
```

```
[1, 5, 8, 6, 3, 7, 2, 4]
```

```
X - - - - - -
- - - - - X -
- - - X - - -
- - - - - - X
- X - - - - -
- - X - - - -
- - - - X - -
```

-	-	X	-	-	-	-	-
---	---	---	---	---	---	---	---

[1, 6, 8, 3, 7, 4, 2, 5]

X	-	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	X	-	-	-	-
-	-	-	-	-	X	-	-
-	-	-	-	-	-	-	X
-	X	-	-	-	-	-	-
-	-	-	-	X	-	-	-
-	-	X	-	-	-	-	-

[1, 7, 4, 6, 8, 2, 5, 3]

X	-	-	-	-	-	-	-
-	-	-	-	-	X	-	-
-	-	-	-	-	-	-	X
-	-	X	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	X	-	-	-	-
-	X	-	-	-	-	-	-
-	-	-	-	X	-	-	-

[1, 7, 5, 8, 2, 4, 6, 3]

X	-	-	-	-	-	-	-
-	-	-	-	X	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	-	X	-	-
-	-	X	-	-	-	-	-
-	-	-	-	-	-	X	-
-	X	-	-	-	-	-	-
-	-	-	X	-	-	-	-

[2, 4, 6, 8, 3, 1, 7, 5]

-	-	-	-	-	X	-	-
X	-	-	-	-	-	-	-
-	-	-	-	X	-	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	X	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	X	-	-	-	-

[2, 5, 7, 1, 3, 8, 6, 4]

-	-	-	X	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	X	-	-	-
-	-	-	-	-	-	-	X
-	X	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	X	-	-	-	-	-
-	-	-	-	-	X	-	-

[2, 5, 7, 4, 1, 8, 6, 3]

-	-	-	-	X	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	X	-	-	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	X	-	-	-	-	-
-	-	-	-	-	X	-	-

[2, 6, 1, 7, 4, 8, 3, 5]

-	-	X	-	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	-	X	-	-	-
-	-	-	-	-	-	-	X
-	X	-	-	-	-	-	-
-	-	-	X	-	-	-	-
-	-	-	-	-	X	-	-

[2, 6, 8, 3, 1, 4, 7, 5]

-	-	-	-	X	-	-	-
X	-	-	-	-	-	-	-
-	-	-	X	-	-	-	-
-	-	-	-	-	X	-	-
-	-	-	-	-	-	-	X
-	X	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	X	-	-	-	-	-

[2, 7, 3, 6, 8, 5, 1, 4]

-	-	-	-	-	-	X	-
X	-	-	-	-	-	-	-
-	-	X	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	-	X	-	-
-	-	-	X	-	-	-	-
-	X	-	-	-	-	-	-
-	-	-	-	X	-	-	-

[2, 7, 5, 8, 1, 4, 6, 3]

-	-	-	-	X	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	-	X	-	-
-	-	X	-	-	-	-	-
-	-	-	-	-	-	X	-
-	X	-	-	-	-	-	-
-	-	-	X	-	-	-	-

[2, 8, 6, 1, 3, 5, 7, 4]

-	-	-	X	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	X	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	-	X	-	-
-	-	X	-	-	-	-	-
-	-	-	-	-	-	X	-
-	X	-	-	-	-	-	-

[3, 1, 7, 5, 8, 2, 4, 6]

-	X	-	-	-	-	-	-
-	-	-	-	-	X	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	X	-	-	-	-
-	-	-	-	-	-	-	X
-	-	X	-	-	-	-	-
-	-	-	-	X	-	-	-

[3, 5, 2, 8, 1, 7, 4, 6]

-	-	-	-	X	-	-	-
-	-	X	-	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	-	X	-	-
-	-	-	X	-	-	-	-

[3, 5, 2, 8, 6, 4, 7, 1]

-	-	-	-	-	-	-	X
-	-	X	-	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	X	-	-
-	X	-	-	-	-	-	-
-	-	-	-	X	-	-	-
-	-	-	-	-	-	X	-
-	-	-	X	-	-	-	-

[3, 5, 7, 1, 4, 2, 8, 6]

-	-	-	X	-	-	-	-
---	---	---	---	---	---	---	---

-	-	-	-	-	X	-	-
X	-	-	-	-	-	-	-
-	-	-	-	X	-	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	X	-	-	-	-	-
-	-	-	-	-	-	X	-

[3, 5, 8, 4, 1, 7, 2, 6]

-	-	-	-	X	-	-	-
-	-	-	-	-	-	X	-
X	-	-	-	-	-	-	-
-	-	-	X	-	-	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	-	X	-	-
-	-	X	-	-	-	-	-

[3, 6, 2, 5, 8, 1, 7, 4]

-	-	-	-	-	X	-	-
-	-	X	-	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	X	-	-	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	-	X	-	-	-

[3, 6, 2, 7, 1, 4, 8, 5]

-	-	-	-	X	-	-	-
-	-	X	-	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	X	-	-
-	-	-	-	-	-	-	X
-	X	-	-	-	-	-	-
-	-	-	X	-	-	-	-
-	-	-	-	-	-	X	-

[3, 6, 2, 7, 5, 1, 8, 4]

-	-	-	-	-	X	-	-
-	-	X	-	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	X	-	-	-
-	X	-	-	-	-	-	-
-	-	-	X	-	-	-	-
-	-	-	-	-	-	X	-

[3, 6, 4, 1, 8, 5, 7, 2]

-	-	-	X	-	-	-	-
-	-	-	-	-	-	-	X
X	-	-	-	-	-	-	-
-	-	X	-	-	-	-	-
-	-	-	-	-	X	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	-	X	-	-	-

[3, 6, 4, 2, 8, 5, 7, 1]

-	-	-	-	-	-	-	X
-	-	-	X	-	-	-	-
X	-	-	-	-	-	-	-
-	-	X	-	-	-	-	-
-	-	-	-	-	X	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	-	X	-	-	-

[3, 6, 8, 1, 4, 7, 5, 2]

-	-	-	X	-	-	-	-
-	-	-	-	-	-	-	X
X	-	-	-	-	-	-	-

-	-	-	-	X	-	-	-
-	-	-	-	-	-	X	-
-	X	-	-	-	-	-	-
-	-	-	-	-	X	-	-
-	-	X	-	-	-	-	-

[3, 6, 8, 1, 5, 7, 2, 4]

-	-	-	X	-	-	-	-
-	-	-	-	-	-	X	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	X	-	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	X	-	-
-	-	X	-	-	-	-	-

[3, 6, 8, 2, 4, 1, 7, 5]

-	-	-	-	-	X	-	-
-	-	-	X	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	X	-	-	-
-	-	-	-	-	-	-	X
-	X	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	X	-	-	-	-	-

[3, 7, 2, 8, 5, 1, 4, 6]

-	-	-	-	-	X	-	-
-	-	X	-	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	-	X	-	-	-
-	-	-	-	-	-	-	X
-	X	-	-	-	-	-	-
-	-	-	X	-	-	-	-

[3, 7, 2, 8, 6, 4, 1, 5]

-	-	-	-	-	-	X	-
-	-	X	-	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	X	-	-
-	-	-	-	-	-	-	X
-	-	-	-	X	-	-	-
-	X	-	-	-	-	-	-
-	-	-	X	-	-	-	-

[3, 8, 4, 7, 1, 6, 2, 5]

-	-	-	-	X	-	-	-
-	-	-	-	-	-	X	-
X	-	-	-	-	-	-	-
-	-	X	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	-	X	-	-
-	-	-	X	-	-	-	-
-	X	-	-	-	-	-	-

[4, 1, 5, 8, 2, 7, 3, 6]

-	X	-	-	-	-	-	-
-	-	-	-	X	-	-	-
-	-	-	-	-	-	X	-
X	-	-	-	-	-	-	-
-	-	X	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	-	X	-	-
-	-	-	X	-	-	-	-

[4, 1, 5, 8, 6, 3, 7, 2]

-	X	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	-	X	-	-
X	-	-	-	-	-	-	-
-	-	X	-	-	-	-	-

-	-	-	-	X	-	-	-
-	-	-	-	-	-	X	-
-	-	-	X	-	-	-	-

[4, 2, 5, 8, 6, 1, 3, 7]

-	-	-	-	-	X	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	X	-
X	-	-	-	-	-	-	-
-	-	X	-	-	-	-	-
-	-	-	-	X	-	-	-
-	-	-	-	-	-	-	X
-	-	-	X	-	-	-	-

[4, 2, 7, 3, 6, 8, 1, 5]

-	-	-	-	-	-	X	-
-	X	-	-	-	-	-	-
-	-	-	X	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	X	-	-	-
-	-	X	-	-	-	-	-
-	-	-	-	-	X	-	-

[4, 2, 7, 3, 6, 8, 5, 1]

-	-	-	-	-	-	-	X
-	X	-	-	-	-	-	-
-	-	-	X	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	-	X	-	-	-
-	-	X	-	-	-	-	-
-	-	-	-	-	X	-	-

[4, 2, 7, 5, 1, 8, 6, 3]

-	-	-	-	X	-	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	-	X
X	-	-	-	-	-	-	-
-	-	-	X	-	-	-	-
-	-	-	-	-	-	X	-
-	-	X	-	-	-	-	-
-	-	-	-	X	-	-	-

[4, 2, 8, 5, 7, 1, 3, 6]

-	-	-	-	-	X	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	X	-
X	-	-	-	-	-	-	-
-	-	-	X	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	X	-	-	-
-	-	X	-	-	-	-	-

[4, 2, 8, 6, 1, 3, 5, 7]

-	-	-	-	X	-	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	X	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	X	-	-	-	-
-	-	-	-	-	-	-	X
-	-	X	-	-	-	-	-

[4, 6, 1, 5, 2, 8, 3, 7]

-	-	X	-	-	-	-	-
-	-	-	-	X	-	-	-
-	-	-	-	-	-	X	-
X	-	-	-	-	-	-	-
-	-	-	X	-	-	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	-	X

-	-	X	-	-	-	-	-
-	-	-	-	-	X	-	-
-	-	-	-	-	-	-	X
X	-	-	-	-	-	-	-
-	-	-	X	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	-	X	-	-	-
-	X	-	-	-	-	-	-

[4, 8, 5, 3, 1, 7, 2, 6]

-	-	-	-	X	-	-	-
-	-	-	-	-	-	X	-
-	-	-	X	-	-	-	-
X	-	-	-	-	-	-	-
-	-	X	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	-	X	-	-
-	X	-	-	-	-	-	-

[5, 1, 4, 6, 8, 2, 7, 3]

-	X	-	-	-	-	-	-
-	-	-	-	-	X	-	-
-	-	-	-	-	-	-	X
-	-	X	-	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	X	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	-	X	-	-	-

[5, 1, 8, 4, 2, 7, 3, 6]

-	X	-	-	-	-	-	-
-	-	-	-	X	-	-	-
-	-	-	-	-	-	X	-
-	-	-	X	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	-	X	-	-
-	-	X	-	-	-	-	-

[5, 1, 8, 6, 3, 7, 2, 4]

-	X	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	-	X	-	-	-
-	-	-	-	-	-	-	X
X	-	-	-	-	-	-	-
-	-	-	X	-	-	-	-
-	-	-	-	-	X	-	-
-	-	X	-	-	-	-	-

[5, 2, 4, 6, 8, 3, 1, 7]

-	-	-	-	-	-	X	-
-	X	-	-	-	-	-	-
-	-	-	-	-	X	-	-
-	-	X	-	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	X	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	X	-	-	-

[5, 2, 4, 7, 3, 8, 6, 1]

-	-	-	-	-	-	-	X
-	X	-	-	-	-	-	-
-	-	-	-	X	-	-	-
-	-	X	-	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	X	-	-	-	-
-	-	-	-	-	X	-	-

[5, 2, 6, 1, 7, 4, 8, 3]

-	-	-	X	-	-	-	-
---	---	---	---	---	---	---	---

-	X	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	-	X	-	-
X	-	-	-	-	-	-	-
-	-	X	-	-	-	-	-
-	-	-	-	X	-	-	-
-	-	-	-	-	-	X	-

[5, 2, 8, 1, 4, 7, 3, 6]

-	-	-	X	-	-	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	-	X	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	-	X	-	-
-	-	X	-	-	-	-	-

[5, 3, 1, 6, 8, 2, 4, 7]

-	-	X	-	-	-	-	-
-	-	-	-	-	X	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	X	-
X	-	-	-	-	-	-	-
-	-	-	X	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	X	-	-	-

[5, 3, 1, 7, 2, 8, 6, 4]

-	-	X	-	-	-	-	-
-	-	-	-	X	-	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	-	X
X	-	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	X	-	-	-	-
-	-	-	-	-	X	-	-

[5, 3, 8, 4, 7, 1, 6, 2]

-	-	-	-	-	X	-	-
-	-	-	-	-	-	-	X
-	X	-	-	-	-	-	-
-	-	-	X	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	-	X	-	-	-
-	-	X	-	-	-	-	-

[5, 7, 1, 3, 8, 6, 4, 2]

-	-	X	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	X	-	-	-	-
-	-	-	-	-	-	X	-
X	-	-	-	-	-	-	-
-	-	-	-	-	X	-	-
-	X	-	-	-	-	-	-
-	-	-	-	X	-	-	-

[5, 7, 1, 4, 2, 8, 6, 3]

-	-	X	-	-	-	-	-
-	-	-	-	X	-	-	-
-	-	-	-	-	-	-	X
-	-	-	X	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	X	-	-	-	-	-	-
-	-	-	-	-	X	-	-

[5, 7, 2, 4, 8, 1, 3, 6]

-	-	-	-	-	X	-	-
-	-	X	-	-	-	-	-
-	-	-	-	-	-	X	-

-	-	-	X	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	X	-	-	-	-	-	-
-	-	-	-	X	-	-	-

[5, 7, 2, 6, 3, 1, 4, 8]

-	-	-	-	-	X	-	-
-	-	X	-	-	-	-	-
-	-	-	-	X	-	-	-
-	-	-	-	-	-	X	-
X	-	-	-	-	-	-	-
-	-	-	X	-	-	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	-	X

[5, 7, 2, 6, 3, 1, 8, 4]

-	-	-	-	-	X	-	-
-	-	X	-	-	-	-	-
-	-	-	-	X	-	-	-
-	-	-	-	-	-	-	X
X	-	-	-	-	-	-	-
-	-	-	X	-	-	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	X	-

[5, 7, 4, 1, 3, 8, 6, 2]

-	-	-	X	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	X	-	-	-
-	-	X	-	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	X	-	-	-	-	-	-
-	-	-	-	-	X	-	-

[5, 8, 4, 1, 3, 6, 2, 7]

-	-	-	X	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	-	X	-	-	-
-	-	X	-	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	X	-	-
-	-	-	-	-	-	-	X
-	X	-	-	-	-	-	-

[5, 8, 4, 1, 7, 2, 6, 3]

-	-	-	X	-	-	-	-
-	-	-	-	-	X	-	-
-	-	-	-	-	-	-	X
-	-	X	-	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	-	X	-	-	-
-	X	-	-	-	-	-	-

[6, 1, 5, 2, 8, 3, 7, 4]

-	X	-	-	-	-	-	-
-	-	-	X	-	-	-	-
-	-	-	-	-	X	-	-
-	-	-	-	-	-	-	X
-	-	X	-	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	-	X	-	-	-

[6, 2, 7, 1, 3, 5, 8, 4]

-	-	-	X	-	-	-	-
-	X	-	-	-	-	-	-
-	-	-	-	X	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	-	X	-	-

X	-	-	-	-	-	-	-
-	-	X	-	-	-	-	-
-	-	-	-	-	-	X	-

[6, 2, 7, 1, 4, 8, 5, 3]

-	-	-	X	-	-	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	X	-	-	-
-	-	-	-	-	-	X	-
X	-	-	-	-	-	-	-
-	-	X	-	-	-	-	-
-	-	-	-	-	X	-	-

[6, 3, 1, 7, 5, 8, 2, 4]

-	-	X	-	-	-	-	-
-	-	-	-	-	-	X	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	X	-	-	-
X	-	-	-	-	-	-	-
-	-	-	X	-	-	-	-
-	-	-	-	-	X	-	-

[6, 3, 1, 8, 4, 2, 7, 5]

-	-	X	-	-	-	-	-
-	-	-	-	-	X	-	-
-	X	-	-	-	-	-	-
-	-	-	-	X	-	-	-
-	-	-	-	-	-	-	X
X	-	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	X	-	-	-	-

[6, 3, 1, 8, 5, 2, 4, 7]

-	-	X	-	-	-	-	-
-	-	-	-	-	X	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	-	-	X	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	X	-	-	-	-

[6, 3, 5, 7, 1, 4, 2, 8]

-	-	-	-	X	-	-	-
-	-	-	-	-	-	X	-
-	X	-	-	-	-	-	-
-	-	-	-	-	X	-	-
-	-	X	-	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	X	-	-	-	-
-	-	-	-	-	-	-	X

[6, 3, 5, 8, 1, 4, 2, 7]

-	-	-	-	X	-	-	-
-	-	-	-	-	-	X	-
-	X	-	-	-	-	-	-
-	-	-	-	-	X	-	-
-	-	X	-	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	X	-	-	-	-

[6, 3, 7, 2, 4, 8, 1, 5]

-	-	-	-	-	-	X	-
-	-	-	X	-	-	-	-
-	X	-	-	-	-	-	-
-	-	-	-	X	-	-	-
-	-	-	-	-	-	-	X
X	-	-	-	-	-	-	-
-	-	X	-	-	-	-	-

- - - - X - -
[6, 3, 7, 2, 8, 5, 1, 4]

- - - - - X -
- - - X - - - -
- X - - - - -
- - - - - - X
- - - - - X - -
X - - - - - -
- - X - - - - -
- - - - X - - -
[6, 3, 7, 4, 1, 8, 2, 5]

- - - - X - - -
- - - - - X -
- X - - - - -
- - - X - - - -
- - - - - - X
X - - - - - -
- - X - - - - -
- - - - X - - -
[6, 4, 1, 5, 8, 2, 7, 3]

- - X - - - -
- - - - - X - -
- - - - - - X
- X - - - - -
- - - X - - - -
X - - - - - -
- - - - - X -
- - - - X - - -
[6, 4, 2, 8, 5, 7, 1, 3]

- - - - - X -
- - X - - - -
- - - - - - X
- X - - - - -
- - - - X - - -
X - - - - - -
- - - - - X - -
- - - X - - - -
[6, 4, 7, 1, 3, 5, 2, 8]

- - - X - - - -
- - - - - X -
- - - - X - - -
- X - - - - -
- - - - - X - -
X - - - - - -
- - X - - - - -
- - - - - - X
[6, 4, 7, 1, 8, 2, 5, 3]

- - - X - - - -
- - - - - X - -
- - - - - - X
- X - - - - -
- - - - - X -
X - - - - - -
- - X - - - - -
- - - - X - - -
[6, 8, 2, 4, 1, 7, 5, 3]

- - - - X - - -
- - X - - - -
- - - - - - X
- - - X - - - -
- - - - - X -
X - - - - - -
- - - - X - - -
- X - - - - -
[7, 1, 3, 8, 6, 4, 2, 5]

-	X	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	X	-	-	-	-	-
-	-	-	-	-	X	-	-
-	-	-	-	-	-	-	X
-	-	-	-	X	-	-	-
X	-	-	-	-	-	-	-
-	-	-	X	-	-	-	-

[7, 2, 4, 1, 8, 5, 3, 6]

-	-	-	X	-	-	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	X	-
-	-	X	-	-	-	-	-
-	-	-	-	-	X	-	-
-	-	-	-	-	-	-	X
X	-	-	-	-	-	-	-
-	-	-	X	-	-	-	-

[7, 2, 6, 3, 1, 4, 8, 5]

-	-	-	-	X	-	-	-
-	X	-	-	-	-	-	-
-	-	-	X	-	-	-	-
-	-	-	-	-	X	-	-
-	-	-	-	-	-	-	X
-	-	X	-	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	-	-	X	-

[7, 3, 1, 6, 8, 5, 2, 4]

-	-	X	-	-	-	-	-
-	-	-	-	-	-	X	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	-	X	-	-
-	-	-	X	-	-	-	-
X	-	-	-	-	-	-	-
-	-	-	-	X	-	-	-

[7, 3, 8, 2, 5, 1, 6, 4]

-	-	-	-	-	X	-	-
-	-	-	X	-	-	-	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	X	-	-	-
-	-	-	-	-	-	X	-
X	-	-	-	-	-	-	-
-	-	X	-	-	-	-	-

[7, 4, 2, 5, 8, 1, 3, 6]

-	-	-	-	-	X	-	-
-	-	X	-	-	-	-	-
-	-	-	-	-	-	X	-
-	X	-	-	-	-	-	-
-	-	-	X	-	-	-	-
-	-	-	-	-	-	-	X
X	-	-	-	-	-	-	-
-	-	-	-	X	-	-	-

[7, 4, 2, 8, 6, 1, 3, 5]

-	-	-	-	-	X	-	-
-	-	X	-	-	-	-	-
-	-	-	-	-	-	X	-
-	X	-	-	-	-	-	-
-	-	-	-	-	-	-	X
-	-	-	-	X	-	-	-
X	-	-	-	-	-	-	-
-	-	-	X	-	-	-	-

[7, 5, 3, 1, 6, 8, 2, 4]

-	-	-	X	-	-	-	-
---	---	---	---	---	---	---	---

```

- - - - - X -
- - X - - - -
- - - - - - X
- X - - - - -
- - - - X - -
X - - - - -
- - - - X - -
[8, 2, 4, 1, 7, 5, 3, 6]

```

```

- - - X - - -
- X - - - - -
- - - - - X -
- - X - - - -
- - - - X - -
- - - - - X
- - - - X - -
X - - - - -
[8, 2, 5, 3, 1, 7, 4, 6]

```

```

- - - - X - -
- X - - - - -
- - - X - - -
- - - - - X -
- - X - - - -
- - - - - X
- - - - X - -
X - - - - -
[8, 3, 1, 6, 2, 5, 7, 4]

```

```

- - X - - - -
- - - - X - -
- X - - - - -
- - - - - X
- - - - X - -
- - - X - - -
- - - - - X -
X - - - - -
[8, 4, 1, 3, 6, 2, 7, 5]

```

```

- - X - - - -
- - - - X - -
- - - X - - -
- X - - - - -
- - - - - X
- - - - X - -
- - - - - X -
X - - - - -

```

Viaje por el rio. Programación dinámica

In [5]:

```

TARIFAS = [
[0,5,4,3,999,999,999],
[999,0,999,2,3,999,11],
[999,999, 0,1,999,4,10],
[999,999,999, 0,5,6,9],
[999,999, 999,999,0,999,4],
[999,999, 999,999,999,0,3],
[999,999,999,999,999,999,0]
]

#####
def Precios(TARIFAS):
#####
    #Total de Nodos
    N = len(TARIFAS[0])

```

```

#Inicialización de la tabla de precios
PRECIOS = [ [9999]*N for i in [9999]*N]
RUTA = [ [""]*N for i in [""]*N]

for i in range(0,N-1):
    RUTA[i][i] = i #Para ir de i a i se "pasa por i"
    PRECIOS[i][i] = 0 #Para ir de i a i se se paga 0
    for j in range(i+1, N):
        MIN = TARIFAS[i][j]
        RUTA[i][j] = i

        for k in range(i, j):
            if PRECIOS[i][k] + TARIFAS[k][j] < MIN:
                MIN = min(MIN, PRECIOS[i][k] + TARIFAS[k][j] )
                RUTA[i][j] = k #Anota que para ir de i a j hay que pasar por k
            PRECIOS[i][j] = MIN

    return PRECIOS,RUTA
#####

PRECIOS,RUTA = Precios(TARIFAS)
#print(PRECIOS[0][6])

print("PRECIOS")
for i in range(len(TARIFAS)):
    print(PRECIOS[i])

print("\nRUTA")
for i in range(len(TARIFAS)):
    print(RUTA[i])

#Determinar la ruta con Recursividad
def calcular_ruta(RUTA, desde, hasta):
    if desde == hasta:
        print("Ir a :" + str(desde))
        return ""
    else:
        return str(calcular_ruta( RUTA, desde, RUTA[desde][hasta])) + \
            ',' + \
            str(RUTA[desde][hasta] \
            )

print("\nLa ruta es:")
calcular_ruta(RUTA, 0,6)

```

```

PRECIOS
[0, 5, 4, 3, 8, 8, 11]
[9999, 0, 999, 2, 3, 8, 7]
[9999, 9999, 0, 1, 6, 4, 7]
[9999, 9999, 9999, 0, 5, 6, 9]
[9999, 9999, 9999, 9999, 0, 999, 4]
[9999, 9999, 9999, 9999, 9999, 0, 3]
[9999, 9999, 9999, 9999, 9999, 9999, 9999]

```

```

RUTA
[0, 0, 0, 0, 1, 2, 5]
['', 1, 1, 1, 1, 3, 4]
['', '', 2, 2, 3, 2, 5]
['', '', '', 3, 3, 3, 3]
['', '', '', '', 4, 4, 4]
['', '', '', '', '', 5, 5]
['', '', '', '', '', '', '']

```

La ruta es:
Ir a :0

Out[5]:
' ,0,2,5'

Problema: Encontrar los dos puntos mas cercanos

Dado un conjunto de puntos en el plano, encontrar los dos puntos mas cercanos.

In [9]:

```
import random
lista_1d = [random.randrange(0, 10000) for i in range(10000)]
lista_2d = [(random.randrange(0, 10000), random.randrange(0, 10000)) for i in range(1000)]
```

Fuerza bruta

In [10]:

```
def par_mas_cercano_fuerza_bruta(puntos):
    distancia_minima = float('inf')
    par = None
    for i in range(len(puntos)):
        for j in range(i+1, len(puntos)):
            distancia = abs(puntos[i] - puntos[j])
            if distancia < distancia_minima:
                distancia_minima = distancia
                par = (puntos[i], puntos[j])
    return par

# puntos = [3403, 4537, 9089, 9746, 7259]
par_mas_cercano_fuerza_bruta(lista_1d)
```

Out[10]:

(3732, 3732)

complejidad = $O(n^2)$

Dividir y vencerás 1D

In [11]:

```
def par_mas_cercano_divide_y_venceras_1D(puntos):
    sorted_puntos = sorted(puntos)
    distancia_minima = float('inf')
    par = None
    for i in range(1, len(sorted_puntos)):
        distancia = abs(sorted_puntos[i] - sorted_puntos[i-1])
        if distancia < distancia_minima:
            distancia_minima = distancia
            par = (sorted_puntos[i-1], sorted_puntos[i])
    return par

# puntos = [3403, 4537, 9089, 9746, 7259]
par_mas_cercano_divide_y_venceras_1D(lista_1d)
```

Out[11]:

(9, 9)

complejidad = $O(n \log n)$

In []: